Programmieren in der IBM/360 ASSEMBLER-Sprache

IBM

ASS-PU
Text 3
Programmierte Unterweisung

IBM Schule für Datenverarbeitung

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

Eine Programmierte Unterweisung

Text 3

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:	Das symbolische Programm und seine Umwandlung in das Maschinenprogramm	5
Kapitel II:	Das Programmschema	47
Kapitel III:	Die Assembleranweisungen START, END, ORG, DS	81
Kapitel IV:	Die Definition der Konstanten	185
Kapitel V:	Die Assembleranweisungen USING und DROP	273
Kapitel VI:	Das Operanden-Format für Instruktionen im SS-Format	307

KAPITEL I

DAS SYMBOLISCHE PROGRAMM UND SEINE UMWANDLUNG IN DAS MASCHINENPROGRAMM

Lernstoff:

Die Assemblersprache, eine symbolische Programmiersprache

Die Instruktion in der Assemblersprache

Die Umwandlung des symbolischen Programms in das Maschinenprogramm

Die Aufgabe des Programmierers ist es, ein Problem der Datenverarbeitung in einem Programmablaufplan (früher Blockdiagramm genannt) darzustellen und entsprechend hierzu das Programm für das Datenverarbeitungssystem (DVS) zu schreiben.

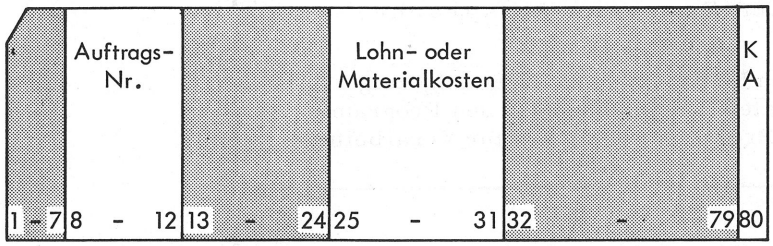
Ein Programmablaufplan ist die Auflösung eines Problems in Einzelschritte. Der Programmablaufplan ist die Grundlage für das Schreiben eines Programms, denn die Funktionen und der Ablauf eines Programmes sind im Programmablaufplan schon bildhaft vorgegeben.

Um dem DVS bestimmte Anweisungen zur Verarbeitung der Daten zu geben, bedient man sich einer besonderen Sprache, nämlich der Programmiersprache.

In dem vorliegenden Kapitel wird der Weg aufgezeigt, den ein Problem der Datenverarbeitung über den Programmablaufplan und die Programmierung bis zur Verarbeitung der Daten nimmt.

ERSTELLEN EINER LISTE DER LOHN- UND MATERIALKOSTEN

Eingabe: Lohn- und Materialkarten



KA = Kartenart (Spalte 80)
Kartenart 1 = Lohnkarte
Kartenart 2 = Materialkarte

Die letzte Karte des Kartenpakets ist die "ENDE-Karte" (Spalten 1 - 4 enthalten ENDE).

Ausgabe: Liste

Auftrags-Nr.	Lohnkosten	Materialkosten
00513	943,50	
00513		2.051,03
00513		758,81
00513	1.236,39	
01024	1.098,75	
01024		382,56
01024	885,86	
01024		1.340,19
03152	1.125,70	
03152		826,63
03152		92,84
03152	974,23	
03604	4.247,14	
03604		10.387,36

Druckstellen: 4 - 8 | 11 - 19 | 22 - 30

Abb. 1

Betrachten Sie bitte Abb. 1 genau und stellen Sie fest, um welche Aufgabenstellung es sich in dieser Abbildung handelt. -

Wie Sie sehen, soll eine Liste der Lohn- und Materialkosten erstellt werden. In das DVS werden Lohn- und Materialkarten eingelesen, die folgende Daten enthalten:

a) eine Auftrags-Nr. (Spalte 8-12),

b) entweder Lohnkosten oder Materialkosten (Spalte 25-31),

c) die Kartenart (Spalte 80).

Lohnkarten enthalten in den Spalten 25-31 Lohnkosten und in der Spalte 80 die Kartenart (Kennlochung) 1. Materialkarten enthalten in den Spalten 25-31 Materialkosten und in Spalte 80 die Kartenart (Kennlochung) 2.

Abhängig von der Kartenart sollen nun die Beträge aus den Spalten 25-31 entweder unter Lohnkosten oder unter Materialkosten auf der Liste aufgeführt werden.

So oder in ähnlicher Form werden die Probleme dem Programmierer unterbreitet. Sicherlich sind diese Probleme oft umfangreicher und komplizierter, wahrscheinlich auch manchmal nicht so deutlich aufgezeigt.

Dennoch muß eines immer an erster Stelle stehen:

Die Problemstellung muß k l a r e r k a n n t und f e s t u m r i s s e n sein.

PROGRAMMABLAUFPLAN

Eingabe: Lohn- und Materialkarten

	Auftrags-Nr.		Lohn- oder Materialkosten		KA
1 - 7	8 - 12	13 - 24	25 - 31	32 - 79	80

KA = Kartenart (Spalte 80)
Kartenart 1 = Lohnkarte
Kartenart 2 = Materialkarte

Die letzte Karte des Kartenpakets ist die "ENDE-Karte" (Spalten 1 - 4 enthalten ENDE).

Ausgabe: Liste

Auftrags-Nr.	Lohnkosten	Materialkosten
00513	943,50	
00513		2.051,03
00513		758,81
00513	1.236,39	
01024	1.098,75	
01024		382,56
01024	885,86	
01024		1.340,19
03152	1.125,70	
03152		826,63
03152		92,84
03152	974,23	
03604	4.247,14	
03604		10.387,36

Druckstellen: 4 - 8 | 11 - 19 | 22 - 30

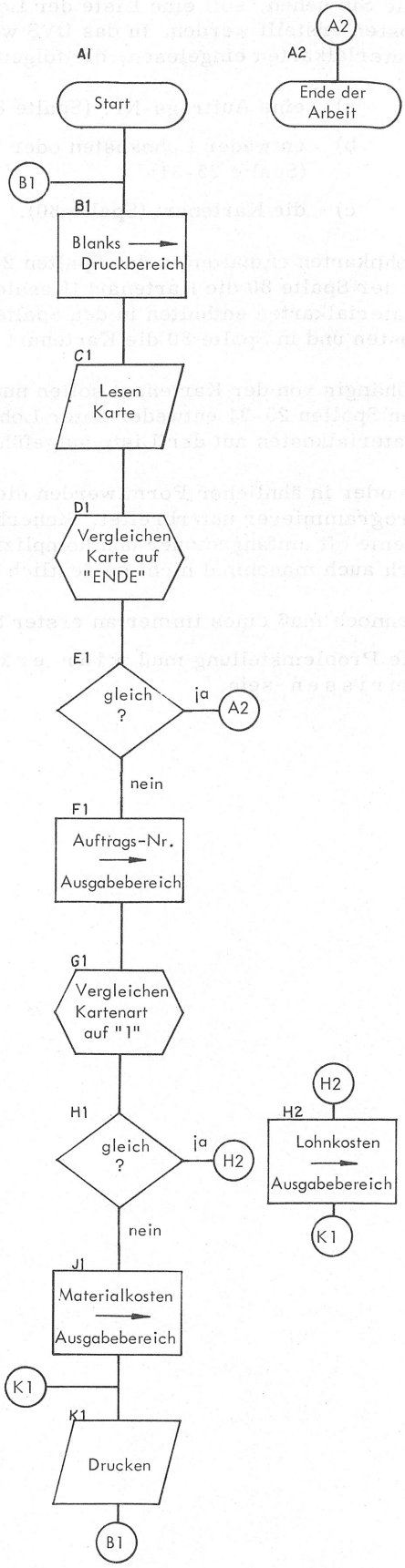


Abb. 2

1 Betrachten Sie bitte Abb. 2. -

Nachdem eine Problemstellung analysiert und klar erkannt ist, kann ein weiterer Schritt auf dem Wege zum Schreiben eines Programms getan werden.

In Abb. 2 ist für das zu bearbeitende Problem ein _____ erstellt worden.

Programmablaufplan

2 In dem Programmablaufplan der Abb. 2 wird die Kartenart (Spalte 80 der Lochkarte) mit einer Konstanten verglichen. Diesem Vergleich folgt eine Abfrage mit bedingter Verzweigung.

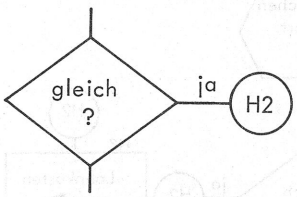
Suchen Sie bitte diese beiden Blöcke im Programmablaufplan. -

Wie nennt man diese Blöcke (Symbole der Zeichen-Schablone)?

Programm-Modifikation



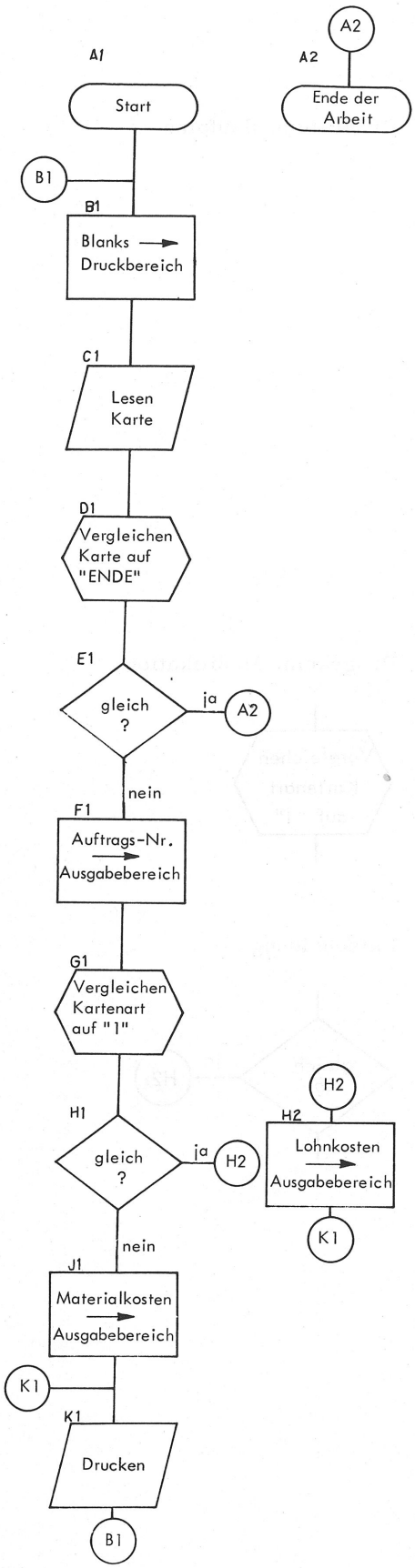
Entscheidung



Anmerkung:

Die Symbole des Programmablaufplans sind im Merkblatt 301 in Ihren Arbeitsmitteln zusammengestellt.

PROGRAMMABLAUFPLAN



PROGRAMM

Name	Operation	Operand und Bemerkungen
	START	1Ø,Ø,Ø,
ANFANG	BALR	8,Ø
	USING	*,8
*PROGRAMM	MABLAUF	
B1	MVC	ABER,KCBLANK
	GET	KARTE,EBER
	CLC	EBER(4),KCENDE
	BE	A2
	MVC	ANR,ENR
	CLC	EKR,KC1
	BE	H2
	MVC	AMRT-1(1Ø),SCHABL
	PACK	EMRT,EMRT
	ED	AMRT-1(1Ø),EMRT+3
K1	PUT	LISTE,ABER
	B	B1
H2	MVC	ALOHN-1(1Ø),SCHABL
	PACK	ELOHN,ELOHN
	ED	ALOHN-1(1Ø),ELOHN+3
	B	K1
A2	Ende der Arbeit	
*DEFINIT		

Abb. 3

3 Abb. 3 zeigt die Gegenüberstellung von Programmablaufplan und Programm. (Einzelheiten dieser Abbildung interessieren uns hier nicht.)

Im Programmablaufplan ist der Programmablauf schon bildhaft vorgegeben.

Der Programmablaufplan ist die Grundlage für das Schreiben eines _____.

Programms

4 Wir fassen zusammen:

Ein Problem der Datenverarbeitung wird zunächst in Einzelschritte aufgelöst: der _____ wird erstellt.

Als nächstes muß der Programmablaufplan in das _____ umgesetzt werden.

Die Funktionen und der Ablauf eines Programms sind durch den _____ schon festgelegt.

Programmablaufplan

Programm

Programmablaufplan

5 Der Programmablaufplan muß in eine Sprache übertragen werden, die eine Verständigung mit dem DVS ermöglicht.

Das Programm wird in einer s y m b o l i s c h e n _____ geschrieben.

Programmiersprache/Sprache

6 Eine Programmiersprache muß leicht erlernbar und gut zu handhaben sein.

Diese Forderungen erfüllt eine _____ Programmiersprache.

symbolische

7 In einem Programm müssen Anweisungen an das DVS gegeben werden, Felder und Bereiche sind festzulegen und Konstanten müssen angegeben werden.

All das läßt sich bequem und einfach in einer ___(WW)___ formulieren.

symbolischen Programmiersprache

8 Ein Programm, das in einer s y m b o l i s c h e n Programmiersprache geschrieben ist, bezeichnet man als ein _____ Programm.

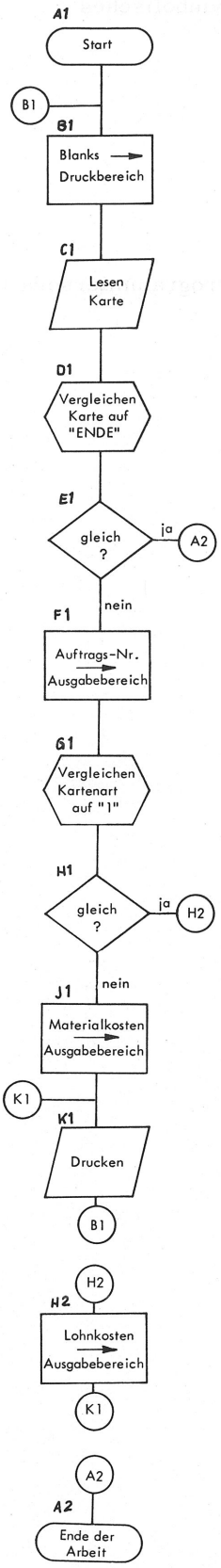
symbolisches

9 In diesem Lehrgang befassen wir uns mit dem Schreiben symbolischer Programme in der A s s e m b l e r s p r a c h e .

Die Assemblersprache ist eine symbolische _____.

Programmiersprache

DIE CODIERUNG DES PROGRAMMABLAUFPLANS IN DER ASSEMBLERSPRACHE



Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)
	START	1000
B1	MVC	ABER,KCBLANK
	GET	KARTE,EBER
	CLC	EBER(4),KCENDE
	BE	A2
	MVC	ANR,ENR
	CLC	EKA,KC1
	BE	H2
	MVC	AMAT-1(10),SCHABL
	PACK	EMAT,EMAT
	ED	AMAT-1(10),EMAT+3
K1	PUT	LISTE,ABER
	B	B1
H2	MVC	ALOHN-1(10),SCHABL
	PACK	ELOHN,ELOHN
	ED	ALOHN-1(10),ELOHN+3
	B	K1
A2	Ende der	Arbeit

Abb. 4

10 Betrachten Sie Abb. 4. -

Der Programmablaufplan wird Schritt für Schritt in ein _____ Programm übertragen. In diesem Lehrgang verwenden wir hierzu die _____-sprache.

symbolisches

Assembler

11 Das Übertragen des Programmablaufplans in die Assemblersprache bezeichnet man als das C o d i e r e n des Programmablaufplans.

Die C o d i e r u n g ist die Umsetzung eines _____ in ein ___(WW)___.

Programmablaufplans

symbolisches Programm

12 Wie heißt die symbolische Programmiersprache, in der die Codierung in Abb. 4 geschrieben ist?

Assemblersprache

13 Aus Abb. 4 ersehen Sie, daß ein Block des Programmablaufplans in eine oder mehrere A n w e i s u n g e n in der Assemblersprache übertragen wird.

Zwei Blöcke in Abb. 4 erfordern jeweils 3 Anweisungen. Es sind dies die Schritte _____ und _____.

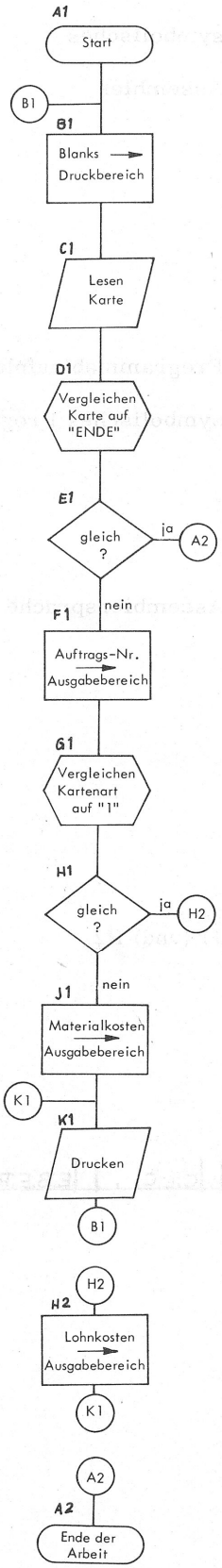
J1 (und) H2

14 Der Block D1 in Abb. 4 ist eine Programm-Modifikation, die eine Entscheidung vorbereitet.

Wie lautet für den Block D1 die Anweisung in der Assemblersprache?

| |C.L.C. , | |E.B.E.R.(,4.),,K.C.E.N.D.E. ,

DIE CODIERUNG DES PROGRAMMABLAUFPLANS IN DER ASSEMBLERSPRACHE



Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)
	START	1000
B1	MVC	ABER,KCBLANK
	GET	KARTE,EBER
	CLC	EBER(4),KCENDE
	BE	A2
	MVC	ANR,ENR
	CLC	EKA,KC1
	BE	H2
	MVC	AMAT-1(10),SCHABL
	PACK	EMAT,EMAT
	ED	AMAT-1(10),EMAT+3
K1	PUT	LISTE,ABER
	B	B1
H2	MVC	ALOHN-1(10),SCHABL
	PACK	ELOHN,ELOHN
	ED	ALOHN-1(10),ELOHN+3
	B	K1
A2	Ende der	Arbeit

Abb. 4

15 Eine Anweisung, in der der Programmierer sagt, was und mit welchen Daten etwas getan werden soll, bezeichnet man auch als eine Instruktion.

In den folgenden Lernelementen befassen wir uns etwas eingehender mit einer solchen _____.

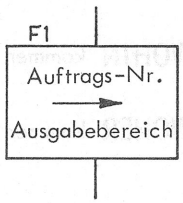
Instruktion

16 Abb. 4 zeigt:

Der Block F1 des Programmablaufplans wird in der Assemblersprache umgesetzt in die Instruktion _____.

MVC . | ANR, ENR . .

17



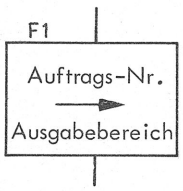
Die Eintragung im Block F1 bedeutet:

"Übertrage _____ nach _____".

Auftrags-Nr.

Ausgabebereich

18



MVC . | ANR, ENR .

Die Bedeutung des Blocks F1 und der Instruktion MVC ANR, ENR ist dieselbe.
Beide drücken aus:
"Übertrage Auftrags-Nr. nach Ausgabebereich".

In Block F1 wird durch die Eintragung gesagt, was und mit welchen Daten etwas getan werden soll. In der Assemblersprache geschieht dies durch die symbolischen Bezeichnungen ___(WW)___.

MVC, ANR (und) ENR

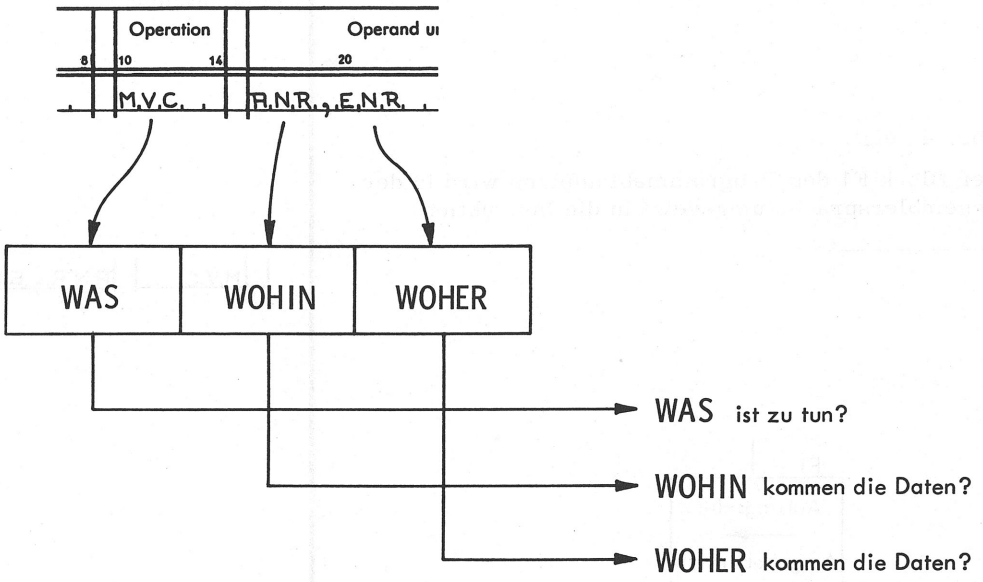

Operation
Operand u
8
10
14
20
M,V,C,
R,N,R,,E,N,R,
WAS
WOHIN
WOHER
WAS ist zu tun?
WOHIN kommen die Daten?
WOHER kommen die Daten?

Abb. 5

19 Betrachten Sie bitte Abb. 5. -

Wie Sie sehen, hat jede symbolische Bezeichnung der Instruktion eine besondere Funktion. Dem DVS wird mitgeteilt:

a) _____ zu tun ist,

b) _____ die Daten kommen,

c) _____ die Daten kommen.

Was

Wohin

Woher

20 Wir wenden uns zunächst der symbolischen Bezeichnung MVC in Abb. 5 zu.

MVC ist eine Abkürzung aus dem Englischen: MoVe Characters, das heißt soviel wie "Übertragen Zeichen".

MVC ist also eine symbolische Bezeichnung für die Operation "___(WW)___".

Übertragen Zeichen

21 Die symbolische Bezeichnung für die Operation, die durchgeführt werden soll, nennt man den symbolischen Operationsschlüssel.

MVC ist der symbolische _____ für die Operation "___(WW)___".

Operationsschlüssel

Übertragen Zeichen

22 Die symbolischen Operationsschlüssel sind für alle Operationen des Systems /360 festgelegt. Der symbolische Operationsschlüssel für eine bestimmte Operation kann also vom Programmierer (frei, nicht frei) _____ gewählt werden.

Für die Operation "Übertragen Zeichen" heißt der symbolische _____ immer _____.

nicht frei

Operationsschlüssel

MVC

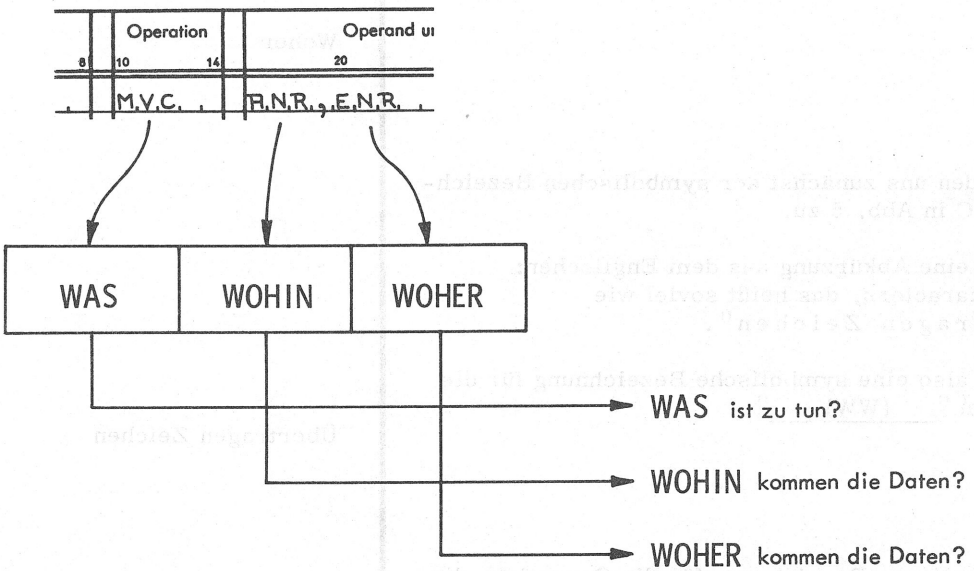

Operation
Operand u
8
10
14
20
M,V,C,
R,N,R,,E,N,R,
WAS
WOHIN
WOHER
WAS ist zu tun?
WOHIN kommen die Daten?
WOHER kommen die Daten?

Abb. 5

23 Die erste Frage in Abb. 5, das Was, wird durch den symbolischen Operationsschlüssel hinreichend beantwortet.

Durch den Operationsschlüssel MVC weiß das Datenverarbeitungssystem, daß Zeichen _____ werden sollen.

übertragen

24 Es sollen Zeichen (Daten) im Hauptspeicher des DVS von irgendwoher irgendwohin übertragen werden.

Dem DVS muß natürlich auch mitgeteilt werden, woher die Daten kommen und wohin die Daten übertragen werden sollen.

Das geschieht in Abb. 5 durch die symbolischen Bezeichnungen _____ und _____.

ANR (und) ENR

25 ANR bezeichnet die Stelle im Hauptspeicher, (wohin, woher) _____ die Daten kommen.

ENR bezeichnet die Stelle im Hauptspeicher, (wohin, woher) _____ die Daten kommen.

wohin

woher

26 ENR und ANR bezeichnet man auch als die Operanden in einer Instruktion.

Die symbolischen Namen dieser _____ sind frei wählbar.

Operanden

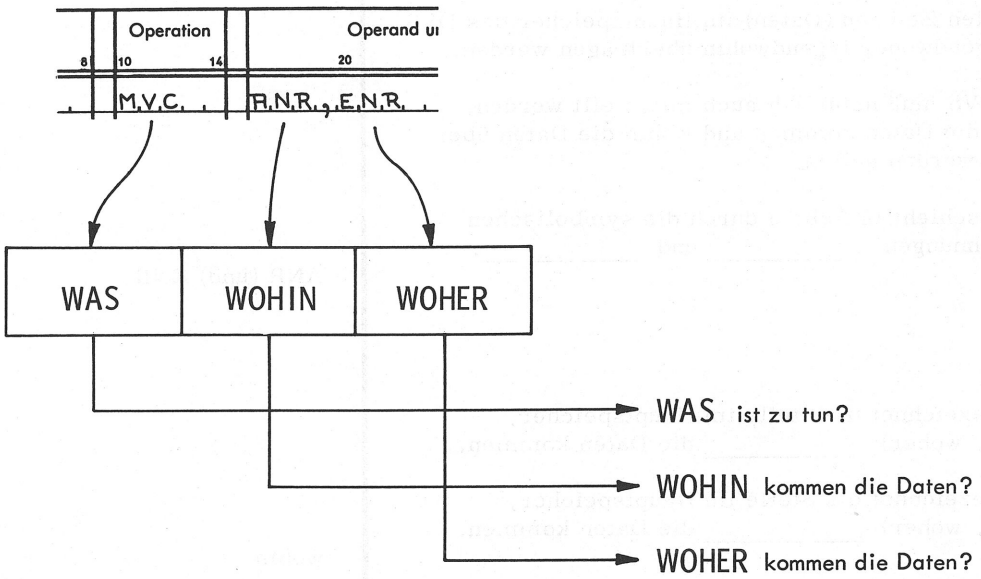

Operation
Operand u
8
10
14
20
M,V,C,
R,N,R,,E,N,R,
WAS
WOHIN
WOHER
WAS ist zu tun?
WOHIN kommen die Daten?
WOHER kommen die Daten?

Abb. 5

27 Wir fassen zusammen:

Jeder Block des Programmablaufplans ergibt eine oder mehrere _____ in der Assemblersprache.

Durch die Instruktion bringt der Programmierer zum Ausdruck, was und mit welchen Daten etwas geschehen soll:

Im Operationsschlüssel wird angegeben, _____ mit den Daten getan werden soll. Die Operanden geben an, _____ die verarbeiteten Daten im Hauptspeicher gebracht werden und _____ die zu verarbeitenden Daten im Hauptspeicher kommen.

Instruktionen/Anweisungen

was

wohin

woher

Ein Datenverarbeitungssystem versteht nur e i n e Sprache, nämlich die Sprache, die der Konzeption des Systems entspricht, die M a s c h i n e n s p r a c h e.

Assemblersprache:

AP SUMME,BETRAG

Maschinensprache:

FA54 8419 8089

Hier ist die Instruktion "Addiere Betrag zur Summe" sowohl in der Assemblersprache als auch in der Maschinensprache dargestellt.
Diese Gegenüberstellung macht deutlich, wie wenig aussagekräftig die Maschinensprache ist und wie sehr die symbolische Programmiersprache das Schreiben eines Programms vereinfacht.

Ein symbolisches Programm (ein Programm in symbolischer Sprache) kann jedoch nicht direkt vom System /360 durchgeführt werden. Denn das System /360 versteht nur die Maschinensprache.

Das symbolische Programm muß also in ein Programm in Maschinensprache u m g e w a n d e l t werden, damit die Instruktionen vom DVS verstanden werden können. Ein Programm, das in die Maschinensprache umgewandelt ist, bezeichnet man als M a s c h i n e n p r o g r a m m.

Wenn Sie bereits mit den Grundzügen der Umwandlung eines symbolischen Programms in ein Maschinenprogramm vertraut sind, dann fahren Sie bitte gleich mit LE 43 fort.

In LE 43 ff. wird die Umwandlung des symbolischen Programms in ein Maschinenprogramm noch einmal zusammenfassend beschrieben.

SYMBOLISCHES PROGRAMM

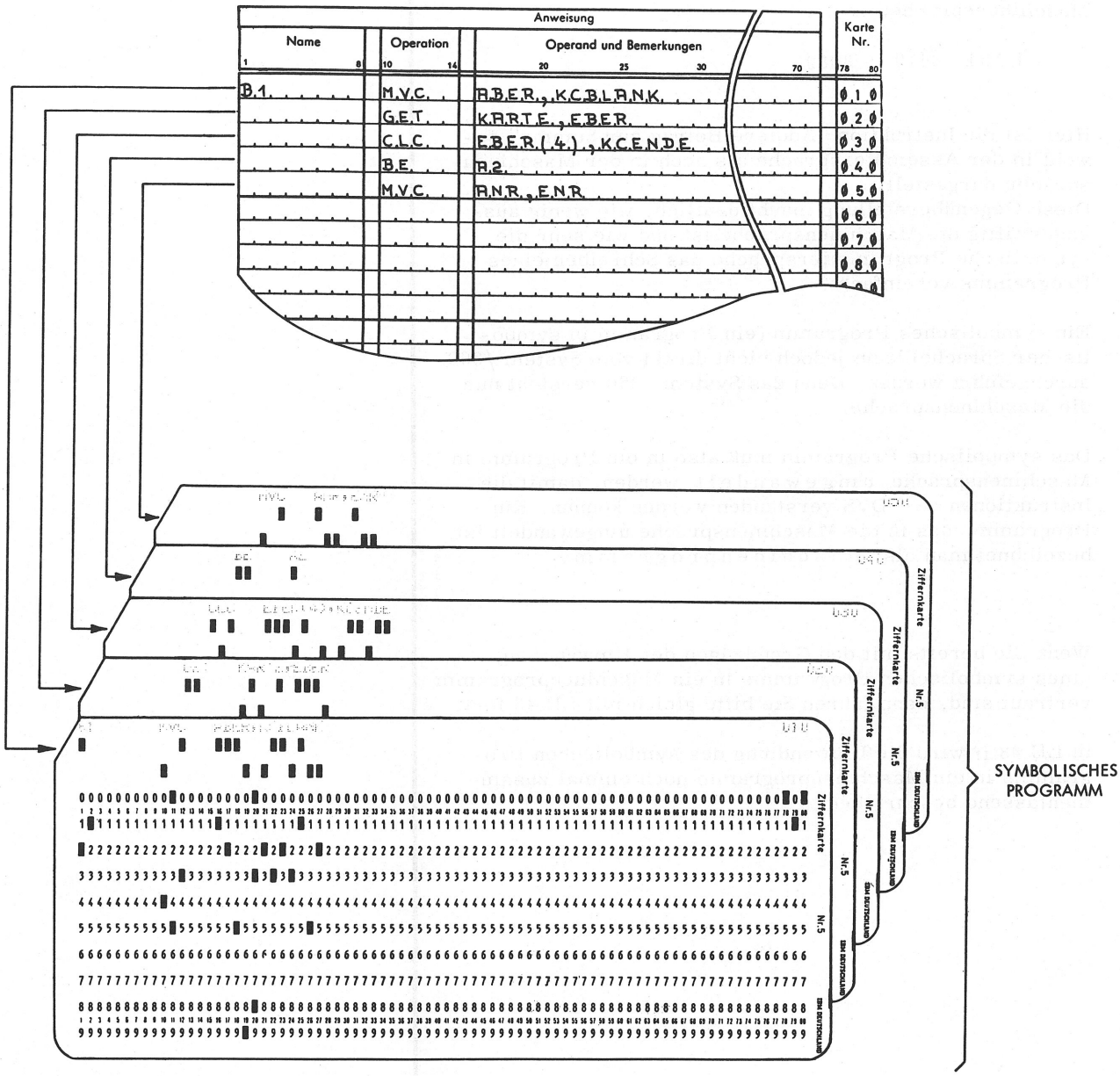


Abb. 6

28 Damit ein Programm vom DVS durchgeführt werden kann, muß es sich in einer Form befinden, die wir als Maschinensprache bezeichnen.

Die einzige Sprache, die das DVS versteht, ist die _____.

Ein symbolisches Programm (ein Programm in symbolischer Sprache) muß also in ein Programm in Maschinensprache _____ werden.

Maschinensprache

umgewandelt

29 Wenn man ein symbolisches Programm umwandelt, erhält man ein M a s c h i n e n p r o g r a m m .

Nur ein Programm in der Maschinensprache, also ein _____, kann vom System /360 durchgeführt werden.

Maschinenprogramm

In den folgenden Lernelementen werden Sie erfahren, wie man zu einem Maschinenprogramm gelangt. Wir befassen uns also mit der Umwandlung des symbolischen Programms in das Maschinenprogramm.

30 Aus Abb. 6 ersehen Sie, was mit der Codierung des Programmierers geschieht:

Jede Zeile des symbolischen Programms wird in eine _____ abgelocht.

Lochkarte

31 E i n e Anweisung in der Assemblersprache ergibt also _____ Lochkarte.

Die so erstellten Lochkarten bezeichnet man ebenfalls als ___(WW)___. (Siehe Abb. 6)

eine

symbolisches Programm

In der Literatur werden statt "symbolisches Programm" auch die Begriffe "Symbolprogramm", "Quellenprogramm" und "Ursprungsprogramm" verwendet. Alle diese Begriffe bedeuten dasselbe: ein Programm, geschrieben in einer symbolischen Programmiersprache.

DIE UMWANDLUNG DES SYMBOLISCHEN PROGRAMMS IN DAS MASCHINENPROGRAMM

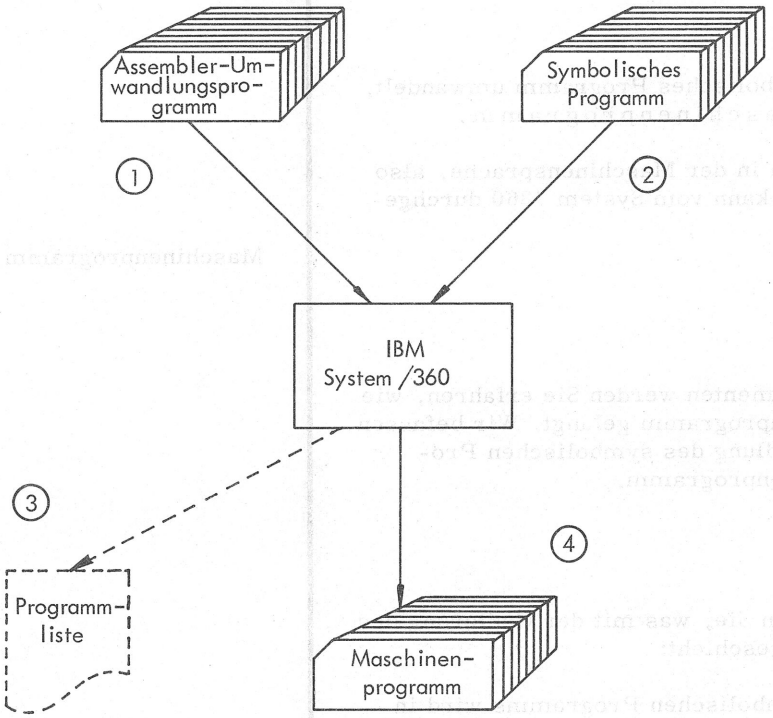


Abb. 7

32 Betrachten Sie bitte die schematische Darstellung einer Umwandlung in Abb. 7. -

Sie sehen, daß die Umwandlung eines symbolischen Programms in ein Maschinenprogramm mit Hilfe eines besonderen Umwandlungsprogramms durchgeführt wird, nämlich mit Hilfe des _____.

Assembler-Umwandlungsprogramms

33 Das Assembler-Umwandlungsprogramm, auch kurz Umwandlungsprogramm oder Übersetzungsprogramm genannt, wird von der IBM zur Verfügung gestellt.

"Assembler" ist ein Wort der englischen Sprache und heißt soviel wie Umwandler. Man nennt daher das Umwandlungsprogramm auch kurz und einfach den Assembler. -

Wir müssen zweierlei unterscheiden:

a) Assemblersprache und

b) Assembler-Umwandlungsprogramm.

Die Assemblersprache ist eine _____(WW)_____.

Das Assembler-Umwandlungsprogramm dient der _____ eines in der Assemblersprache geschriebenen _____(WW)_____ in ein _____.

symbolische Programmiersprache

Umwandlung

symbolischen Programms

Maschinenprogramm

34 Anhand der Abb. 7 verfolgen wir die einzelnen Schritte, die getan werden müssen, um die Umwandlung durchzuführen.

Als erstes (1) wird das _____ in den Hauptspeicher des _____ geladen.

Assembler-Umwandlungsprogramm

IBM Systems /360

DIE UMWANDLUNG DES SYMBOLISCHEN PROGRAMMS IN DAS MASCHINENPROGRAMM

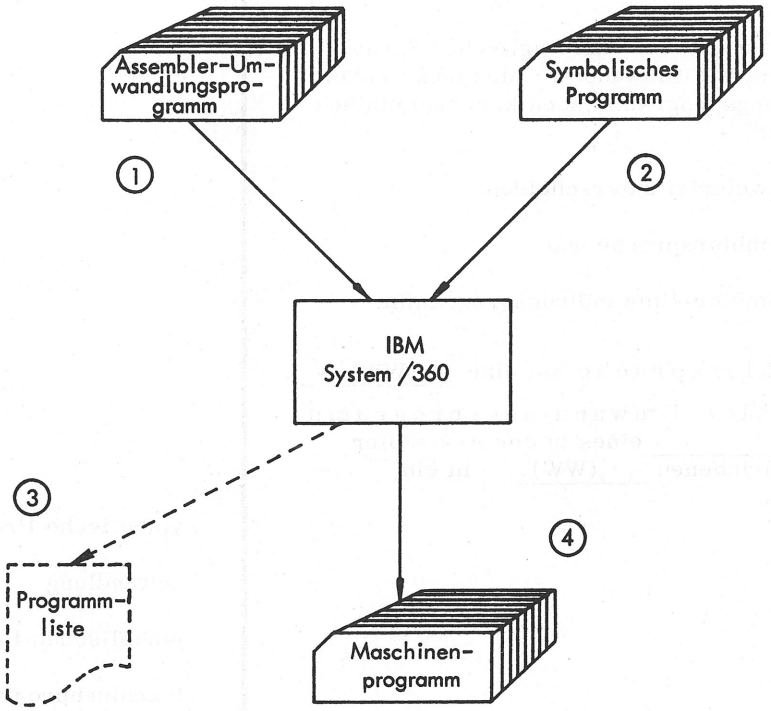


Abb. 7

35 Das System /360, in dem das Umwandlungsprogramm gespeichert ist, wandelt dann das ___(WW)___ (2) in das _____ (4) um.

symbolische Programm

Maschinenprogramm

36 Das Maschinenprogramm, das mit Hilfe des Assembler-Umwandlungsprogramms aus dem symbolischen Programm erstellt wird, steht in Lochkarten (4) für die Verarbeitung der Daten bereit.

Die Verarbeitung von Daten kann nur durch das _____ erfolgen.

Maschinenprogramm

Das Maschinenprogramm wird auch als "echtes Programm" oder "Objektprogramm" bezeichnet.

"Echtes Programm", "Maschinenprogramm" oder "Objektprogramm" sind Bezeichnungen, die dasselbe zum Ausdruck bringen: ein Programm in der Maschinensprache.

37 Die Umwandlung des symbolischen Programms in das Maschinenprogramm liefert noch ein Nebenprodukt. Das ist die _____(3), auch Umwandlungsliste genannt.

Die Programmliste ist ein Protokoll der Umwandlung. Sie enthält im wesentlichen die Instruktionen des symbolischen Programms und die des Maschinenprogramms. Außerdem gibt sie Hinweise auf formale Fehler im symbolischen Programm.

Programmliste

DIE UMWANDLUNG DES SYMBOLISCHEN PROGRAMMS IN DAS MASCHINENPROGRAMM

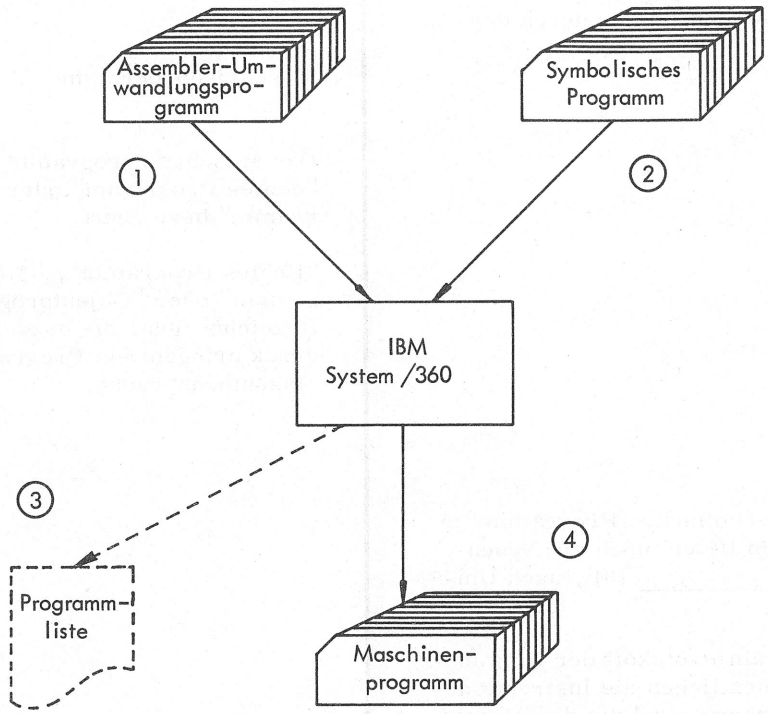


Abb. 7

38 Beantworten Sie bitte erst jede der folgenden Fragen, ehe Sie Ihre Antworten überprüfen. Benutzen Sie ein Extrablatt.

a) Welches Programm bewirkt die Umwandlung?

b) Wie nennt man das Produkt der Umwandlung?

c) Welches Programm wird umgewandelt?

a) Das Assembler-Umwandlungsprogramm

b) Das Maschinenprogramm/echte Programm/Objektprogramm

c) Das symbolische Programm/ Quellenprogramm/Ursprungsprogramm

DIE VERARBEITUNG DER DATEN DURCH DAS
MASCHINENPROGRAMM

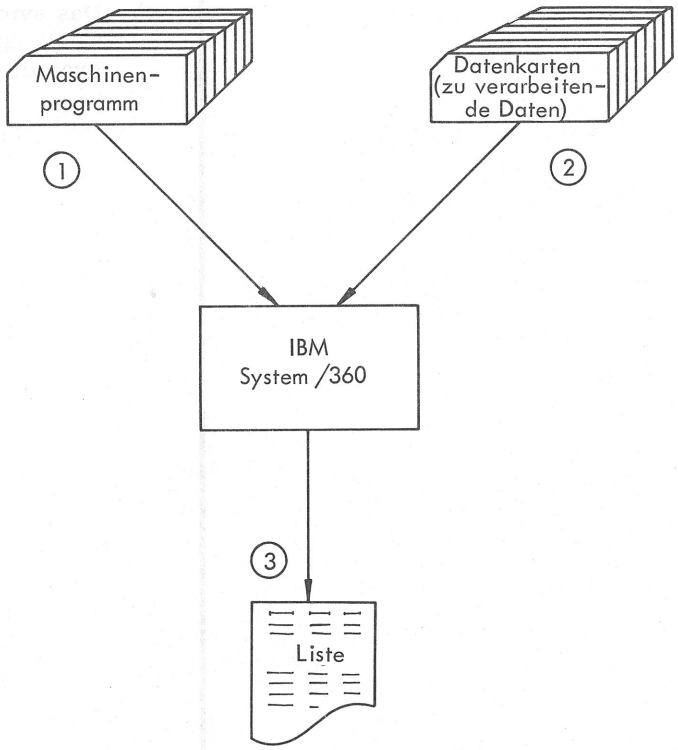


Abb. 8

39 Wir wenden uns jetzt der V e r a r b e i t u n g der Daten im System /360 durch das Maschinenprogramm zu. Betrachten Sie bitte Abb. 8.-

Als erstes (1) wird das _____ in den Hauptspeicher des _____ geladen.

Maschinenprogramm

IBM Systems /360

40 Das Maschinenprogramm, auch echtes Programm genannt, steuert die Verarbeitung der _____ (2) im System /360.

Datenkarten/Daten

41 In Abb. 8 bewirkt das Maschinenprogramm, daß aus den Datenkarten eine _____ (3) erstellt wird.

Liste

42 Beantworten Sie bitte erst beide Fragen, bevor Sie Ihre Antworten überprüfen.

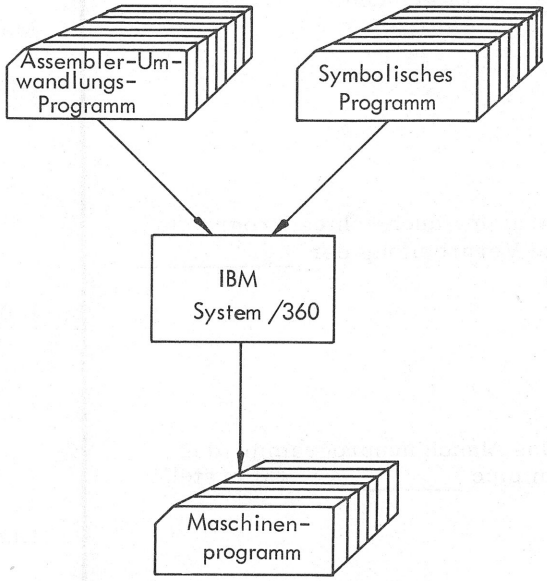
a) Welches Programm bewirkt die Verarbeitung der Daten?

b) Was geschieht mit dem Maschinenprogramm, bevor Daten verarbeitet werden?

a) Das Maschinenprogramm

b) Das Maschinenprogramm wird zuvor in den Hauptspeicher geladen (oder sinngemäß).

I UMWANDLUNG



II VERARBEITUNG

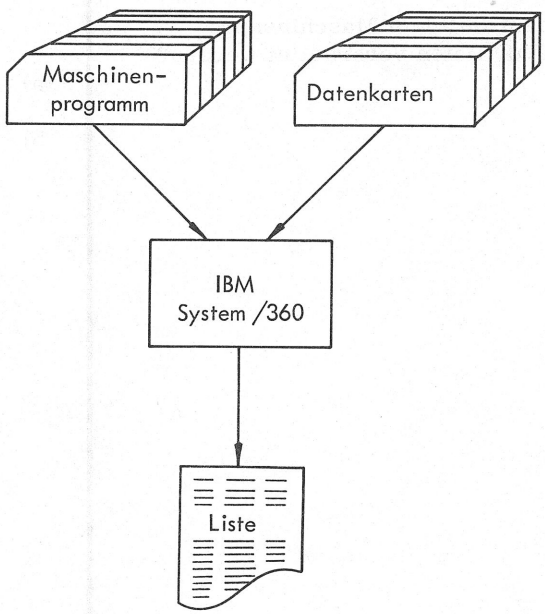


Abb. 9

43 In den vorhergehenden Lernelementen haben wir uns mit der Umwandlung des symbolischen Programms in das Maschinenprogramm und mit der Verarbeitung der Daten durch das Maschinenprogramm beschäftigt.

Anhand der Abb. 9 fassen wir zusammen:

Das in der Assemblersprache geschriebene Programm ist ein _____ Programm, das in Lochkarten abgelocht wird.

Das abgelochte ___(WW)___ wird in das _____ umgewandelt.

symbolisches

symbolische Programm

Maschinenprogramm

44 Die Umwandlung des symbolischen Programms in das Maschinenprogramm erfolgt im System /360 und wird durch das _____ gesteuert.

Assembler-Umwandlungsprogramm

45 Das Maschinenprogramm, auch echtes Programm genannt, ist ein Programm in Maschinensprache.

Damit Datenkarten verarbeitet werden können, muß zunächst das _____ in den _____ des Systems /360 geladen werden.

Maschinenprogramm

Hauptspeicher

46 Die Verarbeitung der Daten im System /360 kann nur durch das im Hauptspeicher befindliche _____ erfolgen.

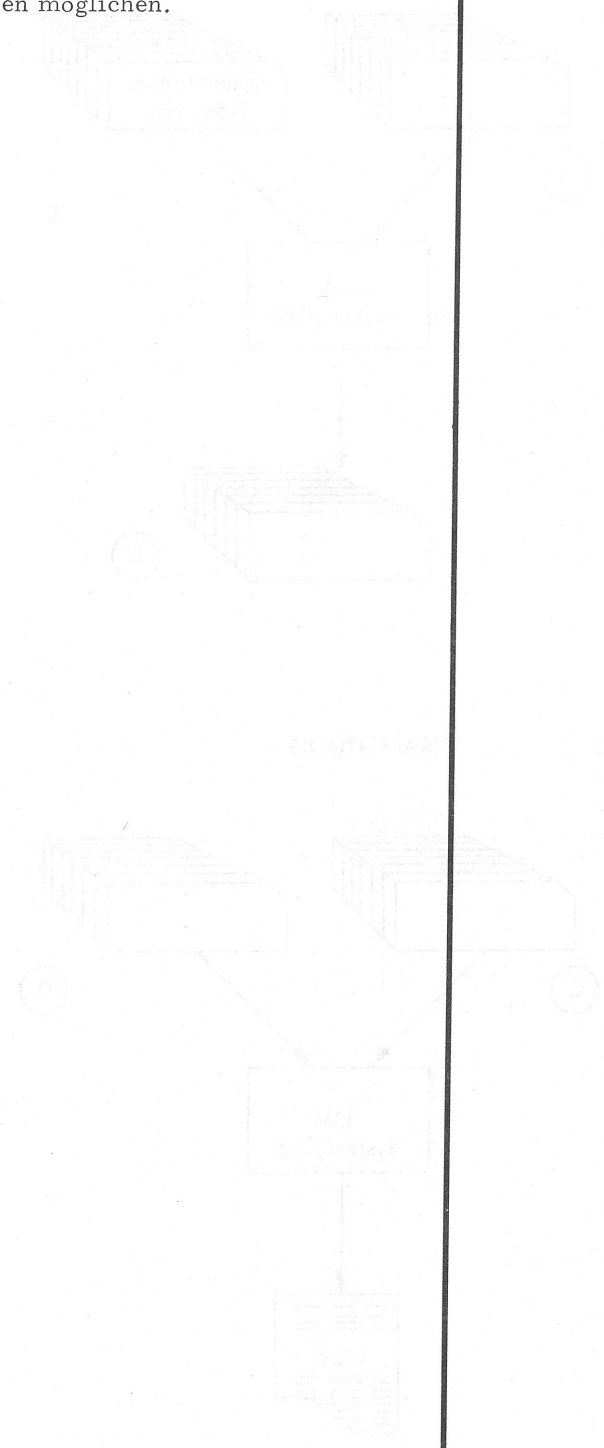
Maschinenprogramm

47 Ist das Maschinenprogramm durch die Umwandlung einmal erstellt, kann es immer wieder für die Verarbeitung der Daten verwendet werden.

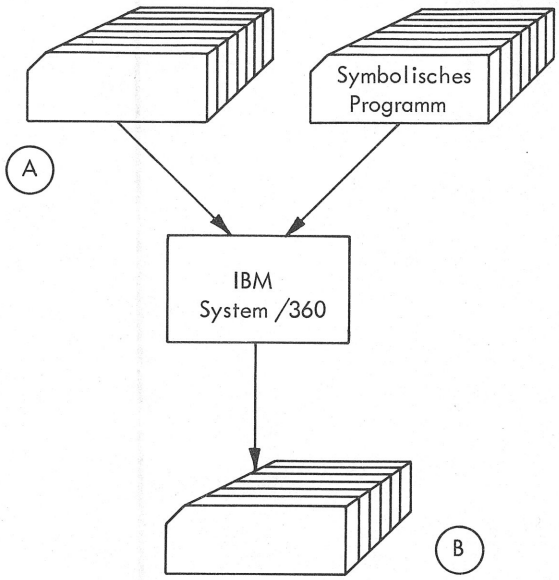
Das Maschinenprogramm kann beliebig oft für die Steuerung einer Datenverarbeitung in den _____ des Systems /360 geladen werden.

Hauptspeicher

Sie wissen nun, daß ein symbolisches Programm mit Hilfe des Assembler-Umwandlungsprogramms in ein Maschinenprogramm umgewandelt werden muß, ehe Datenkarten verarbeitet werden können. Der hier aufgezeigte Gang der Umwandlung und anschließenden Verarbeitung ist nur einer von mehreren möglichen.



UMWANDLUNG



VERARBEITUNG

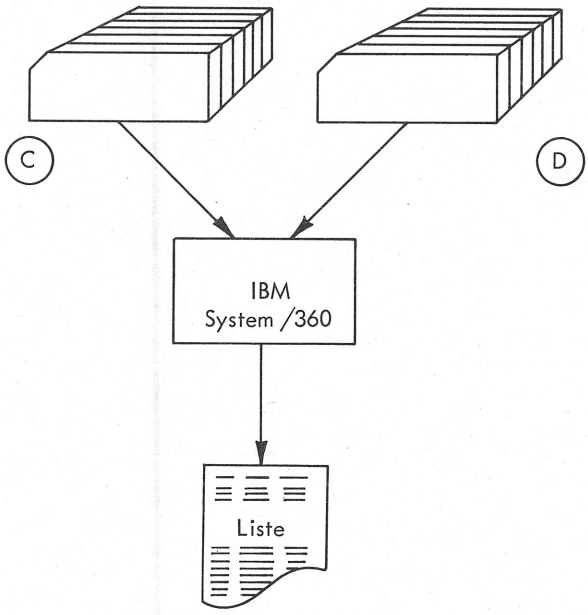


Abb. 10

FRAGEN ZUM KAPITEL I

Anhand der folgenden Fragen sollen Sie nun prüfen, ob Sie den Lernstoff dieses Kapitels verstanden haben.

Beantworten Sie erst alle Fragen, ehe Sie Ihre Lösungen mit den Antworten auf der folgenden Seite vergleichen.
Benutzen Sie ein Extrablatt!

1. Die Codierung ist die Umsetzung eines _____ in ein __(WW)_____.

2. Durch den symbolischen Operationsschlüssel wird dem DVS mitgeteilt, _____ zu tun ist. Die Bezeichnung für den Operationsschlüssel ist (festgelegt, frei wählbar) _____.

3. Die Operanden bezeichnen die Stellen im Hauptspeicher, _____ die Daten übertragen werden sollen und _____ die Daten kommen. Die symbolischen Namen der Operanden sind (festgelegt, frei wählbar) _____.

4. Ergänzen Sie bitte die fehlenden Bezeichnungen der Abb. 10.

   A _____

   B _____

   C _____

   D _____

ANTWORTEN ZUM 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. Die Codierung ist die Umsetzung eines Programmablaufplans in ein symbolisches Programm.

   (LE 10 - 12)

2. Durch den symbolischen Operationsschlüssel wird dem DVS mitgeteilt, was zu tun ist. Die Bezeichnung für den Operationsschlüssel ist festgelegt.

   (LE 20 - 23)

3. Die Operanden bezeichnen die Stellen im Hauptspeicher, wohin die Daten übertragen werden sollen und woher die Daten kommen. Die symbolischen Namen der Operanden sind frei wählbar.

   (LE 24 - 27)

4. A Assembler-Umwandlungsprogramm

   B Maschinenprogramm

   C Maschinenprogramm

   D Datenkarten

   (LE 32 - 42)

Schauen Sie sich nun Merkblatt 302 an. -



Band 2, Seite 5 (Kapitel I)

KAPITEL II

DAS PROGRAMMSCHEMA

Lernstoff:

Die Anweisungsfelder: Operation, Operand, Name

Die Regeln für Symbole

Die Bemerkungen

Sonstige Eintragungen

Wir befassen uns in diesem Lehrgang mit der Assemblersprache. Wenn man ein Programm in der Assemblersprache schreibt, dann müssen einige f o r m a l e R e g e l n eingehalten werden. Werden diese Regeln nicht beachtet, dann ist eine fehlerfreie Umwandlung des symbolischen Programms in das Maschinenprogramm nicht möglich.

Es steht uns ein Hilfsmittel zur Verfügung, das die Einhaltung der formalen Regeln beim Schreiben eines Programms erleichtert. Dieses Hilfsmittel ist das P r o g r a m m s c h e m a.

Wir befassen uns nun mit diesem Programmschema und den formalen Regeln, die bei seiner Benutzung zu beachten sind.

1 Entnehmen Sie Ihren Arbeitsmitteln ein Programmschema (IBM Form 76 682). -

Das Programmschema ist eine Ablochvorlage, die das Ablochen des symbolischen Programms in Lochkarten erleichtert. Die einzelnen Zeilen des Programmschemas sind in sich von links nach rechts durchnumeriert.

Die einzelnen Stellen einer Zeile entsprechen den _____ einer Lochkarte.

Spalten

2 Pro Zeile ist nur e i n e Anweisung erlaubt, jedoch dürfen nach Belieben Zeilen freigelassen werden.

Jede benutzte Zeile des Programmschemas ergibt beim Ablochen des symbolischen Programms _(ZW)_.

1 Lochkarte

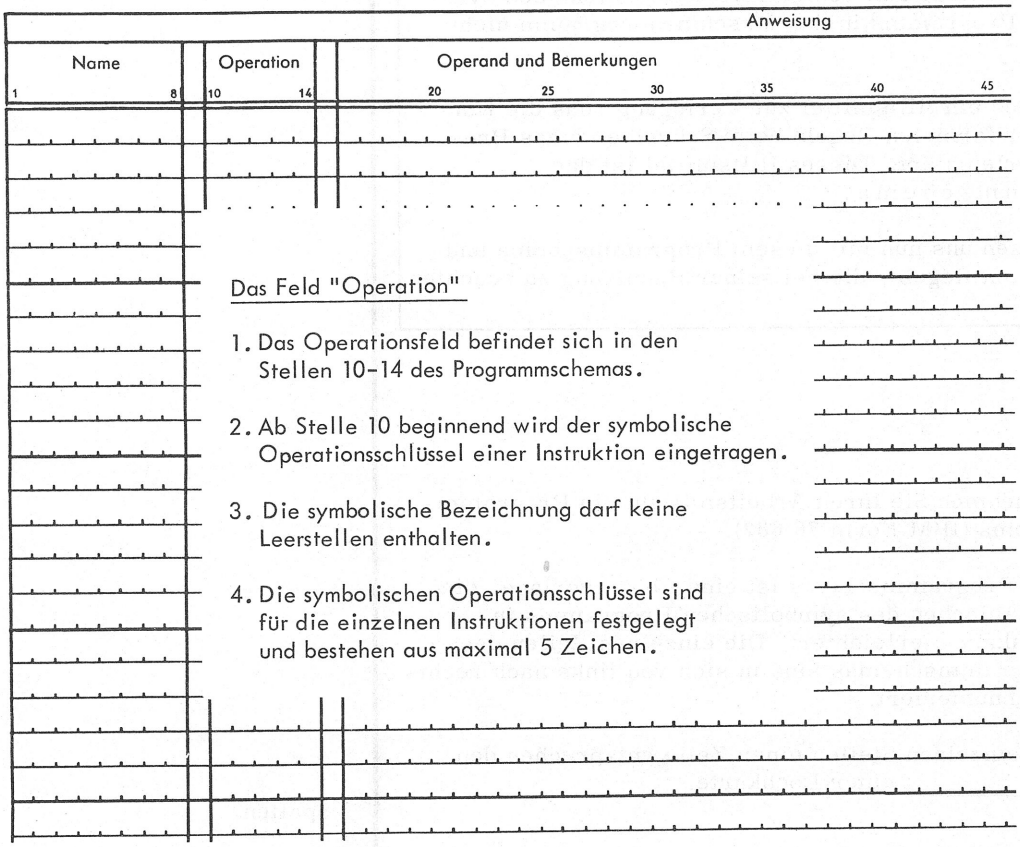

Anweisung
Name
1
8
Operation
10
14
Operand und Bemerkungen
20
25
30
35
40
45
Das Feld "Operation"
1. Das Operationsfeld befindet sich in den Stellen 10-14 des Programmschemas.
2. Ab Stelle 10 beginnend wird der symbolische Operationsschlüssel einer Instruktion eingetragen.
3. Die symbolische Bezeichnung darf keine Leerstellen enthalten.
4. Die symbolischen Operationsschlüssel sind für die einzelnen Instruktionen festgelegt und bestehen aus maximal 5 Zeichen.

Abb. 1

3 Wir befassen uns zunächst mit den 3 Anweisungsfeldern des Programmschemas, die deutlich gegeneinander abgehoben sind, nämlich "Name", "Operation" und "Operand und Bemerkungen".

Lesen Sie bitte den Text in Abb. 1 aufmerksam durch. -

Der symbolische Operationsschlüssel wird im Programmschema in die Stellen __(Z)__ bis __(Z)__ eingetragen.

Das bedeutet, daß der Operationsschlüssel im abgelochten Programm in den Spalten __(Z)__ bis __(Z)__ der Lochkarte steht.

10 (bis) 14

10 (bis) 14

4 Der symbolische Operationsschlüssel wird ab der Stelle __(Z)__ des Programmschemas eingetragen und kann maximal aus __(Z)__ Zeichen bestehen.

10

5

	Name	Operation	Operc
	1 8	10 14	20
a)		C,L,C,	
b)		P A,C,K,	
c)		P,A,C,K,	
d)		A,P,	
e)		A P	
f)		M,V,C,	
g)		M, ,V,C,	
h)		E,N,D,	

Abb. 2

	Anweisung
Name	Operation / Operand und Bemerkungen
1 8	10 14 / 20 25 30 35 40

Das Operandenfeld

1. Das Operandenfeld beginnt mit der Stelle 16 im Programmschema.

2. Abhängig von der Instruktion sind im Operandenfeld ein oder mehrere Operanden möglich.

3. Mehrere Operanden werden durch Kommas voneinander getrennt.

4. Leerstellen zwischen den Operanden sind nicht erlaubt.

Abb. 3

5 Betrachten Sie bitte Abb. 2. -

Ein Anfänger hat in Abb. 2 versucht, symbolische Operationsschlüssel den Regeln entsprechend in das Programmschema einzutragen. Wie Sie sicherlich festgestellt haben, ist das in 3 Fällen nicht ganz geglückt.

Bezeichnen Sie bitte die Fälle und begründen Sie, warum die jeweilige Eintragung nicht vorschriftsmäßig ist.

Sinngemäß:

b) Operationsschlüssel PACK ab Stelle 9

e) Operationsschlüssel AP ab Stelle 14

g) Leerstelle im Operationsschlüssel MVC

6 Lesen Sie bitte zunächst den Text in Abb. 3. -

Die Operanden werden im Programmschema ab der Stelle __(Z)__ eingetragen.

Je nach Art der Instruktion können im Operandenfeld ein oder __(WW)__ erforderlich sein.

16

mehrere Operanden

7 Sind mehrere Operanden im Operandenfeld erforderlich, so müssen die Operanden durch __(WW)__.

Zwischen den Operanden sind keine _____ erlaubt.

Kommas voneinander getrennt werden

Leerstellen

	Name (1 … 8)	Operation (10 … 14)	Operand und Bemerkungen (20 … 25 … 30)
a)		C,L,C,	E,N,R,,K,O,N,S,T,
b)		P,R,C,K,	B,E,T,R,A,G,,B,E,T,R,A,G,
c)		P,R,C,K,K	O S,T,E,N,,K,O,S,T,E,N,
d)		A,P,	S,U,M,M,E,,B,E,T,R,A,G,
e)		A,P,	G,E,K,O,S,T,E,N, K,O,S,T,E,N,
f)		M,V,C,	A,M,E,N,G,,M,E,N,G,E,N,
g)		M,V,C,	A,P,R,E,I,S,, P,R,E,I,S,E,
h)		E,N,D,	A,N,F,A,N,G,

Abb. 4

8 Betrachten Sie bitte Abb. 4. -

Einige Operanden sind nicht ordnungsgemäß in das Programmschema eingetragen. In welchen Fällen sind die Eintragungen nicht vorschriftsmäßig und warum sind sie jeweils nicht vorschriftsmäßig?

Sinngemäß:

c) Operanden ab Stelle 14

e) Fehlendes Komma

g) Leerstelle zwischen den Operanden

Anweisung		
Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30 35 40 45

Das Feld "Name"

Dieses Feld dient dazu, einer Anweisung einen symbolischen Namen zuzuordnen. Hat eine Anweisung einen Namen erhalten, so kann sie immer wieder angesprochen werden.

1. Das Feld "Name" umfaßt die Stellen 1 - 8.
2. Es wird nur benutzt, wenn einer Anweisung ein symbolischer Name zugeordnet wird.
3. Ein symbolischer Name muß in der Stelle 1 des Programmschemas beginnen und darf maximal 8 Zeichen lang sein.

Beispiele:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	:	
WEITER	PACK	BETRAG,BETRAG
	PACK	KOSTEN,KOSTEN
	AP	SUMME,BETRAG
	AP	GEKOSTEN,KOSTEN
B2	MVC	AMENGE,MENGEN
	MVC	APREIS,PREISE
	:	
	B	WEITER

Abb. 5

9 Lesen Sie bitte zunächst den Text in Abb. 5. -

In der Codierung sind zwei symbolische Namen im Namenfeld eingetragen: _____ und _____.

WEITER

B2

10 Man kann auf eine Anweisung, die einen symbolischen Namen erhalten hat, im Programm immer wieder Bezug nehmen.

In einem solchen Fall taucht der Name der Anweisung als Operand auf (siehe Abb. 5).

In der Anweisung B WEITER wird Bezug genommen auf die Anweisung _____.

WEITER	PACK	BETRAG, BETRAG

11 Ein symbolischer Name für eine Anweisung muß immer in der Stelle __(Z)__ des Programmschemas beginnen und darf maximal __(Z)__ Zeichen lang sein.

1

8

Name (1 … 8)	Operation (10 … 14)	Operand und Bemerkungen (20 … 25 … 30)
	:	
WEITER	PRCK	BETRAG,BETRAG
	PRCK	KOSTEN,KOSTEN
	AP	SUMME,BETRAG
	AP	GEKOSTEN,KOSTEN
B2	MVC	AMENG,MENGEN
	MVC	APREIS,PREISE
	:	
	B	WEITER

Abb. 6

Die Symbole

Wie Sie wissen, sind die symbolischen Bezeichnungen für die einzelnen Operationen festgelegt. Für jede Operation gibt es einen bestimmten symbolischen Operationsschlüssel (z.B. PACK, AP, MVC in der Codierung der Abb. 6).

Anders verhält es sich mit den symbolischen Bezeichnungen für das Operanden- und Namenfeld. Diese symbolischen Bezeichnungen, auch kurz Symbole genannt, sind frei wählbar (z. B. WEITER, KOSTEN, MENGEN in der Codierung der Abb. 6).

Symbole (symbolische Bezeichnungen) werden vom Programmierer festgelegt. Sie sollen aussagekräftig sein, d.h. der symbolische Name soll die Sache, für die er vergeben wurde, deutlich erkennen lassen (z.B. GESKOST für "Gesamtkosten"). Durch die Symbole werden Adressen von Feldern, Bereichen und Instruktionen im Hauptspeicher symbolisch benannt und angesprochen.

Einige wenige Regeln sind bei der Vergabe symbolischer Namen zu beachten:

1. Ein Symbol kann 1 bis 8 Zeichen lang sein.

2. Folgende Zeichen dürfen verwendet werden:

   a) die Buchstaben A-Z

   b) die Ziffern 0-9

   c) die 3 Zeichen $, #, @

3. Das erste Zeichen eines Symbols darf keine Ziffer sein.

4. Leerstellen sind innerhalb eines Symbols unzulässig.

Die Symbole

1. Ein Symbol kann 1 bis 8 Zeichen lang sein.
2. Folgende Zeichen dürfen verwendet werden:
   a) die Buchstaben A - Z
   b) die Ziffern 0 - 9
   c) die 3 Zeichen $, #, @
3. Das erste Zeichen eines Symbols darf keine Ziffer sein.
4. Leerstellen sind innerhalb eines Symbols unzulässig.

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
	:	
WEITER	PRCK	BETRAG,BETRAG
	PRCK	KOSTEN,KOSTEN
	AP	SUMME,BETRAG
	AP	GEKOSTEN,KOSTEN
B2	MVC	AMENG,MENGEN
	MVC	APREIS,PREISE
	:	
	B	WEITER

Abb. 7

12 Gültige Symbole sind z.B.:

a) KTONR. . .
b) RBT4567. .
c) N. . . . . .
d) R2. . . . . .
e) BEREICHE

Wie Sie aus den Beispielen b) und d) ersehen, kann sich eine symbolische Bezeichnung aus _____ und _____ zusammensetzen. Das erste Zeichen muß (abgesehen von $, #, @) immer ein _____ sein.

In Beispiel e) ist die maximal zulässige Länge eines Symbols voll ausgenutzt, denn Symbole dürfen maximal __(Z)__ Zeichen lang sein.

Buchstaben

Ziffern

Buchstabe

8

13 a) BETR. 48. . . .
b) 2758ETR. . . .
c) PERSONRLNR.
d) PERS..NR. . . .

Alle diese Symbole sind ungültig!

Begründen Sie bitte für jedes einzelne Symbol, warum es gegen die Regeln für Symbole verstößt. Notieren Sie erst alle Ihre Begründungen auf einem Extrablatt, bevor Sie Ihre Ergebnisse überprüfen.

Sinngemäß:

a) Leerstelle innerhalb des Symbols

b) Ziffer als erstes Zeichen

c) Mehr als 8 Zeichen

d) Punkt innerhalb des Symbols

Anweisung				Karte Nr.
Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)		(78–80)
*DER STERN IN	STELLE 1 KENNZEICHNET EINE BEMERKUNGSZEILE.			010
				020
* BEMERKUNGEN	DUERFEN NUR BIS STELLE 71 REICHEN UND KOENNEN IN DER FOL-			030
* GENDEN	ZEILE FORTGESETZT WERDEN. JEDE BENOETIGTE ZEILE IST HIERBEI			040
* DURCH EINEN	STERN ZU KENNZEICHNEN.			050
				060
*BEMERKUNGEN KOENNEN	AUCH RECHTS VON DEN OPERANDEN GESCHRIEBEN WERDEN,			070
*SIE MUESSEN	JEDOCH MINDESTENS DURCH 1 LEERSTELLE VON DIESEN GETRENNT			080
*SEIN. DIE	BEMERKUNGEN RECHTS VON DEN OPERANDEN DUERFEN KEINEN * IN			090
*STELLE 1	ERHALTEN. ZUM BEISPIEL:			100
				110
	CLC	ENR,KCENDE (VERGLEICH MIT KONSTANTE "ENDE")		120
A1	MVC	FELD1,FELD2 UEBERTRAGEN		130
				140
*ALLE ZEICHEN,	ALSO BUCHSTABEN,, ZIFFERN UND SONDERZEICHEN, DUERFEN FUER			150
*BEMERKUNGEN BENUTZT WERDEN. AUCH LEERSTELLEN SIND ERLAUBT.				160
				170
* WENDEN	SIE SICH NUN BITTE LERNELEMENT 14 ZU!			180
				190

Abb. 8

Die Bemerkungen

Die Bemerkungen sind nur für den Programmierer bestimmt. Sie dienen zur näheren Erklärung einzelner Anweisungen oder zur Bezeichnung ganzer Programmabschnitte. Die Bemerkungen haben keinerlei Einfluß auf das Programm selbst, weder auf die Umwandlung noch auf das umgewandelte Programm.

Die Bemerkungen erscheinen lediglich auf der Programmliste (auch Umwandlungsliste genannt), die bei der Umwandlung des symbolischen Programms erstellt wird.

Betrachten Sie bitte Abb. 8 und lesen Sie die Bemerkungen in dieser Abbildung.

14 Jede Bemerkungszeile muß durch einen _____ in der Stelle __(Z)__ des Programmschemas gekennzeichnet sein.

Stern

1

15 Eine Bemerkung darf bis Stelle __(Z)__ im Programmschema reichen und kann in der nächsten Zeile fortgesetzt werden.

Auch für die Fortsetzung der Bemerkungen in der folgenden Zeile ist ein _____ erforderlich.

71

Stern

	Anweisung			Karte Nr.
	Name (1–9)	Operation (10–14)	Operand und Bemerkungen (20–70)	(78–80)
a)	PROGRAMM ZUR		ERSTELLUNG DER LISTE 'ZUGRENGE'.	010
				020
b)	B2	AP	SUMME, BETRAGBEGINN EINES NEUEN PROGRAMMABSCHNITTS INNER-	030
c)	HALB PROGRAMM	A	1.8.	040
				050
d)	* ENDE	DES	PROGRAMMS	060
				070

Abb. 9

16 Was ist zu beachten, wenn eine Bemerkung rechts von den Operanden geschrieben werden soll?

Sinngemäß:

Zwischen Operand(en) und Bemerkung muß sich mindestens 1 Leerstelle befinden.

Kein Stern in Stelle 1!

17 Bezeichnen Sie bitte die Zeilen in Abb. 9, in denen Fehler auftreten.

Was wurde falsch gemacht?
(Benutzen Sie ein Extrablatt.)

Sinngemäß:

a) Fehlender Stern in Stelle 1

b) Keine Leerstelle zwischen Operand und Bemerkung

c) Fehlender Stern in Fortsetzungszeile

d) Stern nicht in Stelle 1

IBM System / 360 Assembler
Programmschema

Programm Nr. 15		Lochhinweise		Problem L 1 5 (73 75)
Erstellen der Liste Lohn - und Materialkosten		Zeichen	' *	
Programmierer Alfons Hermes	Datum 10.3.68	Lochung	5-8 1 1-4-8	Seite Ø 1 (76 77) von 7

Anweisung			Karte Nr.
Name (1 8)	Operation (10 14)	Operand und Bemerkungen (20 25 30 35 40 45 50 55 60 65 70)	(78 80)
			Ø1Ø
			Ø2Ø
			Ø3Ø
			Ø4Ø
			Ø5Ø
			Ø6Ø
			Ø7Ø
			Ø8Ø
			Ø9Ø
			1ØØ
			11Ø
			12Ø
			13Ø
			14Ø
			15Ø
			16Ø
			17Ø
			18Ø
			19Ø
			2ØØ

Abb. 10

Der Kopf des Programmschemas und die Numerierung der Zeilen

Im Kopf des Programmschemas ist Raum für Eintragungen des Programmierers freigehalten.

Betrachten Sie bitte Abb. 10. -

Wie Sie aus der Abbildung ersehen, können im Kopf des Programmschemas Angaben zum Programm, zum Namen des Programmierers und das Erstellungsdatum des Programms eingetragen werden. Auch kann der Programmierer für das Ablochen des Programms besondere Hinweise geben (z.B. die Lochung einzelner Zeichen). Diese Angaben werden nicht abgelocht.

Anders hingegen verhält es sich mit "Problem" und "Seite" in den Stellen 73-75 und 76-77.
Diese Angaben sind für alle Zeilen eines Programmschemas gleich und werden auch für jede Zeile abgelocht.

Die Numerierung der Zeilen in den Spalten 78-80 des Programmschemas ergibt beim Ablochen des symbolischen Programmes eine Durchnumerierung der Karten. Für 20 Zeilen ist diese Numerierung im Programmschema schon vorgegeben. Bei Benutzung der 3 restlichen Zeilen kann die Numerierung vom Programmierer selbst durchgeführt werden.

Die Angaben in den Stellen 73-80 des Programmschemas beeinflussen nicht die Umwandlung, sie erscheinen lediglich auf der Programmliste, wenn diese bei der Umwandlung erstellt wird.

IBM System / 360 Assembler
Programmschema

Programm Nr. 15		Lochhinweise	Problem L 1 5 (73 75)
Erstellen der Liste Lohn - und Materialkosten		Zeichen) *	
Programmierer Alfons Hermes	Datum 10.3.68	Lochung 5-8 1 1-4 8	Seite Ø 1 (76 77) von 7

Anweisung			Karte Nr.
Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20 25 30 35 40 45 50 55 60 65 70)	(78–80)
			Ø1Ø
			Ø2Ø
			Ø3Ø
			Ø4Ø
			Ø5Ø
			Ø6Ø
			Ø7Ø
			Ø8Ø
			Ø9Ø
			1ØØ
			11Ø
			12Ø
			13Ø
			14Ø
			15Ø
			16Ø
			17Ø
			18Ø
			19Ø
			2ØØ

Abb. 10

18 Wir halten fest:

Die Angaben zum Programm und zum Namen des Programmierers, die im Kopf des Programmschemas eingetragen werden können, werden nicht _____.

Unter "Lochhinweise" können Zeichen und deren _____ eingetragen werden.

abgelocht

Lochungen

19 Die Eintragungen, die in den Stellen 73-75 und 76-77 vorgenommen werden können, dienen zur besseren Überschaubarkeit bei einer großen Anzahl von Programmen sowie bei sehr umfangreichen Programmen.

Die Angaben in den Stellen 73-77 und die Numerierung in den Stellen 78-80 (beeinflussen, beeinflussen nicht) _____ die Umwandlung des symbolischen Programms.

Diese Angaben erscheinen bei der Umwandlung lediglich auf der _____.

beeinflussen nicht

Programmliste/Umwandlungsliste

IBM System / 360 Assembler
Programmschema

Programm Nr. 13		Lochhinweise	Problem R13 (73–75)
Kostenrechnung		Zeichen	
Programmierer Edi Pulcher	Datum 4.4.68	Lochung	Seite 05 (76 77) von 5

Name (1–8)	Operation (10–14)	Anweisung: Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
A2	MVC	ANR,ENR	010
	AP	SUMME,BETRAG	020
	AP	GESKOST,KOSTEN	030
	:		040
	:		050
	:		060
	:		070
	:		080
	:		090
	:		100
	:		110
	:		120
	:		130
	CLC	FELDA,KONST100	140
	BH	A2	150
	END	ANFANG	160
			170
			180
			190
			200
	PACK	BETRAG,BETRAG	011
	PACK	KOSTEN,KOSTEN	012

Abb. 11

20 Zu den Stellen 78-80 im Programmschema, die der Numerierung der Zeilen (Lochkarten) dienen, bleibt folgendes zu beachten:

An der Numerierung, die schon vorgegeben ist, fällt zunächst auf, daß alle Nullen durchgestrichen sind. - Damit die Null und das "O" für das Ablochen deutlich voneinander unterschieden sind, hat man festgelegt, alle Nullen durchzustreichen (z. B. KON1ØØ).

Die Numerierung der Zeilen erfolgt nicht fortlaufend mit 1 beginnend, die Zeilen sind vielmehr mit 10, 20, 30, 40 usw. bis 200 durchnumeriert. Durch diese Art der Numerierung können noch nachträglich Anweisungen quasi "zwischen den Zeilen" eingefügt werden.

Betrachten Sie bitte Abb. 11. -

Der Programmierer bemerkte erst nach der Fertigstellung des Programms, daß er zwischen den Anweisungen in den Zeilen 10 und 20 zwei Instruktionen versehentlich nicht eingetragen hat. Ohne deswegen die ganze Seite neu schreiben zu müssen, kann er nachträglich noch die Instruktionen _____ (011) und _____ (012) in das Programmschema eintragen.

P.R.C.K.	B.E.T.R.A.G.,B.E.T.R.A.G. ,
P.R.C.K.	K.O.S.T.E.N.,K.O.S.T.E.N. ,

Sind alle Anweisungen auf dem Programmschema abgelocht, so kann man die nachträglich hinzugefügten Instruktionen an ihrem eigentlichen Platz einordnen.

Für das Beispiel in Abb. 11 ergäbe sich in richtiger Reihenfolge die Kartennumerierung 10, __(Z)__, __(Z)__, 20, 30, usw.

11

12

Wir fassen abschließend zum Thema Programmschema noch einmal zusammen:

22 Das symbolische Programm muß in einer Form geschrieben sein, die dem Umwandlungsprogramm eine eindeutige Übersetzung der Assemblersprache in die Maschinensprache ermöglicht.
Deshalb müssen beim Schreiben eines Programms in der Assemblersprache einige Regeln beachtet werden.

Als Hilfsmittel hierzu dient uns eine Ablochvorlage, nämlich das _____.

Programmschema

IBM **System / 360 Assembler**
Programmschema

Programm		Lochhinweise		Problem (73 75)
		Zeichen		
Programmierer	Datum	Lochung		Seite (76 77) von ____

Anweisung				Karte Nr.
Name (1 8)	Operation (10 14)	Operand und Bemerkungen (20 25 30 35 40 45 50 55 60 65 70)		(78 80)
				010
				020
				030
				040
				050
				060
				070
				080
				090
				100
				110
				120
				130
				140
				150
				160
				170
				180
				190
				200

Abb. 12

23 Wesentlich für die Umwandlung des symbolischen Programms ist das, was in die Anweisungsfelder _____ (1-8), _____ (10-14) und _____ (16-71) eingetragen wird.

Durch 3 besonders kenntlich gemachte Spalten sind diese Felder abgegrenzt. Es sind dies die Spalten 9, 15 und 72. Diese Spalten dürfen nur in Ausnahmefällen benutzt werden, so z.B. für Bemerkungszeilen die Spalten __(Z)__ und __(Z)__.

Name

Operation

Operand und Bemerkungen

9 (und) 15

24 Was geschieht nun, wenn die formalen Regeln beim Schreiben eines Programms nicht eingehalten werden, wenn z.B. ein Symbol nicht korrekt gewählt ist, der Stern bei Bemerkungszeilen vergessen wurde, usw. ?

Wenn gegen diese formalen Regeln verstoßen wurde, dann werden bei der Umwandlung des symbolischen Programms auf der Programmliste Fehlerhinweise ausgedruckt. Das Programm muß entsprechend korrigiert und danach erneut umgewandelt werden.

Verstöße gegen die in diesem Kapitel besprochenen formalen Regeln werden also bei der Umwandlung festgestellt und auf der _____ vermerkt.

Umwandlungsliste/Programmliste

IBM System / 360 Assembler
Programmschema

Programm Nr. 12		Lochhinweise	Problem 012 (73–75)
Monatliche Dispositionsliste		Zeichen) &	
Programmierer Johann Alois Kohr	Datum 11.11.68	Lochung 5-8 12	Seite 02 (76–77) von 4

Anweisung			Karte Nr.
Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	(78–80)
* VERARBE	ITUNG	DER DAUERAUFTRAEGE	010
			020
B4	MVC	ANR,ENR	030
			040
A5	CLC	BETRAG,KONST1000 EINE KONSTANTE 1000 WIRD MIT DEM BE-	050
TRAG VERG	LICHEN		060
			070
BETA	PACK	PFELD,ALPHA&	080
			090
1AB	AP	SUMME, BETRAG	100
			110
GAMMA	AP	GESKOST,EIKOST	120
			130
	PACK	PERS NR,PERSNR	140
			150
			160
			170
			180
			190
			200

Abb. 13

FRAGEN ZUM KAPITEL II

Überprüfen Sie nun, ob Sie den Lernstoff dieses Kapitels beherrschen. Beantworten Sie erst alle Fragen auf einem Extrablatt, bevor Sie Ihre Antworten auf der folgenden Seite überprüfen.

In Abb. 13 sehen Sie ein Programmschema, in dem einige Eintragungen vorgenommen sind. Beantworten Sie bitte die folgenden Fragen zu Abb. 13. Wenn Sie Fehler in der Abbildung feststellen, sollen Sie auch jeweils sagen, was falsch gemacht wurde.

1. Entsprechen die Eintragungen in den Zeilen 10 und 30 den Regeln?

2. In der Zeile 50 sind dem Programmierer Fehler unterlaufen. Was ist falsch?

3. Ist die Eintragung in Zeile 60 ordnungsgemäß erfolgt?

4. Sind die Eintragungen in Zeile 80 korrekt?

5. Was ist in der Anweisung der Zeile 100 falsch?

6. Verstößt die Anweisung in Zeile 120 gegen die formalen Regeln, die Sie in diesem Kapitel kennengelernt haben?

7. Was ist in der Zeile 140 falsch?

ANTWORTEN ZUM KAPITEL II

Ihre Antworten sollen sinngemäß den hier vorgegebenen Antworten entsprechen.

Sind Ihnen bei der Beantwortung Fehler unterlaufen, dann sollten Sie die entsprechenden Lernelemente oder Lesetexte (Angabe in Klammern) noch einmal wiederholen.

1. Ja, die Eintragungen sind korrekt.

2. Das Symbol KONST1ØØØ ist zu groß. - Spalte 72 darf nicht benutzt werden.

   (LE 12 - 17, 23)

3. Nein. - In der Zeile 60 fehlt der Stern (*) in Stelle 1 zur Kennzeichnung einer Bemerkungszeile.

   (LE 14 - 17)

4. Nein. - Zeile 80 enthält 2 Fehler:

   a) Der Name BETA beginnt nicht in Stelle 1.
   (LE 9 - 11)

   b) Der Operand ALPHA& enthält ein Sonderzeichen.
   (LE 12 - 13)

5. Zeile 100 enthält 2 Fehler:

   a) Der Name (Symbol) 1AB beginnt mit einer Ziffer.
   (LE 12 - 13)

   b) Zwischen den Operanden befindet sich eine Leerstelle.
   (LE 6 - 8)

6. Nein, die Eintragungen sind korrekt.

7. Das Symbol PERSNR enthält eine Leerstelle.

(LE 12 - 13)

Schauen Sie sich nun Merkblatt 303 an. -



Band 1, Seite 11

KAPITEL III

DIE ASSEMBLERANWEISUNGEN START, END, ORG, DS

Lernstoff:

Die Assembleranweisung START

Die Assembleranweisung END

Die Assembleranweisung ORG

Die Assembleranweisung DS

(Der Zuordnungszähler - Die symbolische Adressierung - Die Definition der Ein-/Ausgabebereiche - Adreßrechnung und explizite Längenangabe für Felder im Hauptspeicher)

Es gibt zwei Arten von Anweisungen im symbolischen Programm:

a) Anweisungen für Maschineninstruktionen

b) Anweisungen für Assemblerinstruktionen

Die Maschineninstruktions-Anweisungen veranlassen die Maschine (das DVS), eine Reihe von Instruktionen durchzuführen. Durch sie wird z. B. das Addieren, Subtrahieren, Vergleichen und Übertragen der Daten veranlaßt. Die Maschineninstruktions-Anweisungen, die vom Programmierer symbolisch angegeben werden, werden durch die Umwandlung in echte Instruktionen (Instruktionen in Maschinensprache) umgewandelt.

Die Assemblerinstruktions - Anweisungen hingegen geben dem Assembler-Umwandlungsprogramm Anweisungen für die Umwandlung des symbolischen Programms. Durch sie werden z. B. die Umwandlung gesteuert und Felder, Bereiche und Konstanten festgelegt. Entgegen den Maschineninstruktions-Anweisungen sind die Assemblerinstruktions-Anweisungen im Maschinenprogramm (echten Programm) nicht mehr enthalten.

Wir befassen uns in dem vorliegenden Band nun hauptsächlich mit den Assemblerinstruktions-Anweisungen und ihrer Verwendung im symbolischen Programm.

START

PROGRAMMANFANG

(Start program)

Beispiel einer Instruktion:

Name	Operation	Oper
1 8	10 14	20
	START	800

1. Der Operationsschlüssel START zeigt den Anfang des symbolischen Programms an.
2. Die START-Anweisung ist die erste Anweisung des Programms.
3. Der Operand der START-Anweisung ist ein Direktwert (eine Dezimalzahl). Dieser Direktwert (z.B. 800) gibt die Startadresse für das in den Hauptspeicher zu ladende Maschinenprogramm an.
4. Durch die START-Anweisung wird der Zuordnungszähler auf einen Anfangswert (Anfangsadresse) eingestellt.- Der Zuordnungszähler weist für Maschineninstruktionen, Felder und Bereiche fortlaufende Speicheradressen zu. An anderer Stelle wird die Funktion dieses Zählers noch eingehend beschrieben werden.

Abb. 1

Schauen Sie sich bitte Abb. 1 an und lesen Sie den Text in dieser Abbildung aufmerksam durch. -

Benutzen Sie Abb. 1 als Hilfe zur Beantwortung der in den folgenden Lernelementen vorkommenden Fragen und Auslassungen.

1 Der Operationsschlüssel START gibt für die Umwandlung den Anfang des symbolischen Programms an. Der Operationsschlüssel ist festgelegt.

Der Programmanfang wird also immer durch den Operationsschlüssel _____ gekennzeichnet.

START

2 Für die START-Anweisung ist nur e i n Operand zulässig. Dieser Operand ist ein _____.

Direktwert

3 Wie Sie sehen, werden nicht nur Symbole als Operanden benutzt. Für bestimmte Instruktionen werden im Operandenfeld Direktwerte verwendet. Direktwerte sind feste Werte, die in Zahlen ausgedrückt werden.

In der Codierung der Abb. 1 gibt der Direktwert __(Z)___ an, daß das umgewandelte Programm im Hauptspeicher ab der Stelle __(Z)___ geladen werden soll.

800

800

In den niedrigsten Hauptspeicherstellen befinden sich von der IBM bereitgestellte Programmroutinen, die der Steuerung und Überwachung des Programmablaufs dienen. Je nach Größe und Ausstattung eines DVS ist somit eine gewisse Anzahl von Bytes schon fest vergeben. Es ist daher nicht möglich, das Programm schon ab der Adresse 0 im Hauptspeicher zu speichern.

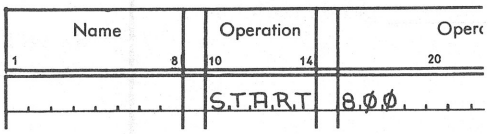
Wir benutzen in diesem Lehrgang im allgemeinen Startadressen, die zwischen 600 und 1000 liegen.

START

PROGRAMMANFANG

(Start program)

Beispiel einer Instruktion:



1. Der Operationsschlüssel START zeigt den Anfang des symbolischen Programms an.

2. Die START-Anweisung ist die erste Anweisung des Programms.

3. Der Operand der START-Anweisung ist ein Direktwert (eine Dezimalzahl). Dieser Direktwert (z.B. 800) gibt die Startadresse für das in den Hauptspeicher zu ladende Maschinenprogramm an.

4. Durch die START-Anweisung wird der Zuordnungszähler auf einen Anfangswert (Anfangsadresse) eingestellt.- Der Zuordnungszähler weist für Maschineninstruktionen, Felder und Bereiche fortlaufende Speicheradressen zu. An anderer Stelle wird die Funktion dieses Zählers noch eingehend beschrieben werden.

Abb. 1

4

Name	Operation	Oper
1 8	10 14	20
	START	600

Diese Anweisung teilt dem Assembler-Umwand - lungsprogramm mit, daß das Programm nach der Umwandlung ab der Adresse __(Z)__ in den Hauptspeicher geladen werden soll.

600

Im Namenfeld einer START-Anweisung kann ein symbolischer Name vergeben werden, der dann dem Programm als Name zugeordnet wird. - Näheres hierzu werden Sie in den Folgelehrgängen erfahren.

Übungen

Wie müssen die START-Anweisungen lauten, wenn

a) ein Programm ab der Adresse 960 in den Hauptspeicher geladen werden soll,

b) ein Programm die Startadresse 720 erhalten soll,

c) ein Programm die Startadresse 1040 erhalten soll?

	Name (1–8)	Operation (10–14)	Operand (20)
a)		START	960
b)		START	720
c)		START	1040



Seite 103 (DS)

END

PROGRAMMENDE

(End program)

Beispiel einer Instruktion:

Name	Operation	Operand u
1 8	10 14	20
	E,N,D,	A,N,F,A,N,G,

1. Der Operationsschlüssel END zeigt dem Umwandlungsprogramm das Ende des symbolischen Programms an.
2. Das Namenfeld wird nicht benutzt.
3. Als Operand wird eine symbolische Bezeichnung eingetragen. Der Operand bezeichnet die Instruktion im Programm, die nach dem Laden des Maschinenprogramms als erste ausgeführt werden soll.
4. Die END-Anweisung muß stets die letzte Anweisung im symbolischen Programm sein.

Abb. 1

1 Lesen Sie bitte zunächst den Text in Abb. 1 aufmerksam durch. -

Die END-Anweisung bezeichnet für die Umwandlung das _____ des symbolischen Programmes.

Die END-Anweisung muß stets die ____(WW)____ des symbolischen Programms sein.

Ende

letzte Anweisung

2 Der Operand der END-Anweisung gibt den Namen der Instruktion an, mit der nach dem Laden des Programms der Programmablauf, also die Verarbeitung der Daten, beginnen soll.

Name	Operation	Operand u
1 8	10 14	20
	START	8,ØØ
ANFANG	BALR	9,Ø
	USING	*,9
	:	
	:	
	END	ANFANG

In diesem Ausschnitt aus einem Programm verweist die END-Anweisung auf die Instruktion _____.

Mit dieser Instruktion soll nach dem Laden des Programms der Programmablauf beginnen.

ANFANG	BALR	9,Ø

Übungen

Geben Sie bitte folgende END-Anweisungen an:

a) Die END-Anweisung soll auf eine erste auszuführende Instruktion mit dem Namen BEGINN verweisen.

b) Die als erste auszuführende Instruktion hat den Namen Á3.

	Name (1 … 8)	Operation (10 … 14)	Operand und Beme (20 … 25)
a)		E,N,D,	B,E,G,I,N,N,
b)		E,N,D,	A,3,

ORG

SETZEN ZUORDNUNGSZÄHLER

(Set location counter)

Beispiele für Instruktionen:

Name	Operation	Operand und Be
1 8	10 14	20 25
	ORG	EBER
	ORG	EBER+300
	ORG	*-500

1. Die Instruktion "Setzen Zuordnungszähler" wird dazu benutzt, den Inhalt des Zuordnungszählers im laufenden Programm zu ändern. Der symbolische Operationsschlüssel heißt ORG.

2. Der Operand kann sein:

   a) ein Symbol, das schon vorher definiert ist (z.B. EBER),

   b) ein zusammengesetzter Ausdruck, d.h. ein Symbol in Verbindung mit Adressrechnung (z.B. EBER+300, *-500).

   Ein Direktwert allein ist nicht erlaubt.

3. Der Zuordnungszähler wird auf den Wert gesetzt, den der Ausdruck im Operanden darstellt.

4. Die ORG-Anweisung kann überall und beliebig oft im Programm verwendet werden.

Abb. 1

1 Lesen Sie bitte zunächst den Text in Abb. 1 aufmerksam durch. -

In der Codierung der Abb. 1 sind Beispiele für die möglichen Formen des Operanden der ORG-Anweisung angeführt.

Der Operand kann sein:

a) ein Symbol, z. B. _____,

b) ein zusammengesetzter Ausdruck, bestehend aus einem Symbol und einem Direktwert, z. B. _____ oder _____.

a) EBER

b) EBER+300 (oder)✱-500

2 Der Zuordnungszähler wird auf den Wert gesetzt, den der Ausdruck im Operanden der ORG-Anweisung darstellt.

Wenn EBER der Adresse 1000 entspricht, dann bedeutet das, daß der Zuordnungszähler durch die Anweisung

```
ORG   EBER
```

auf (Z) gesetzt wird.

1000

3

```
ORG   EBER+300
```

Unter der Voraussetzung, daß EBER die Adresse 1000 hat, wird der Zuordnungszähler durch die obige Anweisung auf den Wert (Z) eingestellt.

1300

ORG

SETZEN ZUORDNUNGSZÄHLER

(Set location counter)

Beispiele für Instruktionen:

Name	Operation	Operand und Be
1 8	10 14	20 25
	ORG	EBER
	ORG	EBER+300
	ORG	*-500

1. Die Instruktion "Setzen Zuordnungszähler" wird dazu benutzt, den Inhalt des Zuordnungszählers im laufenden Programm zu ändern. Der symbolische Operationsschlüssel heißt ORG.

2. Der Operand kann sein:

   a) ein Symbol, das schon vorher definiert ist (z.B. EBER),

   b) ein zusammengesetzter Ausdruck, d.h. ein Symbol in Verbindung mit Adressrechnung (z.B. EBER+300, *-500).

   Ein Direktwert allein ist nicht erlaubt.

3. Der Zuordnungszähler wird auf den Wert gesetzt, den der Ausdruck im Operanden darstellt.

4. Die ORG-Anweisung kann überall und beliebig oft im Programm verwendet werden.

Abb. 1

4 Sie sehen, in den ORG-Anweisungen wird der Zuordnungszähler immer relativ zu einem definierten Symbol verändert.

Dies geschieht, indem entweder das Symbol selbst (z. B. EBER) oder aber das Symbol in Verbindung mit einem Direktwert (z. B. EBER+300) als Operand der ORG-Anweisung benutzt wird.

Wie muß dementsprechend eine ORG-Anweisung lauten, die den Zuordnungszähler auf den Wert eines Symbols TABA stellen soll?

Name	Operation	Oper
1 8	10 14	20
	ORG	TABA

5 Bezogen auf ein definiertes Symbol, kann der Zuordnungszähler auch zurückgestellt werden. Das geschieht durch die Benutzung des Minus-Zeichens (-) in einem zusammengesetzten Ausdruck.

Name	Operation	Operand und Bemerkung
1 8	10 14	20 25
	ORG	ABER-90

Durch diese Anweisung wird der Zuordnungszähler um einen Wert zurückgestellt, der um 90 unter dem Wert (der Adresse) des Symbols ABER liegt.

Der Zuordnungszähler soll um 50 unter den Wert eines definierten Symbols BEREICHB zurückgestellt werden. Wie muß die ORG-Anweisung lauten?

Name	Operation	Operand und Ber
1 8	10 14	20 25
	ORG	BEREICHB-50

6 Man kann den Zuordnungszähler auch verändern, indem man von seinem augenblicklichen Wert ausgeht.

ORG	*	-500

Diese Anweisung bewirkt, daß der Wert des Zuordnungszählers um (ZW) wird.

500 vermindert

ORG

SETZEN ZUORDNUNGSZÄHLER

(Set location counter)

Beispiele für Instruktionen:

Name	Operation	Operand und Be
1 8	10 14	20 25
	ORG	EBER
	ORG	EBER+300
	ORG	*-500

1. Die Instruktion "Setzen Zuordnungszähler" wird dazu benutzt, den Inhalt des Zuordnungszählers im laufenden Programm zu ändern. Der symbolische Operationsschlüssel heißt ORG.

2. Der Operand kann sein:

   a) ein Symbol, das schon vorher definiert ist (z.B. EBER),

   b) ein zusammengesetzter Ausdruck, d.h. ein Symbol in Verbindung mit Adressrechnung (z.B. EBER+300, *-500).

   Ein Direktwert allein ist nicht erlaubt.

3. Der Zuordnungszähler wird auf den Wert gesetzt, den der Ausdruck im Operanden darstellt.

4. Die ORG-Anweisung kann überall und beliebig oft im Programm verwendet werden.

Abb. 1

7 Der Stern (*) ist das Symbol für den augenblicklichen Wert des Zuordnungszählers.

Der augenblickliche Wert des Zuordnungszählers soll

a) um 100 erhöht,

b) um 50 vermindert werden.

Wie müssen die ORG-Anweisungen lauten?

	Name (1–8)	Operation (10–14)	Operand (20–)
a)		ORG	*+100
b)		ORG	*-50

Übungen

Entnehmen Sie bitte Ihren Arbeitsmitteln ein Programmschema und tragen Sie die folgenden ORG-Anweisungen ein:

a) Der Zuordnungszähler soll auf den Wert des Symbols AREA1 gesetzt werden.

b) Der Zuordnungszähler soll, bezogen auf ein Symbol FELDA, um 80 vermindert werden.

c) Der Zuordnungszähler soll, bezogen auf ein Symbol EBER, um 45 erhöht werden.

d) Der momentane Wert des Zuordnungszählers soll um 180 erhöht werden.

e) Der momentane Wert des Zuordnungszählers soll um 185 vermindert werden.

	Name (1 – 8)	Operation (10 – 14)	Operand und Beme (20 – 25)
a)		ORG	AREA1
b)		ORG	FELDA-80
c)		ORG	EBER+45
d)		ORG	*+180
e)		ORG	*-185

Band 1, Seite 419

DS

DEFINIEREN SPEICHER

(Define Storage)

Beispiel einer Instruktion:

Name	Operation	Operand u
1 8	10 14	20
P,E,R,S,N,R, .	D,S, . .	C,L,5, ...

1. Durch die DS-Anweisung wird ein Bereich oder ein Feld im Hauptspeicher reserviert und erhält einen symbolischen Namen zugeordnet.

2. Der Operand (z. B. CL5) gibt die Länge des Feldes oder Bereichs in Bytes an.

3. Die symbolische Bezeichnung im Namenfeld (z. B. PERSNR) wird dem Bereich oder dem Feld im Hauptspeicher als Name zugeordnet. Durch diesen Namen ist jeder definierte Speicherbereich im Laufe des Programms ansprechbar.

Abb. 1

1 Lesen Sie bitte zunächst den Text in Abb. 1 aufmerksam durch und fahren Sie dann mit diesem Lernelement fort. -

Wie Sie wissen, sind die Ein-/Ausgabebereiche im Hauptspeicher, in die die Daten eingelesen und aus denen die Daten ausgegeben werden, frei wählbar. Die Bereiche und Felder für die Daten im Hauptspeicher müssen vom Programmierer reserviert werden. Das geschieht mit Hilfe der Assemblerinstruktion "Definieren Speicher". Der symbolische Operationsschlüssel dieser Assemblerinstruktion lautet _____.

DS

2 Die DS-Anweisung dient der Reservierung von Feldern und Bereichen im _____.

Der Operand gibt die Länge des zu reservierenden Bereiches oder Feldes an. In der Codierung der Abb. 1 lautet der Operand der DS Anweisung _____.

Hauptspeicher

CL5

3 Das "C" in diesem Operanden bedeutet soviel wie Zeichen (engl. Character). Es zeigt an, daß einzelne Speicherstellen zu reservieren sind.

Wieviel Speicherstellen reserviert werden sollen, wird durch die dem Buchstaben L (=Länge) folgende Zahl festgelegt.

In dem Beispiel in Abb. 1 ist das definierte Feld im Hauptspeicher (Z) Bytes groß.

5

4 Wie lauten die Operanden für ein

a) 2-stelliges Feld,

b) 7-stelliges Feld,

c) 25-stelliges Feld?

a) CL2

b) CL7

c) CL25

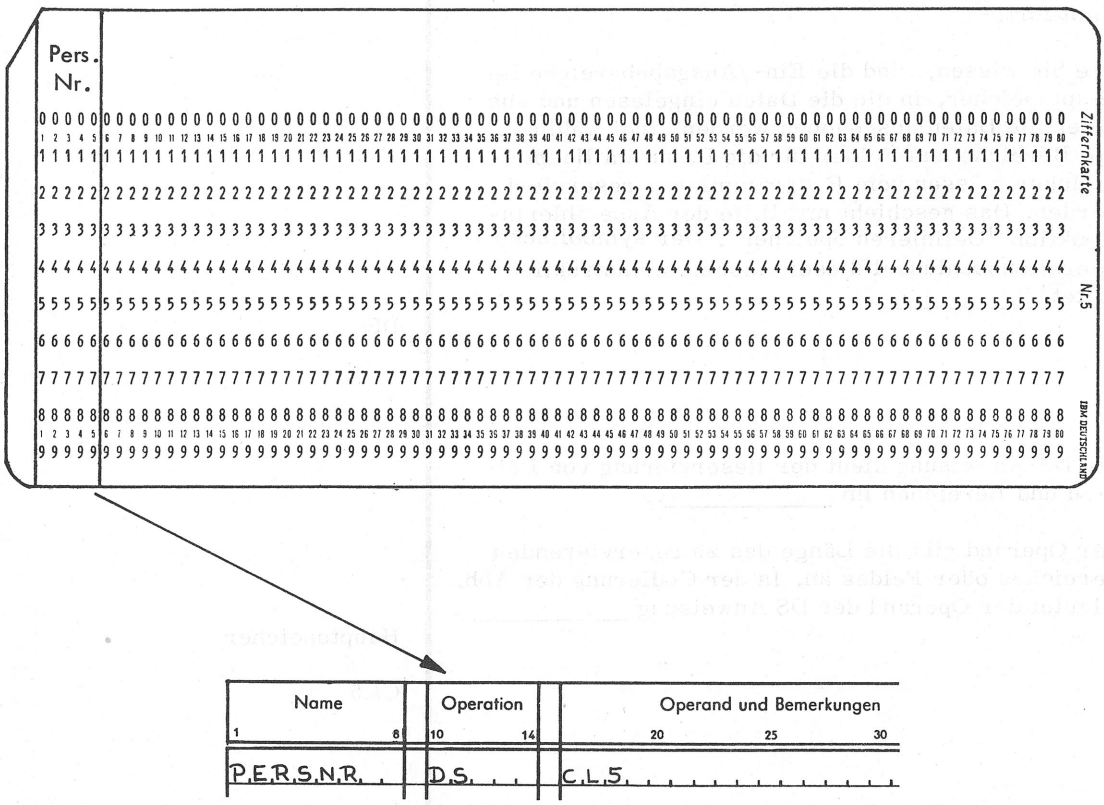

Pers.
Nr.
Ziffernkarte
Nr. 5
IBM DEUTSCHLAND
Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30
P,E,R,S,N,R,
D,S,
C,L,5,

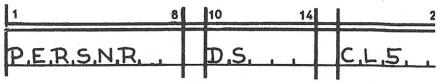
Abb. 2

5 Die symbolische Bezeichnung im Namenfeld wird dem Bereich oder dem Feld im Hauptspeicher als Name zugeordnet.

Betrachten Sie bitte Abb. 2. -

Die Daten aus der Lochkarte, die für die Verarbeitung im DVS benötigt werden, werden in den Hauptspeicher eingelesen. Für das Feld "Personal-Nr." in der Lochkarte (Abb. 2) muß also im Hauptspeicher ein fünfstelliges Feld reserviert werden. Das geschieht durch die Anweisung _____.

Dabei erhält das im Hauptspeicher reservierte Feld den symbolischen Namen _____ zugeordnet.



PERSNR

6

Name	Operation	Operand
1 8	10 14	20
P,E,R,S,N,R, ,	D,S, , ,	C,L,5, , , , ,

Durch diese Anweisung ist das Feld "Personal-Nr." der Lochkarte für das DVS definiert.

Alle Daten einer Lochkarte, die für die Verarbeitung benötigt werden, müssen mit Hilfe der DS-Anweisungen definiert werden. Man bezeichnet daher die DS-Anweisungen auch als Definitionsinstruktionen.

Das Feld "Personal-Nr." (Abb. 2) ist definiert durch

a) die Zuordnung des symbolischen Namens _____,

b) die Längenangaben des Operanden _____.

a) PERSNR

b) CL5

7 Ein Feld in der Lochkarte, das 15 Spalten umfaßt, soll den symbolischen Namen ARTBEZ (Artikel-Bezeichnung) erhalten.

Tragen Sie bitte die entsprechende DS-Anweisung in ein Programmschema ein.

Name	Operation	Operand un
1 8	10 14	20
A,R,T,B,E,Z, ,	D,S, , ,	C,L,1,5, , , , ,

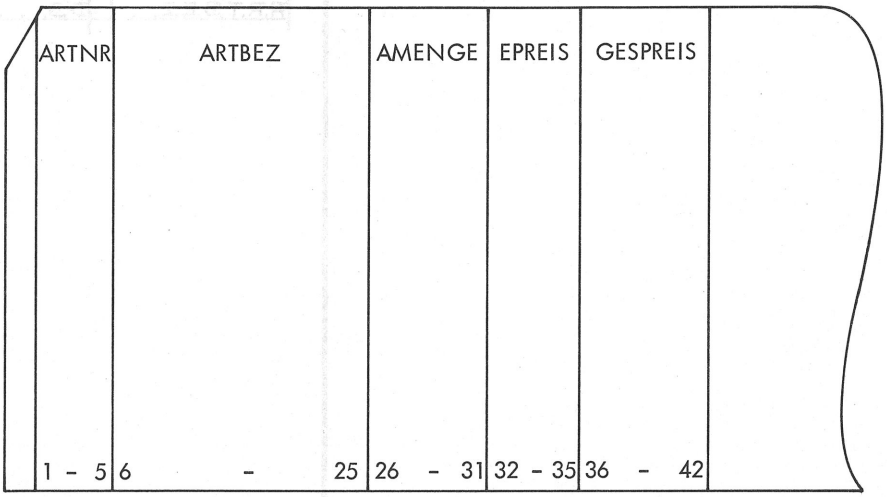


Abb. 3

Übungen

Abb. 3 zeigt Ihnen eine Lochkarte mit 5 Feldern. -

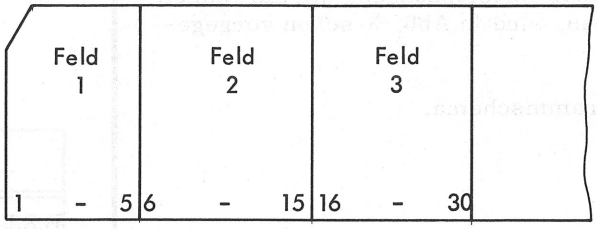
Definieren Sie bitte diese Daten mit Hilfe von DS-Anweisungen. Die symbolischen Bezeichnungen, die als Namen verwendet werden sollen, sind in Abb. 3 schon vorgegeben.

Benutzen Sie ein Programmschema.

Name 1 8	Operation 10 14	Op 20
A.R.T.N.R. . .	D.S. . .	C.L.5. . .
A.R.T.B.E.Z. .	D.S. . .	C.L.2.Ø. .
A.M.E.N.G.E. .	D.S. . .	C.L.6. . .
E.P.R.E.I.S. .	D.S. . .	C.L.4. . .
G.E.S.P.R.E.I.S	D.S. . .	C.L.7. . .

Sind Ihnen Fehler unterlaufen, dann stellen Sie bitte die Fehlerursache fest. Wiederholen Sie gegebenenfalls die Lernelemente 1-7.

a) Daten in der Lochkarte



b) Codierung

Name	Operation	Operand und
1 8	10 14	20 2
	START	1Ø,Ø,1
FELD1	DS	CL5
FELD2	DS	CL1Ø
FELD3	DS	CL15

c) Zuordnung im HSP

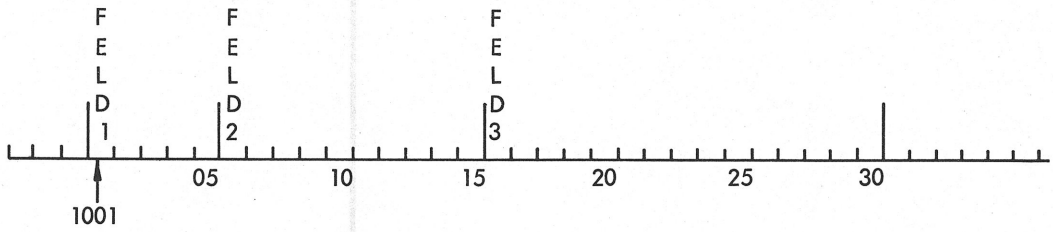


Abb. 4

Der Zuordnungszähler

Um Feldern und Bereichen Hauptspeicheradressen zuzuordnen, benutzt das Assembler-Umwandlungsprogramm einen Zuordnungszähler.

Betrachten Sie bitte Abb. 4. -

Während der Umwandlung des Programms werden den Maschineninstruktionen, Konstanten, Bereichen und Feldern mit Hilfe des Zuordnungszählers in der Reihenfolge der Codierung Hauptspeicherplätze zugewiesen.

Durch die Codierung in Abb. 4 wird festgelegt, daß das Programm ab der Stelle 1001 in den Hauptspeicher geladen wird. Anschließend sollen im Hauptspeicher 3 Felder freigehalten werden, nämlich 5 Bytes für FELD1, 10 Bytes für FELD2 und 15 Bytes für FELD3.

Wie dies im einzelnen geschieht, erfahren Sie in den folgenden Lernelementen.

8 Anhand der Codierung in Abb. 4 verfolgen wir nun, wie die Zuordnung der Bereiche im Hauptspeicher vonstatten geht.

Durch die START-Anweisung wird der Zuordnungszähler auf eine Anfangsadresse eingestellt. Ab dieser Adresse soll das Programm in den Hauptspeicher geladen werden.

In Abb. 4 wird der Zuordnungszähler durch die Instruktion START auf die Anfangsadresse __(Z)__ eingestellt.

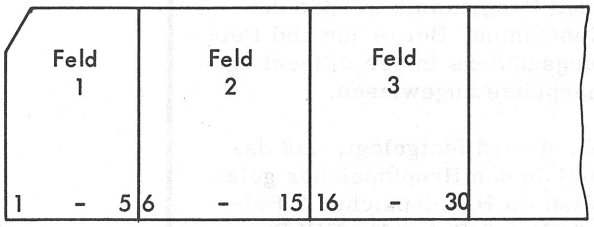
Eine Anweisung, die START 800 lautete, würde den _____ auf die __(WZ)__ einstellen.

1001

Zuordnungszähler

Anfangsadresse 800

a) Daten in der Lochkarte



b) Codierung

Name	Operation	Operand und
1	10 14	20
	START	1Ø01
FELD1	DS	CL5
FELD2	DS	CL10
FELD3	DS	CL15

c) Zuordnung im HSP

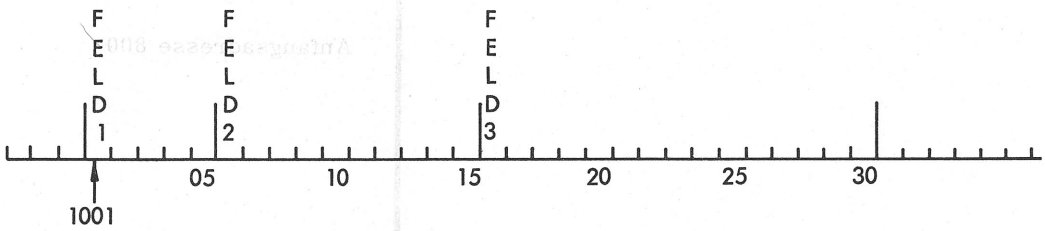


Abb. 4

9 Der Zuordnungszähler weist immer auf die nächste verfügbare Adresse hin. Durch die START-Anweisung wird die erste verfügbare Adresse im Hauptspeicher angegeben.

Die erste verfügbare Adresse lautet gemäß der Codierung in Abb. 4 1001.

```
FELD1...|DS...|CL5.....
```

Durch diese Anweisung aus Abb. 4 werden im Hauptspeicher ab der Adresse __(Z)__ für FELD1 __(Z)__ Bytes freigehalten.

1001

5

10 Der Zuordnungszähler wird nun automatisch um die für FELD1 angegebene Länge erhöht.
Da FELD1 __(Z)__ Stellen groß ist (CL5), wird der Zuordnungszähler entsprechend um __(Z)__ erhöht.

5

5

11 Wir fassen zusammen:

Durch die START-Anweisung in Abb. 4 wird der Zuordnungszähler auf die Adresse __(Z)__ eingestellt.

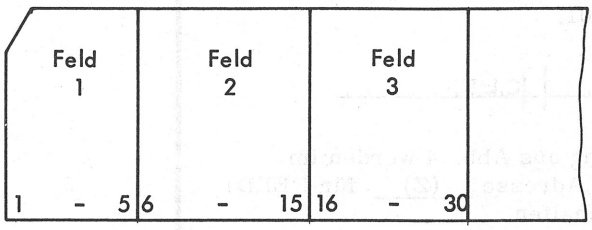
Durch die nachfolgende DS-Anweisung wird der Zuordnungszähler um 5 erhöht. Er verweist also jetzt auf die Adresse __(Z)__.

Diese Adresse ist als nächste verfügbar.

1001

1006

a) Daten in der Lochkarte



b) Codierung

Name	Operation	Operand und
	START	1001
FELD1	DS	CL5
FELD2	DS	CL10
FELD3	DS	CL15

c) Zuordnung im HSP

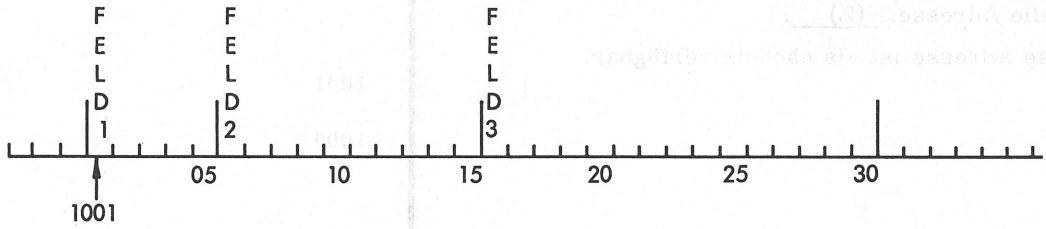


Abb. 4

12 F.E.L.D.2. . . | D.S. . . | C.L.1.Ø. . . . . .

Durch diese Anweisung in Abb. 4 werden für FELD2 die Speicherstellen __(Z)__ bis __(Z)__ freigehalten.

Der Zuordnungszähler wird um __(Z)__ erhöht, so daß die nächste verfügbare Adresse __(Z)__ lautet.

Das bedeutet, daß für FELD3 die Speicherstellen __(Z)__ bis __(Z)__ reserviert werden.

1006 (bis) 1015

10

1016

1016 (bis) 1030

13 Wir merken uns:
Während der Umwandlung des Programms werden den Bereichen und Feldern mit Hilfe des Zuordnungszählers automatisch nacheinanderfolgende Hauptspeicherstellen zugewiesen.

Durch die START-Anweisung wird der Zuordnungszähler auf eine _____ eingestellt.

Ist ein Bereich oder ein Feld zugeordnet, dann verweist der _____ stets auf die nächste verfügbare _____ im Hauptspeicher.

Anfangsadresse

Zuordnungszähler

Adresse

End-Nr.	Anfangs-Nr.	Einzelpreis	Gesamtpreis	
1 - 6	7 - 12	13 - 20	21 - 30	

Name 1 8	Operation 10 14	Operanc 20
	S,T,A,R,T	1,6,Ø,1,
E,N,U,M,M,E,R,	D,S,	C,L,6,
A,N,U,M,M,E,R,	D,S,	C,L,6,
E,P,R,E,I,S,	D,S,	C,L,8,
G,E,S,P,R,E,I,S	D,S,	C,L,1,Ø,

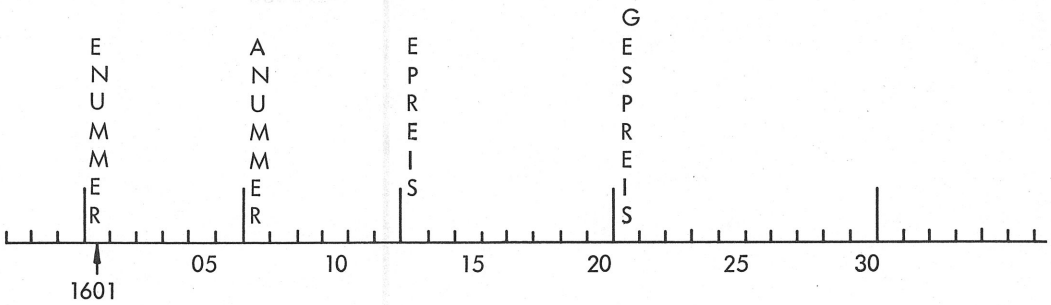


Abb. 5

Die symbolische Adressierung

Wie Sie bereits wissen, ist der Hauptspeicher von 0 beginnend aufwärts durchnumeriert. Die Adresse eines Feldes im Hauptspeicher ist die Nummer des linken Bytes eines Feldes (Linksadressierung!).

Betrachten Sie bitte Abb. 5. -

Die Adresse des mit ENUMMER bezeichneten Feldes im Hauptspeicher lautet 1601. Die Felder EPREIS und GESPREIS haben die Adressen 1613 und 1621.

Den symbolischen Bezeichnungen werden also mittels des Zuordnungszählers bei der Umwandlung Hauptspeicheradressen zugeordnet. Dadurch ist es möglich, daß man im Laufe des Programms z. B. den symbolischen Namen ENUMMER verwendet und daß durch diesen Namen dann im Hauptspeicher die Adresse 1601 angesprochen wird.

Die Adressierung der Felder und Bereiche im Hauptspeicher durch s y m b o l i s c h e Bezeichnungen nennt man s y m b o l i s c h e A d r e s s i e r u n g.

ANUMMER ist z. B. die s y m b o l i s c h e A d r e s s e (symbolische Bezeichnung) für die Adresse 1607 im Hauptspeicher (vgl. Abb. 5).

14 Durch die DS-Anweisung werden einem Feld im Hauptspeicher zwei Attribute zugeordnet, nämlich ein A d r e s s - und ein L ä n g e n a t t r i b u t.

G.E.S.P.R.E.I.S		D.S. . .		C.L.1.Ø. . . . .

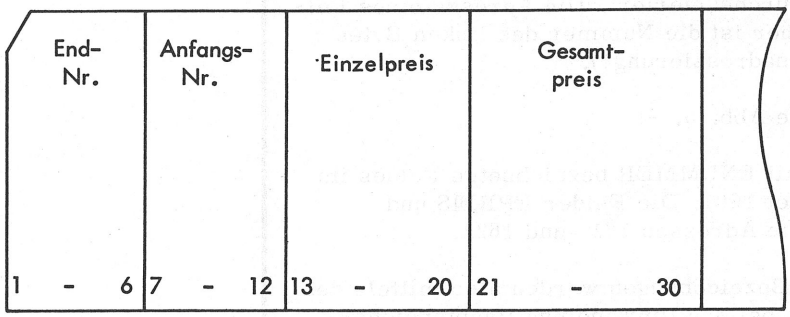
Durch diese Anweisung wird dem zu reservierenden Feld im Hauptspeicher die symbolische Adresse _____ bzw. die Hauptspeicheradresse ___(Z)___ (vgl. Abb. 5) zugeordnet.

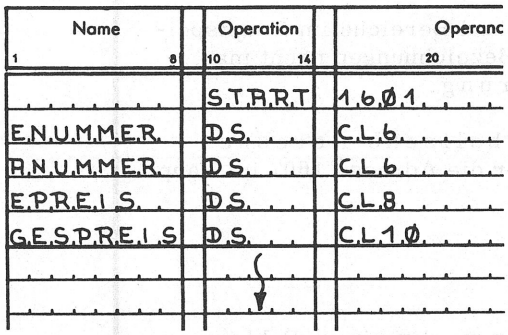
Das Längenattribut dieses Feldes ist die Länge ___(Z)___.

GESPREIS

1621

10


End-
Nr.
Anfangs-
Nr.
Einzelpreis
Gesamt-
preis
1 - 6
7 - 12
13 - 20
21 - 30


Name
Operation
Operanc
1
8
10
14
20
S,T,A,R,T
1,6,Ø,1,
E,N,U,M,M,E,R,
D,S,
C,L,6,
A,N,U,M,M,E,R,
D,S,
C,L,6,
E,P,R,E,I,S,
D,S,
C,L,8,
G,E,S,P,R,E,I,S
D,S,
C,L,1,Ø,

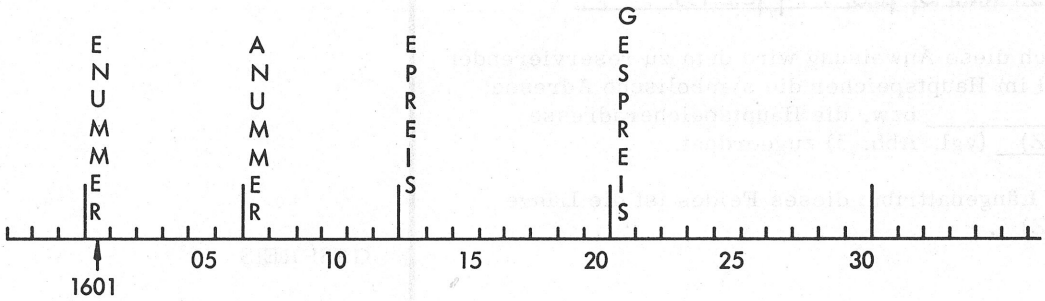

ENUMMER
ANUMMER
EPREIS
GESPREIS
1601
05
10
15
20
25
30

Abb. 5

15 Geben Sie bitte Adress- und Längenattribute für die Felder mit den folgenden symbolischen Adressen an (vgl. Abb. 5):

	Adress-attribut	Längen-attribut
ENUMMER	(Z)	(Z)
ANUMMER	(Z)	(Z)
EPREIS	(Z)	(Z)

Adress-attribut	Längen-attribut
1601	6
1607	6
1613	8

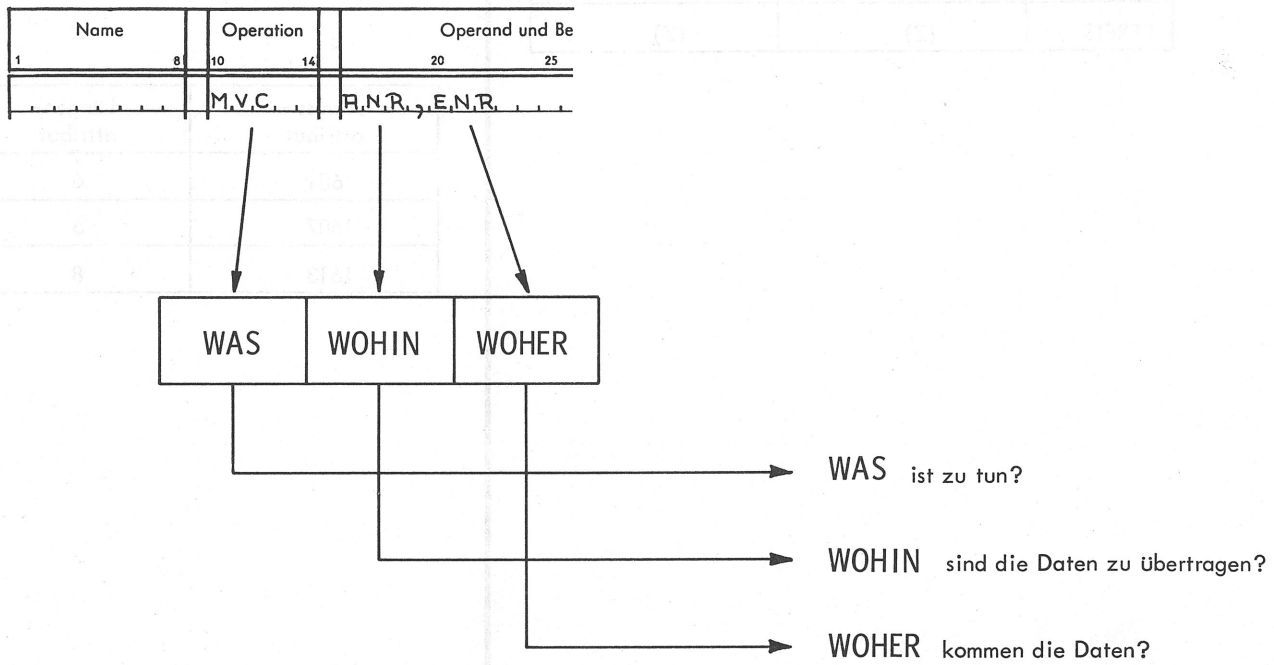

Name
Operation
Operand und Be
1
8
10
14
20
25
M,V,C,
R,N,R,,E,N,R,
WAS
WOHIN
WOHER
WAS ist zu tun?
WOHIN sind die Daten zu übertragen?
WOHER kommen die Daten?

Abb. 6

Betrachten Sie bitte Abb. 6. -

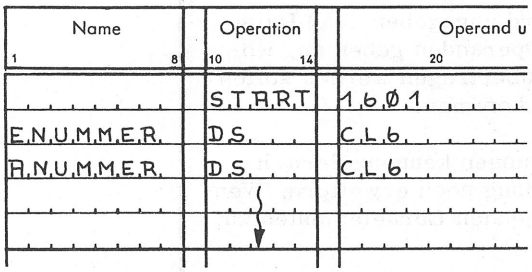
Wir hatten festgestellt:
Durch die Maschineninstruktion

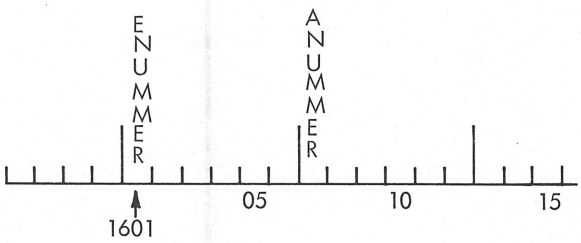
M.V.C. .	A.N.R., E.N.R. . .

bringt der Programmierer zum Ausdruck, was und mit welchen Daten etwas geschehen soll.

Im Operationsschlüssel MVC wird angegeben, daß Daten übertragen werden sollen. Die Operanden geben an, wohin die Daten im Hauptspeicher übertragen werden sollen und woher sie im Hauptspeicher kommen.

Nachdem wir nun die DS-Anweisungen kennengelernt haben, können wir die obige Abbildung noch erweitern. Wenden Sie sich hierzu bitte den folgenden Lernelementen zu.


Name
Operation
Operand u
1
8
10
14
20
S,T,A,R,T
1,6,Ø,1,
E,N,U,M,M,E,R,
D,S,
C,L,6,
A,N,U,M,M,E,R,
D,S,
C,L,6,


ENUMMER
ANUMMER
1601
05
10
15

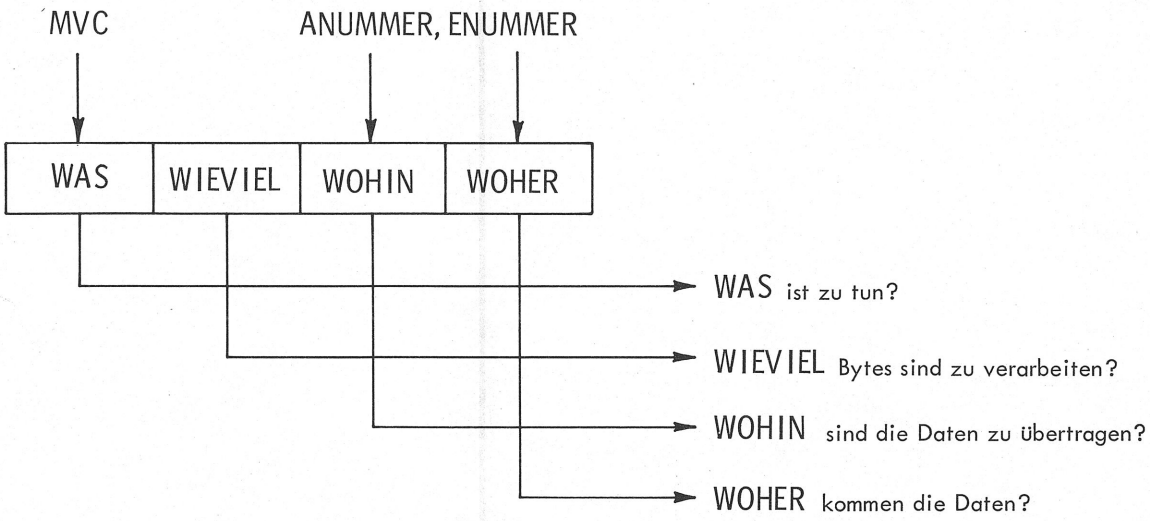

MVC
ANUMMER, ENUMMER
WAS
WIEVIEL
WOHIN
WOHER
WAS ist zu tun?
WIEVIEL Bytes sind zu verarbeiten?
WOHIN sind die Daten zu übertragen?
WOHER kommen die Daten?

Abb. 7

16 Betrachten Sie bitte Abb. 7. -

"WAS ist zu tun?" So lautet die erste Frage, wenn Daten verarbeitet werden sollen.

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

In dieser Instruktion wird die Frage nach dem WAS durch den symbolischen Operationsschlüssel _____ beantwortet.

WIEVIEL Speicherstellen zu verarbeiten sind, erfährt das DVS durch die Operanden der entsprechenden DS-Anweisungen. Bei der Operation MVC werden so viele Bytes verarbeitet, wie der 1. Operand der Instruktion angibt. ANUMMER hat eine Länge von __(Z)__ Bytes, denn der Operand der DS-Anweisung lautet _____.

Bei der Instruktion in Abb. 7 werden also __(ZW)__ verarbeitet.

MVC

6

CL6

6 Bytes

17 WOHIN sind die Daten zu übertragen?
Diese Frage wird durch den 1. Operanden der Maschineninstruktion mit der symbolischen Adresse _____ beantwortet.

Aufgrund der entsprechenden DS-Anweisung bezeichnet ANUMMER die Adresse __(Z)__ im Hauptspeicher.

ANUMMER

1607

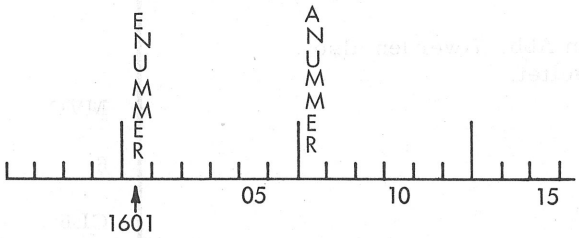
18 WOHER die Daten im Hauptspeicher kommen, wird ebenfalls durch eine symbolische Adresse ausgedrückt, nämlich durch _____.

ENUMMER bezeichnet die Adresse __(Z)__ im Hauptspeicher.

ENUMMER

1601

Name	Operation	Operand u
1 8	10 14	20
	S,T,A,R,T	1,6,Ø,1,
E,N,U,M,M,E,R,	D,S,	C,L,6,
A,N,U,M,M,E,R,	D,S,	C,L,6,



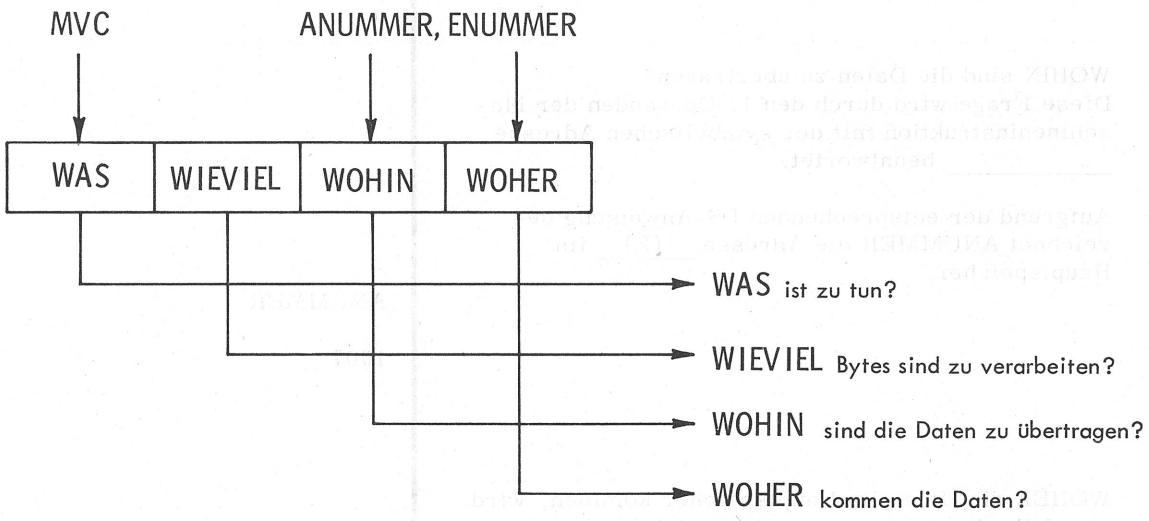


Abb. 7

19

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

Die DS-Anweisungen für ANUMMER und ENUMMER in Abb. 7 sind also unbedingt erforderlich, um dem DVS durch die obige Anweisung mitteilen zu können:

a) _____ Bytes zu verarbeiten sind (Längenangabe),

b) _____ die Daten zu übertragen sind (Adresse des 1. Operanden),

c) _____ die Daten kommen (Adresse des 2. Operanden).

Wieviel

Wohin

Woher

Wir fassen noch einmal in einigen Lernelementen zusammen, was wir bisher über die DS-Anweisung erarbeitet haben.

20 Durch die DS-Anweisung wird ein Bereich oder ein Feld im Hauptspeicher reserviert.

Die symbolische Bezeichnung im Namenfeld wird dem Bereich oder dem Feld im Hauptspeicher als Name zugeordnet. Der Operand gibt die _____ des Feldes oder Bereichs in Bytes an.

Länge

21 Ein Feld, das aus einer Lochkarte eingelesen wird, muß für das DVS definiert werden durch

a) die Zuordnung eines symbolischen _____,

b) die Angabe der _____ im Operanden.

Namens

Länge

22 Um den Feldern und Bereichen Hauptspeicheradressen zuordnen zu können, benutzt das Assembler-Umwandlungsprogramm einen _____.

Zuordnungszähler

23 Durch die START-Anweisung wird der Zuordnungszähler auf eine _____ eingestellt.

Aufgrund der Länge der definierten Felder und Bereiche weist dann der Zuordnungszähler entsprechend nacheinanderfolgende Hauptspeicherstellen zu.

Mit Hilfe des Zuordnungszählers werden einem durch DS reservierten Feld zwei Attribute zugeordnet, nämlich ein _____- und ein _____.

Anfangsadresse

Adress

Längenattribut

24

Name	Operation	Op
1 8	10 14	20
K,O,N,T,O,N,R,	D,S, , ,	C,L,5, , ,

Durch diese Anweisung wird dem zu reservierenden Feld im Hauptspeicher

a) die symbolische Adresse _____ und

b) die Länge __(Z)__ durch den Operanden _____ zugeordnet.

KONTONR

5

CL5

25 Die DS-Anweisungen gehören zu den Assemblerinstruktionen, durch die die Umwandlung gesteuert und die Felder und Bereiche definiert werden.

Erst mit Hilfe der DS-Anweisungen wird es ermöglicht, daß vom Datenverarbeitungssystem z. B. die folgende Maschineninstruktion durchgeführt werden kann:

```
MVC   ANUMMER,ENUMMER
```

Denn durch die DS-Anweisungen ist die _____ der Felder (das WIEVIEL), die _____ des 1. Operanden (das WOHIN) und die _____ des 2. Operanden (das WOHER) festgelegt.

Länge

Adresse

Adresse

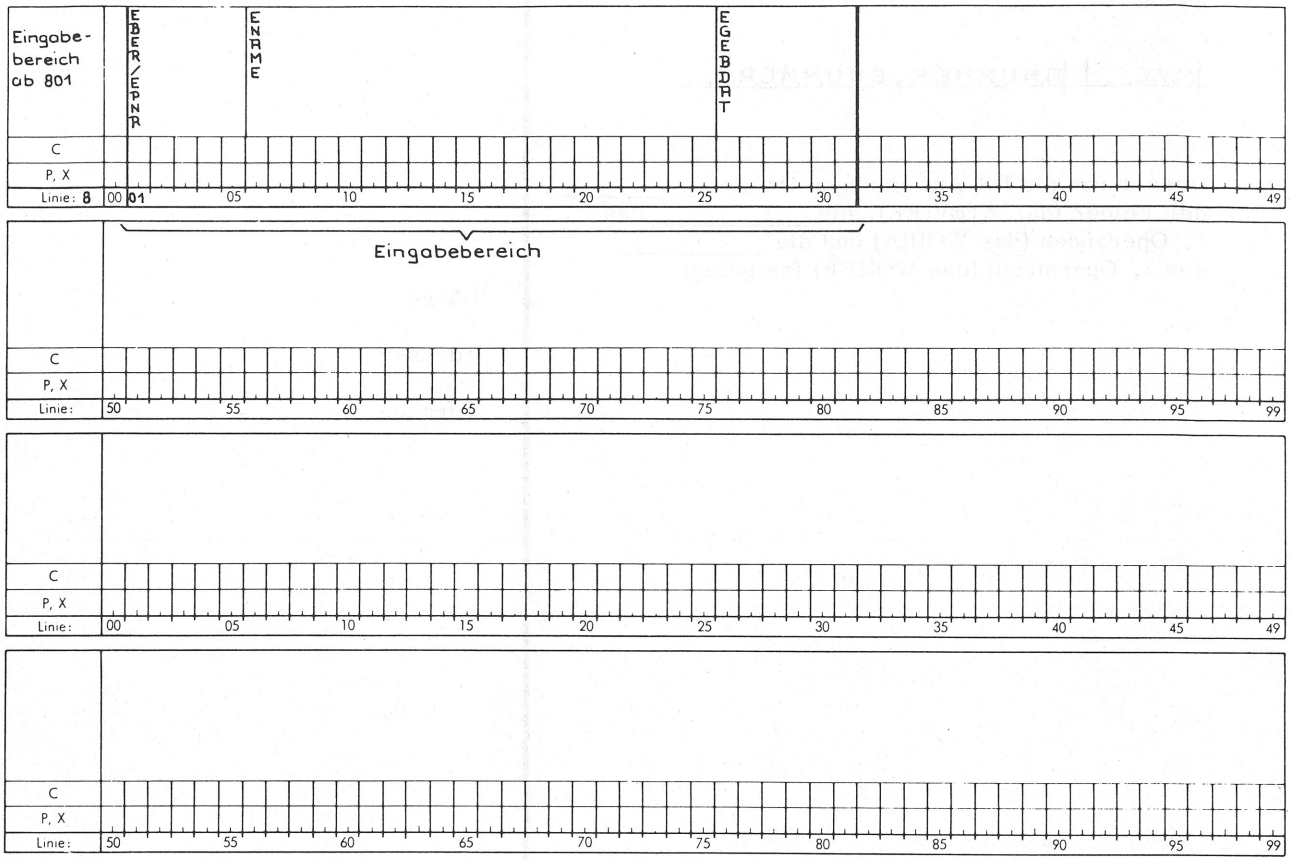
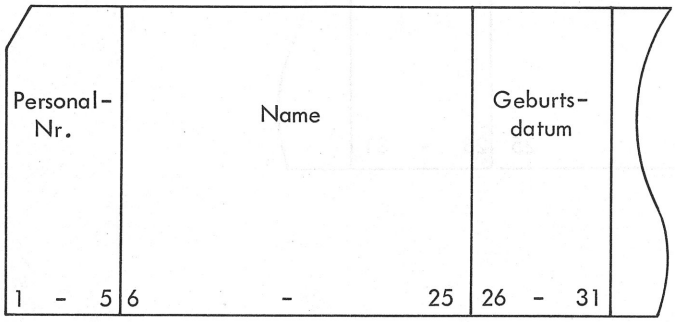

Eingabe-
bereich
ab 801
EBERATENR
ENAME
EGEBDAT
C
P, X
Linie: 8
00 01 05 10 15 20 25 30 35 40 45 49
Eingabebereich
C
P, X
Linie:
50 55 60 65 70 75 80 85 90 95 99
C
P, X
Linie:
00 05 10 15 20 25 30 35 40 45 49
C
P, X
Linie:
50 55 60 65 70 75 80 85 90 95 99

Abb. 8

Die Definition der Eingabe- und Ausgabebereiche

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

Die Speicherbereiche sind frei wählbar. Der Programmierer hat also dafür Sorge zu tragen, daß Platz für die Bereiche und die Felder in ihnen im Hauptspeicher reserviert wird.



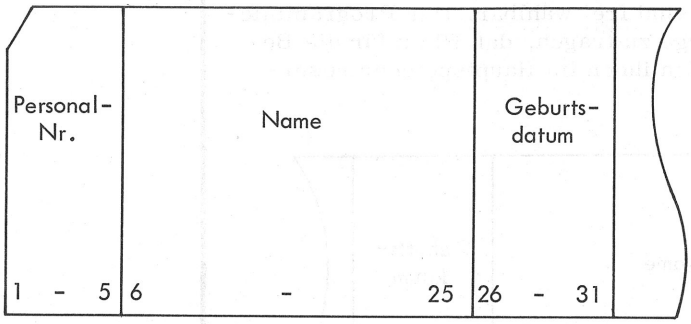
Für die hier abgebildete Personalkarte soll im Hauptspeicher ein Eingabebereich reserviert werden.

Wir benutzen für die Reservierung von Speicherbereichen ein Hilfsmittel, nämlich das Speicherbelegungsblatt (IBM Form 76 683). Entnehmen Sie Ihren Arbeitsmitteln ein solches Speicherbelegungsblatt.

Betrachten Sie bitte Abb. 8.

Das Speicherbelegungsblatt dient uns als Abbild des Hauptspeichers. In Abb. 8 haben wir festgelegt, wo der Eingabebereich für die oben dargestellte Personalkarte im Hauptspeicher beginnen soll. Außerdem haben wir für die Felder in der Lochkarte schon symbolische Namen vergeben.

Dies alles muß dem DVS jetzt durch DS-Anweisungen mitgeteilt werden.


Personal-
Nr.
Name
Geburts-
datum
1 - 5
6 - 25
26 - 31

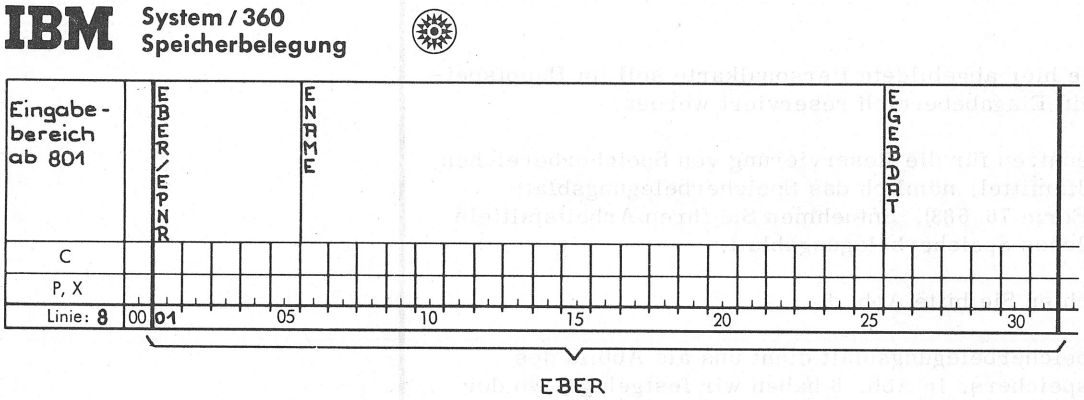

IBM
System / 360
Speicherbelegung
Eingabe-
bereich
ab 801
EBER/EPNR
ENAME
EGEBDAT
C
P, X
Linie: 8
00
01
05
10
15
20
25
30
EBER

Abb. 9

26 Wie Sie aus Abb. 9 ersehen, haben wir den Eingabebereich mit dem symbolischen Namen EBER versehen.

Der Eingabebereich (EBER) soll gemäß unserer Speicherbelegung im Hauptspeicher ab der Stelle __(Z)__ beginnen und insgesamt __(Z)__ Speicherstellen umfassen.

801

31

27 Wie lautet die DS-Anweisung für den 31-stelligen Eingabebereich, der symbolisch als EBER bezeichnet werden soll?

Name	Operation	Operand
1 8	10 14	20
E,B,E,R, . . .	D,S, . .	C,L,3,1, . . .

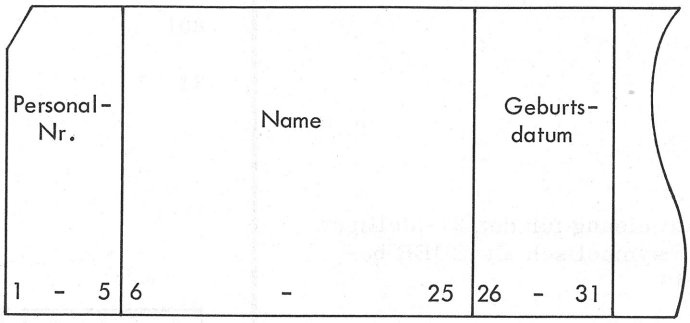
28

Name	Operation	Opera
1 8	10 14	20
E,B,E,R, . . .	D,S, . .	C,L,3,1, . . .

Durch die Anweisung wird der Zuordnungszähler um 31 erhöht.

Wenn EBER die Adresse 801 zugeordnet erhält, dann wird also dem als nächstes zu definierenden Feld vom Zuordnungszähler die Adresse __(Z)__ zugewiesen.

832


Personal-
Nr.
Name
Geburts-
datum
1 - 5
6 - 25
26 - 31

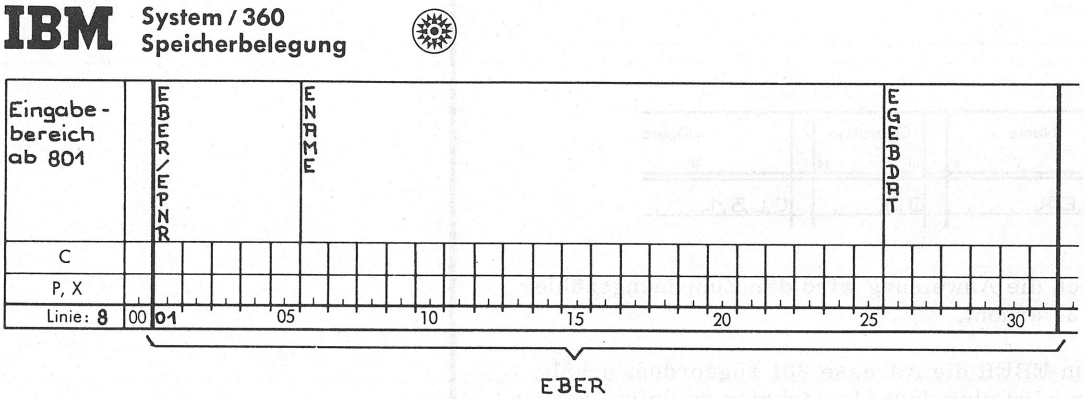

IBM
System / 360
Speicherbelegung
Eingabe-
bereich
ab 801
EBER/EPNR
ENAME
EGEBDAT
C
P, X
Linie: 8
00
01
05
10
15
20
25
30
EBER

Abb. 9

29

Name	Operation	Op
1 8	10 14	20
EBER	DS	CL31

Sie sehen, durch diese Anweisung ist es uns unmöglich, die Felder innerhalb des Eingabebereiches im Hauptspeicher zu reservieren.

Durch die Erhöhung des Zuordnungszählers um 31 würden die Felder EPNR, ENAME, EGEBDAT anschließend an den Eingabebereich reserviert werden. Wir brauchen jedoch, wie in der Speicherbelegung dargestellt, eine Reservierung der Felder EPNR, ENAME und EGEBDAT innerhalb des EBER.

EPNR soll ebenso wie EBER die Adresse ___(Z)___ zugeordnet erhalten.

801

30 Damit sowohl EBER als auch EPNR die gleiche Adresse im Hauptspeicher erhalten, müssen wir verhindern, daß der Zuordnungszähler erhöht wird.

Das geschieht durch diese Anweisung:

Name	Operation	Operan
1 8	10 14	20
EBER	DS	ØCL31

Durch den Operanden 0CL31 wird die _____ des Zuordnungszählers um 31 verhindert.

Erhöhung

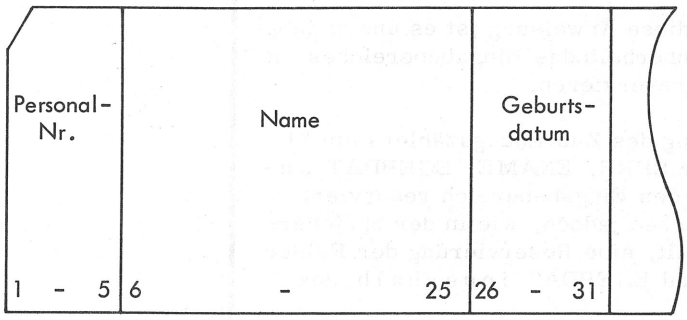
31 Durch den Operanden 0CL31 ist die Länge des Bereiches EBER angegeben. EBER ist ___(Z)___ Stellen groß.

Die Null in 0CL31 bewirkt jedoch, daß der Zuordnungszähler um 0 x 31 erhöht wird.

Das bedeutet:
Dem Bereich EBER ist zwar ein Adress- und ein Längenattribut zugeordnet, der Zuordnungszähler wird jedoch nicht erhöht und somit wird auch im Hauptspeicher kein Platz _____.

31

freigehalten / reserviert


Personal-
Nr.
Name
Geburts-
datum
1 - 5
6 - 25
26 - 31

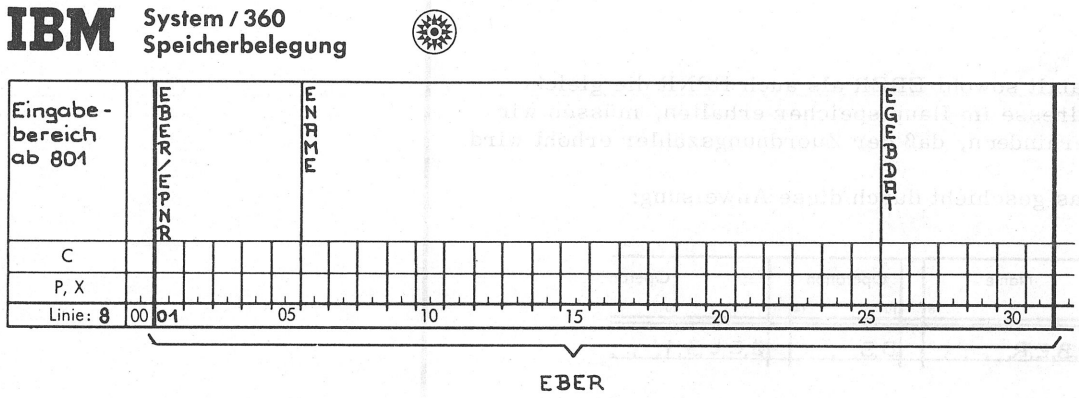

IBM
System / 360
Speicherbelegung
Eingabe-
bereich
ab 801
EBER\EPNR
ENAME
EGEBDAT
C
P, X
Linie: 8
00
01
05
10
15
20
25
30
EBER

Abb. 9

32 Damit die Erhöhung des Zuordnungszählers verhindert, d. h. kein Platz im Hauptspeicher reserviert wird, muß die Definitionsanweisung für den 31-stelligen Eingabebereich (EBER) _____ lauten.

Name	Operation	Operanc
1 8	10 14	20
E,B,E,R, . . .	D,S, . .	Ø,C,L,3,1, . .

33 Wir wollen annehmen, daß das Programm mit den Definitionen des Eingabebereiches und seiner Felder beginnen soll. Der START-Punkt des Programms ist also zugleich die Adresse des Eingabebereichs.

Tragen Sie bitte gemäß der Abb. 9 als erstes die START-Anweisung in ein Programmschema ein.

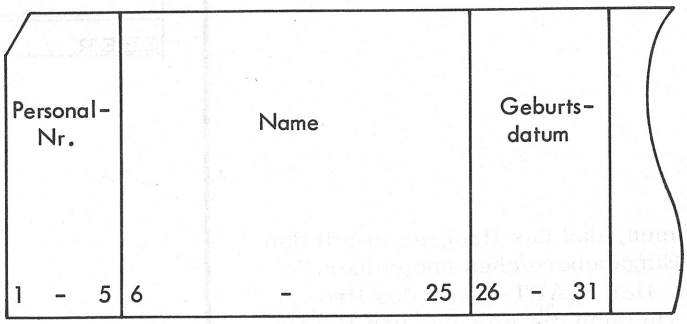
Name	Operation	Operanc
1 8	10 14	20
. . . . .	S,T,A,R,T	8,Ø,1, ...

34 Das Programm soll ab der Stelle 801 im Hauptspeicher gespeichert werden. Deshalb lautet die erste Anweisung:

Name	Operation	Operanc
1 8	10 14	20
. . . . .	S,T,A,R,T	8,Ø,1, ...

Tragen Sie bitte nun als nächstes die Definition des Eingabebereiches EBER in Ihr Programmschema ein, und zwar so, daß der Zuordnungszähler nicht erhöht wird.

Name	Operation	Operanc
1 8	10 14	20
. . . . .	S,T,A,R,T	8,Ø,1, ...
E,B,E,R, . . .	D,S, . .	Ø,C,L,3,1, . .


Personal-
Nr.
Name
Geburts-
datum
1 - 5
6 - 25
26 - 31

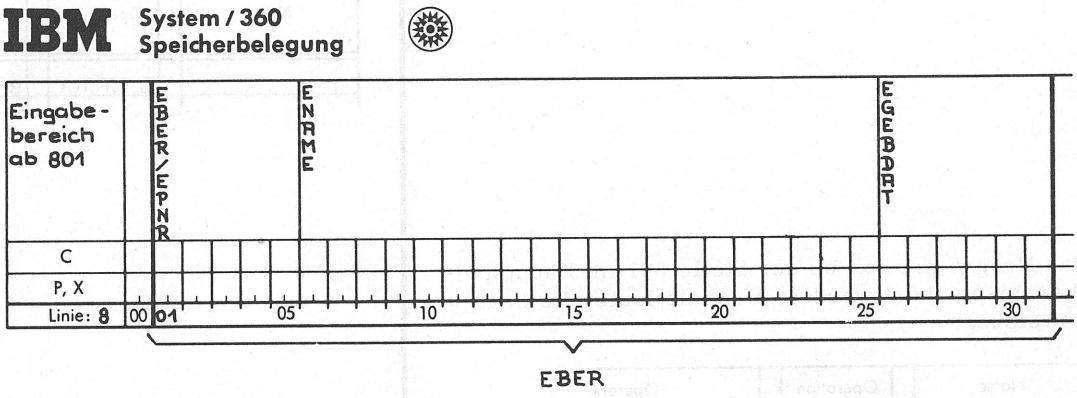

IBM
System / 360
Speicherbelegung
Eingabe-
bereich
ab 801
EBER/EPNR
ENAME
EGEBDAT
C
P, X
Linie: 8
00
01
05
10
15
20
25
30
EBER

Abb. 9

35 Nun müssen nur noch die Felder innerhalb des Eingabebereiches definiert werden. Die Definition dieser Felder muß natürlich so erfolgen, daß der Zuordnungszähler jeweils erhöht wird. Die Speicherplätze für die zu definierenden Felder müssen nacheinander zugeteilt werden.

Fahren Sie bitte fort mit den Definitionen der Felder im Eingabebereich. Benutzen Sie die in der Speicherbelegung (Abb. 9) angegebenen symbolischen Namen und Feldlängen.

Name	Operation	Operanc
1 8	10 14	20
	START	8,0,1
EBER	DS	0CL31
EPNR	DS	CL5
ENAME	DS	CL20
EGEBDAT	DS	CL6

Überprüfen Sie bitte Ihre gesamte Codierung auf ihre Richtigkeit!

Die Summe der 3 Feldlängen = Länge des Bereiches (EBER).

IBM System / 360
Speicherbelegung

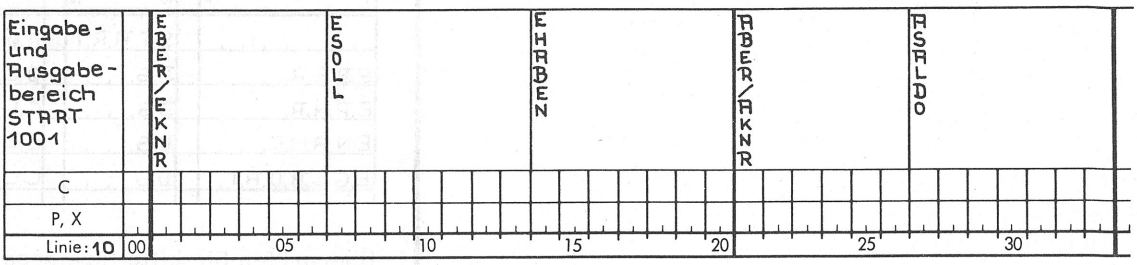


Abb. 10

36 Betrachten Sie bitte Abb. 10. -

An den Eingabebereich (EBER) schließt sich in der Speicherbelegung direkt der Ausgabebereich (ABER) an.

Definieren Sie bitte die Ein-/Ausgabebereiche mit ihren Feldern. Beginnen Sie wiederum mit der START-Anweisung.

Name (1–8)	Operation (10–14)	Operand (20)
	START	1000
EBER	DS	0CL20
EKNR	DS	CL6
ESOLL	DS	CL7
EHABEN	DS	CL7
ABER	DS	0CL13
AKNR	DS	CL6
ASALDO	DS	CL7

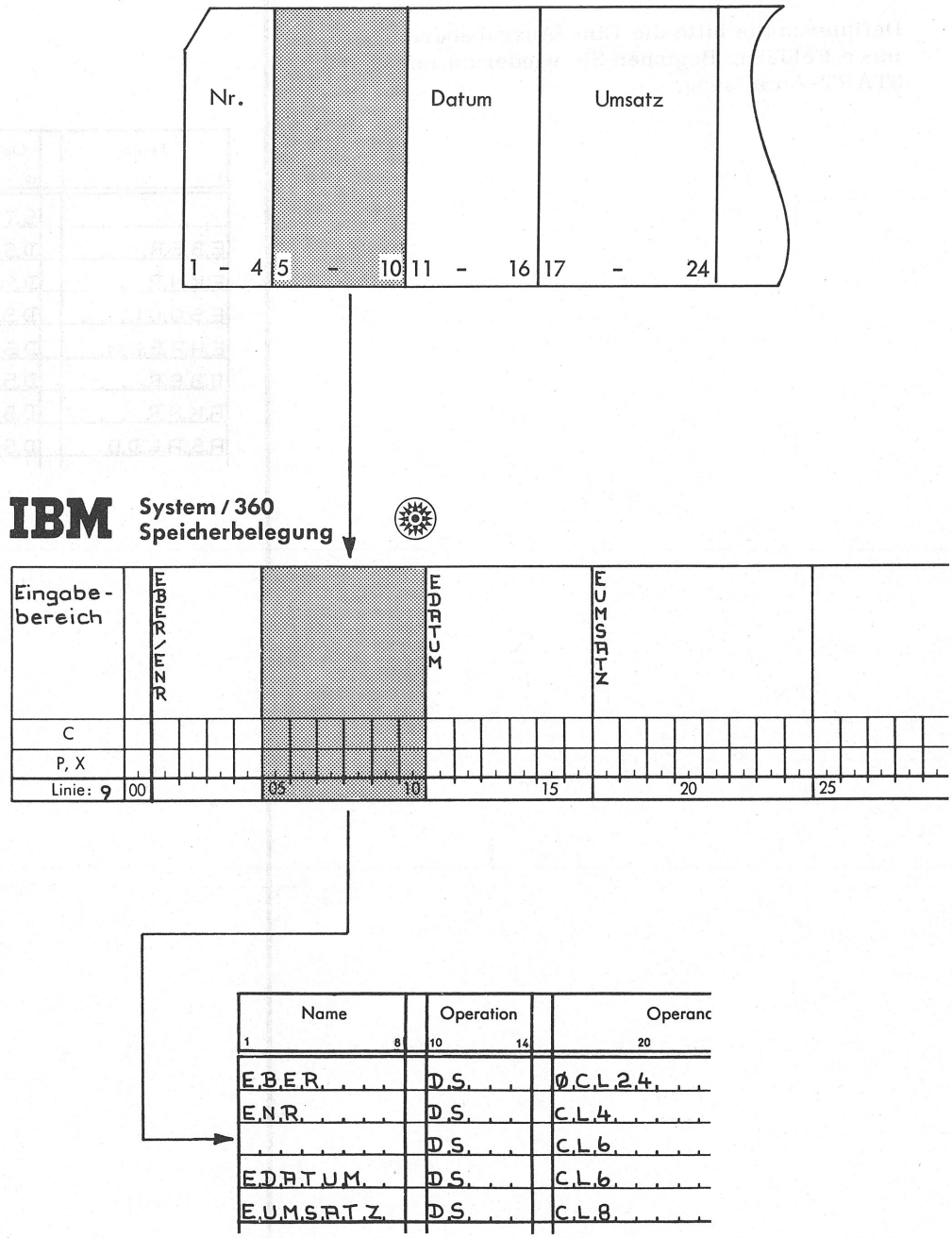

Nr.
Datum
Umsatz
1 - 4
5 - 10
11 - 16
17 - 24
IBM System / 360
Speicherbelegung
Eingabe-bereich
EBER/ENR
EDATUM
EUMSATZ
C
P, X
Linie: 9
00
05
10
15
20
25
Name
Operation
Operanc
1 8
10 14
20
E,B,E,R, . . .
D,S, . .
Ø,C,L,2,4, . .
E,N,R, ...
D,S, . .
C,L,4, ...
. . . . . .
D,S, . .
C,L,6, ...
E,D,A,T,U,M, .
D,S, . .
C,L,6, ...
E,U,M,S,A,T,Z,
D,S, . .
C,L,8, ...

Abb. 11

37 Betrachten Sie bitte Abb. 11. -

Die Daten aus der Lochkarte müssen stellengerecht in den Eingabebereich (EBER) eingelesen werden können. Obwohl sich in den Spalten 5-10 der Lochkarte keine Daten befinden, muß dennoch im Hauptspeicher eine entsprechende Anzahl Speicherstellen freigehalten werden.

Dies geschieht in Abb. 11 durch die Anweisung _____.

38 |.....| |D.S. ...| |C.L.6. ...|

Durch diese Anweisung wird das Längenattribut 6 vergeben.

Da jedoch kein symbolischer Name benutzt wird, wird dem 6-stelligen Feld im Hauptspeicher keine symbolische _____ zugeordnet.

|.....| |D.S. ...| |C.L.6. ...|

Adresse

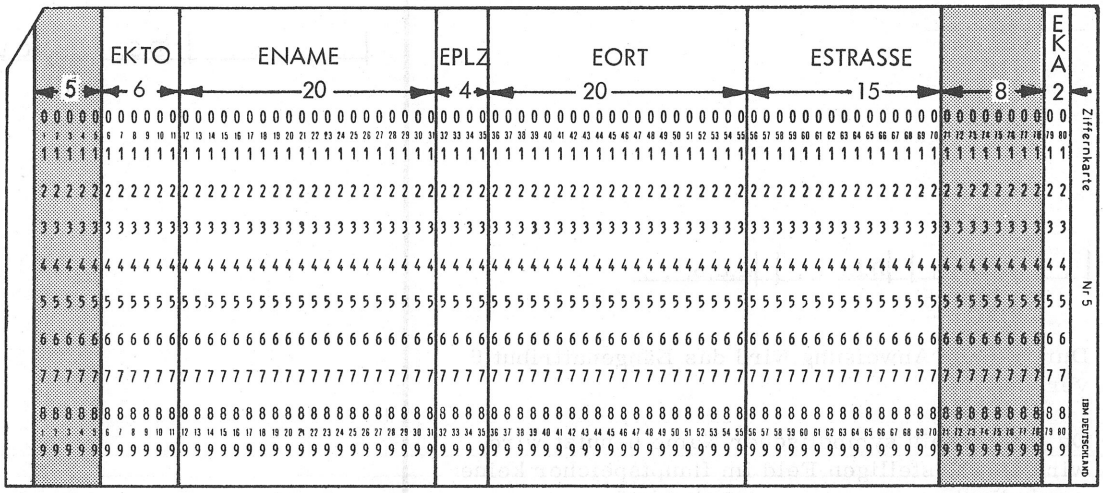

EKTO
ENAME
EPLZ
EORT
ESTRASSE
EKA
5
6
20
4
20
15
8
2
Ziffernkarte
Nr 5
IBM DEUTSCHLAND

Abb. 12

39 Definieren Sie bitte für die Lochkarte in Abb. 12 einen 80-stelligen Eingabebereich (EBER).

Die Feldgrößen und die symbolischen Namen sind vorgegeben. Benutzen Sie bitte ein Programmschema.

Name 1 8	Operation 10 14	Operand 20
E,B,E,R, . . .	D,S, . .	Ø,C,L,8,Ø, . . .
. . . . .	D,S, . .	C,L,5, ...
E,K,T,O, . . .	D,S, . .	C,L,6, ...
E,N,A,M,E, . .	D,S, . .	C,L,2,Ø, ...
E,P,L,Z, . . .	D,S, . .	C,L,4, ...
E,O,R,T, . . .	D,S, . .	C,L,2,Ø, ...
E,S,T,R,A,S,S,E	D,S, . .	C,L,1,5, ...
. . . . .	D,S, . .	C,L,8, ...
E,K,A, ...	D,S, . .	C,L,2, ...

Überprüfen Sie Ihre Codierung genau auf ihre Richtigkeit! Die Summe aller Feldlängen = L des EBER.

	Name 1 8	Operation 10 14	Operan 20
a)	E,N,R, ...	D,S, . .	C,L,5, ...
	. . . . .	. . . . .	. . . . .
b)	. . . . .	D,S, . .	C,L,7, ...
	. . . . .	. . . . .	. . . . .
c)	E,B,E,R, . . .	D,S, . .	Ø,C,L,4,5, . .

Abb. 13

40 In Abb. 13 sind die 3 Formen von DS-Anweisungen, die wir bisher kennengelernt haben, noch einmal zusammengefaßt.

a) Diese Anweisung bewirkt, daß ein ___(Z)___-stelliges Feld im Hauptspeicher reserviert wird. Das Feld ENR erhält aber durch den Zuordnungszähler außer dem Längenattribut auch ein _____ zugeordnet.

b) Durch diese Anweisung wird zwar ein 7-stelliges Feld im Hauptspeicher freigehalten, ein Adressattribut wird diesem Feld jedoch nicht zugeordnet, da kein symbolischer _____ vergeben ist.

c) In dieser Anweisung erfolgt eine Zuordnung von Längen- und Adressattribut. Der Zuordnungszähler wird jedoch nicht erhöht, so daß auch kein Platz im Hauptspeicher _____ wird.

5

Adressattribut

Name

reserviert

Name	Operation	Operanc
1 8	10 14	20
F,E,L,D,A, , ,	D,S, , ,	3,C,L,4, , , ,

IBM System / 360
Speicherbelegung

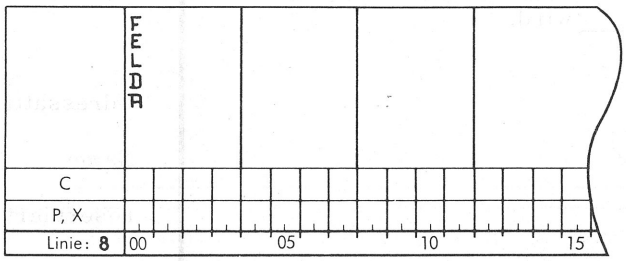
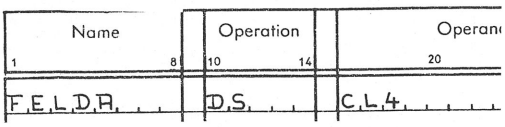
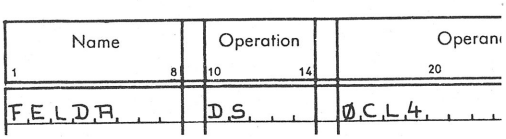


Abb. 14

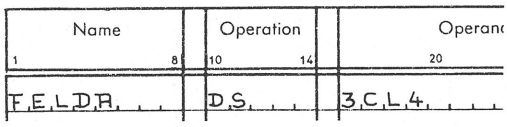
41



Für FELDA wird ein 4 Bytes großes Feld im Hauptspeicher reserviert.



Für FELDA wird 0 mal ein Feld in der Größe von 4 Bytes im Hauptspeicher reserviert. Das heißt, es wird kein Platz im Hauptspeicher reserviert.



Durch diese Anweisung wird 3 mal ein 4 Bytes grosses Feld im Hauptspeicher reserviert.

Die 3 im Operanden 3CL4 bezeichnet man als den D u p l i z i e r f a k t o r. Wie oft für FELDA ein 4 Bytes großes Feld im Hauptspeicher reserviert werden soll, wird durch den _____ angegeben.

Duplizierfaktor

42 Die Angabe eines Duplizierfaktors b e e i n f l u ß t n i c h t das Adress- und Längenattribut eines Feldes im Hauptspeicher.

Betrachten Sie bitte Abb. 14. -

Der symbolische Name FELDA bezeichnet ein Feld im Hauptspeicher, das ab der Adresse __(Z)__ beginnt und __(Z)__ Bytes lang ist.

800

4

43 FELDA erhält also das Adressattribut __(Z)__ und das Längenattribut __(Z)__.

800

4

(Wie es möglich ist, auch die zwei weiteren 4-stelligen Felder anzusprechen, wird noch aufgezeigt).

44 Ein Feld FLD1 sei 5 Stellen groß und es sollen noch weitere 5 Felder davon mit Hilfe des Duplizierfaktors reserviert werden.

Wie lautet die Definitionsanweisung?

```
|F.L.D.1. . . .| |D.S. . .| |6.C.L.S. . .
```

Mit Hilfe des Duplizierfaktors kann also angegeben werden, wie oft ein Feld im Hauptspeicher hintereinander reserviert werden soll.

Der Duplizierfaktor 0 zeigt dem Umwandlungsprogramm an, daß ein Feld nullmal reserviert werden soll. Wir erreichen damit, daß der Zuordnungszähler nicht weitergestellt wird.

Der Duplizierfaktor 6 zeigt dem Umwandlungsprogramm an, daß ein Feld sechsmal hintereinander reserviert werden soll. Der Zuordnungszähler wird durch die Angabe des Duplizierfaktors 6 sechsmal in der Länge des Feldes weitergestellt.

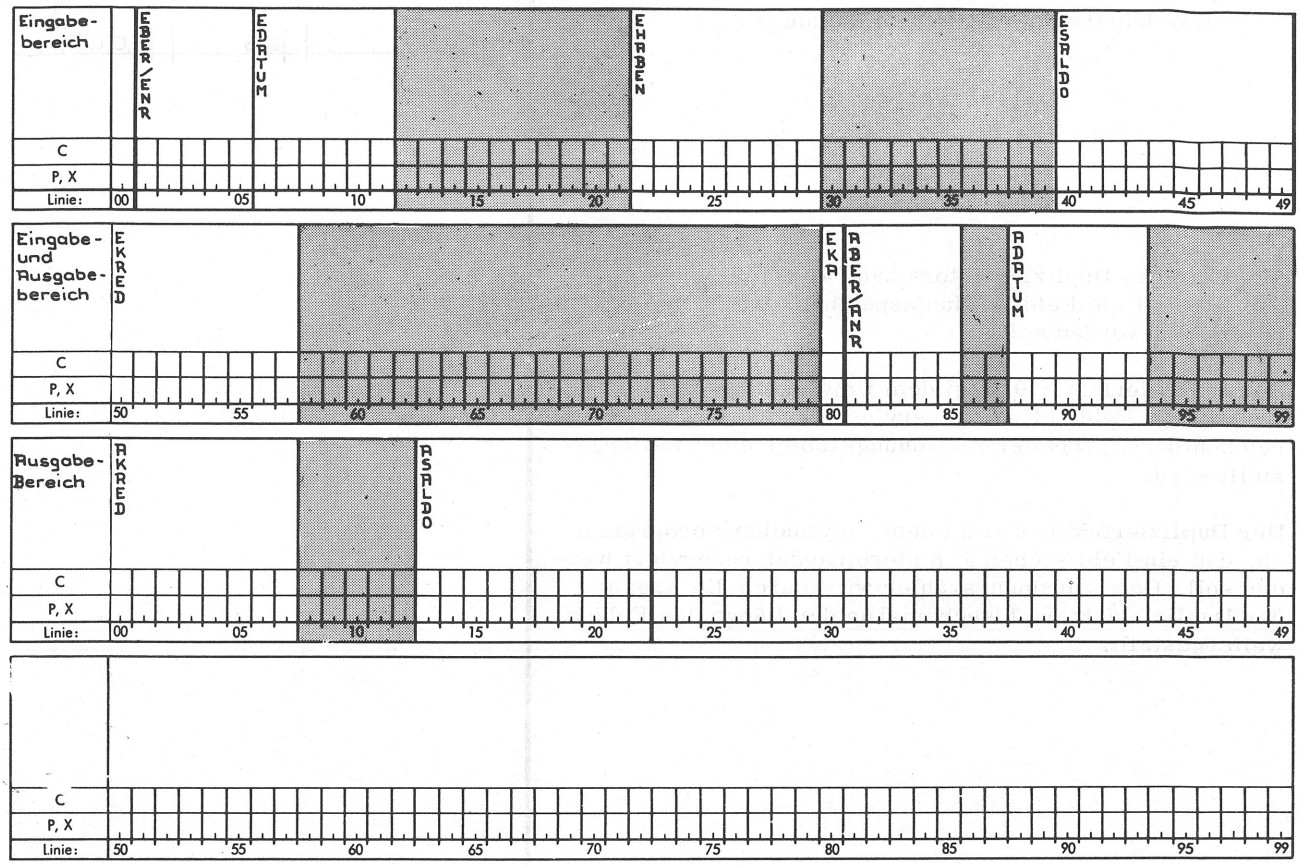

Eingabe-bereich
EBER/ENR
EDATUM
EHABEN
ESALDO
C
P, X
Linie:
00
05
10
15
20
25
30
35
40
45
49
Eingabe- und Ausgabe-bereich
EKRED
EKA
ABER/ANR
ADATUM
50
55
60
65
70
75
80
85
90
95
99
Ausgabe-Bereich
AKRED
ASALDO

Abb. 15

Übungen

Betrachten Sie bitte die Speicherbelegung in Abb. 15. -

Ein 80-stelliger Eingabebereich (EBER) und ein 42-stelliger Ausgabebereich (ABER) und ihre Felder sollen definiert werden.

Tragen Sie bitte erst alle notwendigen Definitionsanweisungen in ein Programmschema ein, bevor Sie Ihre Codierung überprüfen.

Name (1 – 8)	Operation (10 – 14)	Operand (20)
EBER	DS	ØCL8Ø
ENR	DS	CL5
EDATUM	DS	CL6
	DS	CL1Ø
EHABEN	DS	CL8
	DS	CL1Ø
ESALDO	DS	CL1Ø
EKRED	DS	CL8
	DS	CL22
EKA	DS	CL1
ABER	DS	ØCL42
ANR	DS	CL5
	DS	CL2
ADATUM	DS	CL6
	DS	CL6
AKRED	DS	CL8
	DS	CL5
ASALDO	DS	CL1Ø

Band 1, Seite 27

Name	Operation	Operanc
1 8	10 14	20
F.E.L.D.A. . .	D.S. . .	3.C.L.4. . . .

IBM System / 360
Speicherbelegung

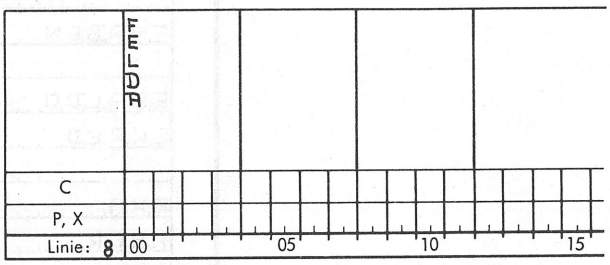


Abb. 1

Die Adressrechnung

Durch Adressrechnung hat der Programmierer die Möglichkeit, jederzeit im Laufe des Programms Adressen im Hauptspeicher relativ zu definierten Symbolen (den symbolischen Adressen) anzusprechen.

Z.B. EBER+4
ABER-25

Hinter der zu verändernden symbolischen Adresse wird durch ein Vorzeichen (+ oder -) und einen Direktwert angegeben, um wieviel die symbolische Adresse erhöht bzw. verringert werden soll. EBER+4 bezeichnet man als einen "zusammengesetzten Ausdruck".

Das Umwandlungsprogramm errechnet sich aus dem zusammengesetzten Ausdruck (Symbol ± Direktwert) die neue Adresse.

1 Betrachten Sie bitte Abb. 1. -

Durch die DS-Anweisung in Abb. 1 werden im Hauptspeicher drei 4 Bytes große Felder freigehalten.

Durch die symbolische Adresse FELDA kann jedoch nur das erste der drei Felder angesprochen werden. Der Zuordnungszähler wird durch den Operanden 3CL4 zwar 3 x um 4 erhöht, eine symbolische Adresse ist jedoch nur für die Speicherstellen 800-803 zugeordnet.

Wir halten fest:

Durch die DS-Anweisung in Abb. 1 ist FELDA als ein Feld im Hauptspeicher definiert, das das Adressattribut __(Z)__ und das Längenattribut __(Z)__ besitzt.

Durch die symbolische Adresse FELDA wird also ein Feld angesprochen, das sich in den Hauptspeicherstellen __(Z)__ bis __(Z)__ befindet.

800

4

800 (bis) 803

Name	Operation	Operanc
1 8	10 14	20
F.E.L.D.A. . .	D.S. . .	3.C.L.4. . . .

IBM System / 360
Speicherbelegung

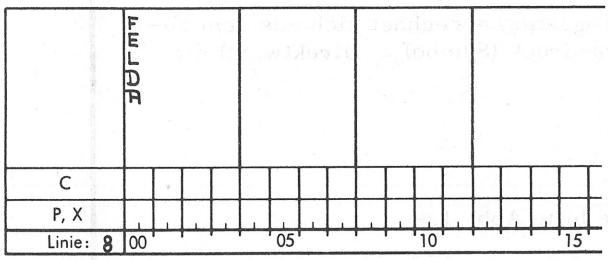


Abb. 1

2 Wie ist es nun möglich, die Adresse 804 relativ zu der symbolischen Adresse FELDA anzusprechen?

Das geschieht durch den zusammengesetzten Ausdruck FELDA+4.

FELDA entspricht 800. Die Adresse 804 ergibt sich also aus der symbolischen Adresse _____ plus der Anzahl der hinzuzuzählenden Bytes, nämlich __(Z)__.

Die Adresse 804 im Hauptspeicher kann also durch den Ausdruck _____ angesprochen werden.

FELDA

4

FELDA+4

3 Der Operand einer Maschineninstruktion soll die Adresse 808 (siehe Speicherbelegung in Abb. 1) ansprechen.

Wie muß der Operand lauten?

FELDA+8

IBM System / 360
Speicherbelegung

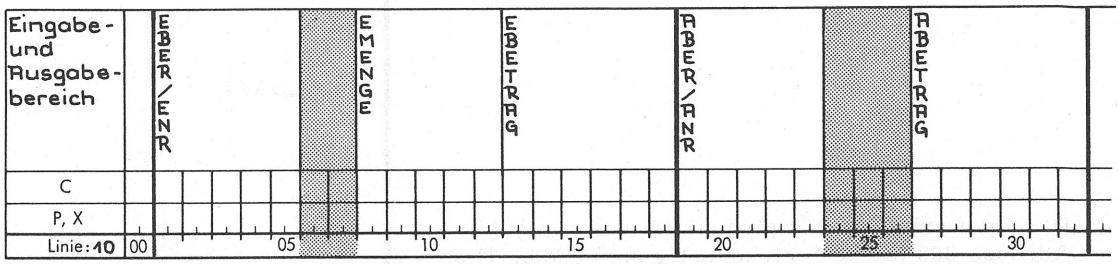


Abb. 2

4 In Abb. 2 ist die Speicherbelegung für einen Eingabe- und einen Ausgabebereich dargestellt.

Die Adresse 1006 soll relativ zu der symbolischen Adresse EBER angesprochen werden. Wie muß der Operand lauten?

EBER+5

5 Durch den Operanden EMENGE-2 wird ebenfalls die Stelle __(Z)__ im Hauptspeicher adressiert.

Adressieren Sie bitte das Byte 1024 relativ zu ABER und zu ABETRAG.

1006

ABER+5

ABETRAG-3

Wir merken uns:

Bei der Adressrechnung wird durch ein Vorzeichen (+ oder -) und die Anzahl der Bytes angegeben, um wieviel die symbolische Adresse erhöht bzw. verringert wird.

Die Spezifikation der Feldlänge

Name	Operation	Operand
1 8	10 14	20
F.E.L.D.A. . .	D.S. . .	C.L.1.5. . . .

Im Normalfall wird die Länge eines Feldes im Hauptspeicher durch die Längenangabe in der DS-Anweisung festgelegt. Durch die obige Codierung erhält das zu reservierende Feld im Hauptspeicher das Längenattribut 15.

Die durch eine DS-Anweisung festgelegte Länge eines Feldes im Hauptspeicher bezeichnet man als die implizite Länge eines Feldes. Das heißt, immer wenn im Laufe des Programms der Operand FELDA benutzt wird, wird diesem Symbol die Länge 15 zugeordnet.

Es ist jedoch durchaus möglich, daß z. B. von einem Feld mit der impliziten Länge 15 nur die ersten 10 Stellen verarbeitet werden sollen:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
F.E.L.D.A. . .	D.S. . .	C.L.1.5. . . . .
. . . . .	. . . .	. . . . .
. . . . .	. . . .	. . . . .
. . . . .	. . . .	. . . . .
. . . . .	M.V.C. .	F.E.L.D.A.(1.Ø.),.K.O.N.S.T.X. .

Durch die DS-Anweisung in dieser Codierung erhält FELDA die implizite Länge 15 zugeordnet. Der 1. Operand der MVC-Instruktion bewirkt jedoch, daß nur die ersten 10 Stellen vom FELDA verarbeitet werden. Die Angabe der Länge in Klammern hinter FELDA bezeichnet man als die explizite Länge des Feldes. Durch die Verwendung der expliziten Längenangabe in dem Operanden FELDA(10) wird in dieser Instruktion FELDA in der Länge von 10 Bytes verarbeitet.

Wir merken uns:

1. Die implizite Länge eines Feldes ist die in der DS-Anweisung definierte Länge.

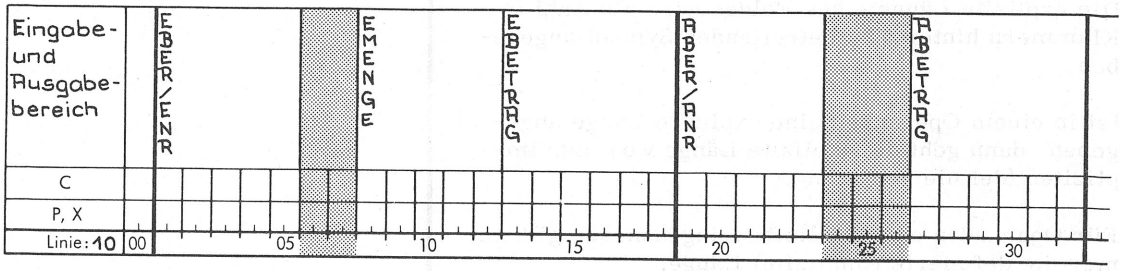
2. Die explizite Länge eines Feldes ist die ausdrücklich im Operanden einer Maschineninstruktion angegebene Länge.

3. Die explizite Länge eines Feldes wird als Zahl in Klammern hinter dem betreffenden Symbol angegeben.

4. Ist in einem Operanden eine explizite Länge angegeben, dann geht die explizite Länge v o r der impliziten (definierten) Länge.

5. Für Operanden ohne explizite Längenangabe gilt immer die definierte (implizite) Länge.

IBM System / 360
Speicherbelegung



Name	Operation	Operand
1 8	10 14	20
E,B,E,R, . . .	D,S, . .	Ø,C,L,1,8, . .
E,N,R, ...	D,S, . .	C,L,5, ...
. . . . .	D,S, . .	C,L,2, ...
E,M,E,N,G,E, .	D,S, . .	C,L,5, ...
E,B,E,T,R,A,G,	D,S, . .	C,L,6, ...
A,B,E,R, . . .	D,S, . .	Ø,C,L,1,4, . .
A,N,R, ...	D,S, . .	C,L,5, ...
. . . . .	D,S, . .	C,L,3, ...
A,B,E,T,R,A,G,	D,S, . .	C,L,6, ...

Abb. 1

1 Betrachten Sie bitte Abb. 1. -

ANR hat die implizite (definierte) Länge __(Z)__.

|M.V.C. . | |A.N.R., E.N.R. .

Durch diese Maschineninstruktion würde also bewirkt werden, daß ANR in der Länge __(Z)__ verarbeitet wird.

5

5

2 EBETRAG hat die implizite Länge __(Z)__.

Durch die Angabe EBÈTRAG(5) im Operanden einer Maschineninstruktion wird jedoch EBETRAG nur in der Ausdehnung von __(Z)__ Bytes verarbeitet.

Denn:

_____ Länge geht vor _____ Länge !

6

5

explizite

impliziter

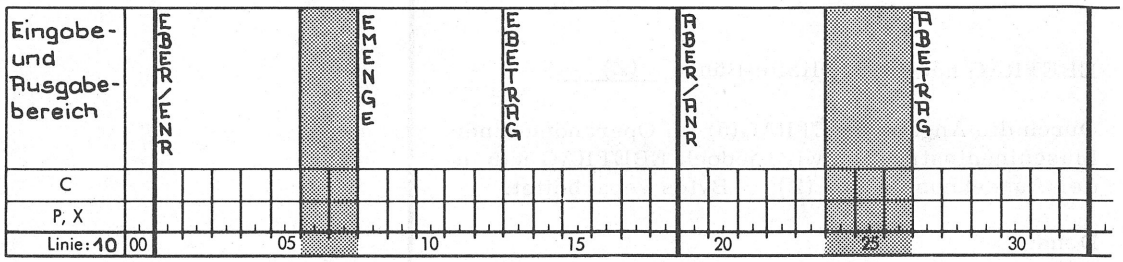
3 In der Speicherbelegung der Abb. 1 schließen die beiden Felder EMENGE und EBETRAG direkt aneinander an.

Stellen Sie sich bitte vor, man wollte diese beiden Felder im Hauptspeicher mittels expliziter Längenangabe durch e i n e n Operanden ansprechen.

Wie muß dieser Operand lauten?

EMENGE(11)

IBM System / 360
Speicherbelegung



Name	Operation	Operand
1 8	10 14	20
EBER	DS	ØCL18
ENR	DS	CL5
	DS	CL2
EMENGE	DS	CL5
EBETRAG	DS	CL6
ABER	DS	ØCL14
ANR	DS	CL5
	DS	CL3
ABETRAG	DS	CL6

Abb. 1

4 Durch den Operanden EBETRAG(5) werden nur die e r s t e n 5 Stellen des Feldes im Hauptspeicher verarbeitet !

EBETRAG ist die symbolische Bezeichnung für die Adresse __(Z)__ im Hauptspeicher (siehe Speicherbelegung in Abb. 1).

EBETRAG(5) bedeutet also:
Ab der Adresse 1013 im Hauptspeicher sollen __(Z)__ Bytes verarbeitet werden.

Der Operand EBETRAG(5) bewirkt also, daß die Bytes __(Z)__ bis __(Z)__ im Hauptspeicher verarbeitet werden.

1013

5

1013 (bis) 1017

5 Welche Speicherstellen (vgl. Speicherbelegung in Abb. 1) werden durch die folgenden Operanden angesprochen?

a) ABETRAG(4)

b) EBER(7)

c) ANR(8)

d) ENR(18)

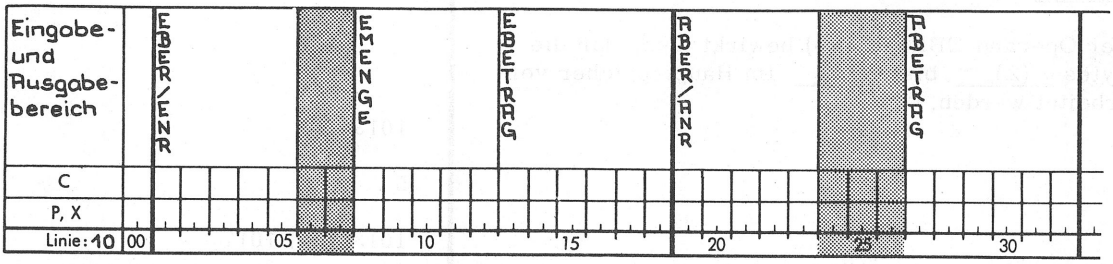
a) 1027 bis 1030

b) 1001 bis 1007

c) 1019 bis 1026

d) 1001 bis 1018

IBM System / 360
Speicherbelegung



Name	Operation	Operand
1 8	10 14	20
E.B.E.R.	D.S.	Ø.C.L.1.8.
E.N.R.	D.S.	C.L.5.
	D.S.	C.L.2.
E.M.E.N.G.E.	D.S.	C.L.5.
E.B.E.T.R.A.G.	D.S.	C.L.6.
A.B.E.R.	D.S.	Ø.C.L.1.4.
A.N.R.	D.S.	C.L.5.
	D.S.	C.L.3.
A.B.E.T.R.A.G.	D.S.	C.L.6.

Abb. 1

6 Wie können wir nun dafür sorgen, daß nur die 4 letzten Stellen von EBETRAG, also die Speicherstellen 1015 bis 1018, verarbeitet werden?

Hierzu müssen wir zunächst durch Adressrechnung veranlassen, daß die Stelle 1015 als Adresse angesprochen wird. Das geschieht durch den Operanden EBETRAG+ __(Z)__ .

Wenn EBETRAG+2 als Operand verwendet wird, dann wird ab der Stelle __(Z)__ ein 6 Bytes großes Feld verarbeitet.

Denn auch der Operand EBETRAG+2 erhält -wie EBETRAG selbst - die (implizite, explizite) _____ Länge __(Z)__ zugeordnet.

2

1015

implizite

6

7 Da wir nur 4 Stellen ab der Adresse 1015 (EBETRAG+2) verarbeiten wollen, müssen wir auch die implizite Länge durch eine explizite Längenangabe ersetzen:

EBETRAG+2(4)

Dieser Operand bedeutet:

Das zu verarbeitende Feld im Hauptspeicher beginnt bei EBETRAG+2, d.h. ab der Adresse __(Z)__, und ist __(Z)__ Bytes groß.

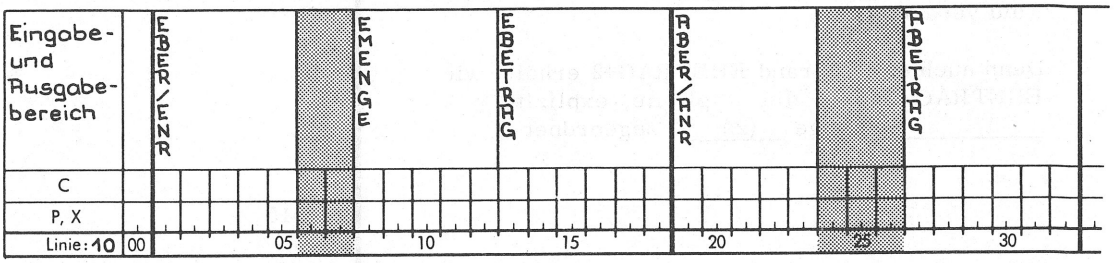
1015

4

8 Durch den Operanden EBETRAG+2(4) ist ein Feld im Hauptspeicher angegeben, das sich in den Speicherstellen __(Z)__ bis __(Z)__ (siehe Speicherbelegung) befindet.

1015 (bis) 1018

IBM System / 360
Speicherbelegung



Name	Operation	Operand
1 8	10 14	20
E.B.E.R. . . .	D.S. . .	Ø.C.L.1.8. . .
E.N.R.	D.S. . .	C.L.5. . . . .
. . . . .	D.S. . .	C.L.2. . . . .
E.M.E.N.G.E. .	D.S. . .	C.L.5. . . . .
E.B.E.T.R.A.G.	D.S. . .	C.L.6. . . . .
A.B.E.R. . . .	D.S. . .	Ø.C.L.1.4. . .
A.N.R.	D.S. . .	C.L.5. . . . .
. . . . .	D.S. . .	C.L.3. . . . .
A.B.E.T.R.A.G.	D.S. . .	C.L.6. . . . .

Abb. 1

9 Welche Speicherstellen (siehe Speicherbelegung in Abb. 1) werden durch die folgenden Operanden angesprochen?

a) EBER+3(2)

b) ABETRAG+1(5)

c) EBETRAG-3(3)

a) 1004 bis 1005

b) 1028 bis 1032

c) 1010 bis 1012

10 Bilden Sie nun die Operanden, die die folgenden Stellen in der Speicherbelegung der Abb. 1 ansprechen. Benutzen Sie die in Klammern angegebenen Symbole.

a) 1002 bis 1005 (ENR)

b) 1006 bis 1007 (EBER)

c) 1010 bis 1012 (EMENGE)

d) 1024 bis 1026 (ABETRAG)

a) ENR+1(4)

b) EBER+5(2)

c) EMENGE+2(3)

d) ABETRAG-3(3)

Wir merken uns:

Bei der Adressrechnung wird durch ein Vorzeichen (+ oder -) und die Anzahl der Bytes angegeben, um wieviel die symbolische Adresse erhöht bzw. verringert wird:

z. B. EBER+10
ABER-8

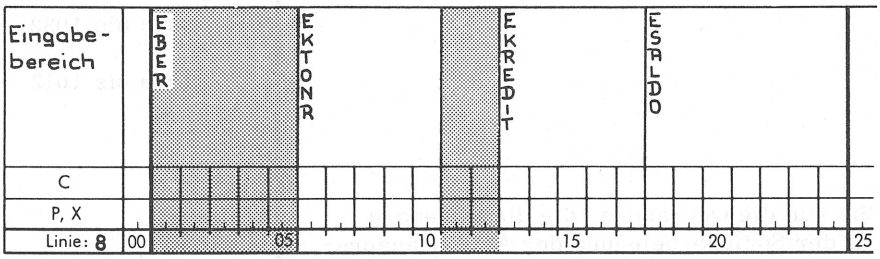
Die explizite Länge eines Feldes wird als Zahl in Klammern hinter dem betreffenden Symbol angegeben:

z. B. EBER(5)
ABER(6)

Beides, Adressrechnung und explizite Längenangabe, kann kombiniert werden, indem zuerst die Adresse und daran ohne Zwischenraum anschließend die explizite Länge angegeben wird:

z. B. EBER+10(1)
ABER-8(2)

IBM System / 360
Speicherbelegung



Name	Operation	Operand
1 8	10 14	20
E.B.E.R. . . .	D.S. . .	Ø.C.L.2.4. . .
. . . . .	D.S. . .	C.L.5. . . . .
E.K.T.O.N.R. .	D.S. . .	C.L.5. . . . .
. . . . .	D.S. . .	C.L.2. . . . .
E.K.R.E.D.I.T.	D.S. . .	C.L.5. . . . .
E.S.A.L.D.O. .	D.S. . .	C.L.7. . . . .

Abb. 2

Übungen zur Adressrechnung und zur expliziten Längenangabe

Die folgenden Übungen beziehen sich auf die Speicherbelegung der Abb. 2. Beantworten Sie bitte erst alle Fragen auf einem Extrablatt, ehe Sie Ihre Antworten überprüfen.

1. Adressieren Sie bitte die Speicherstellen 801 bis 805 mit Hilfe des Symbols EKTONR.
   Wie muß der Operand lauten ?

2. Von dem Feld mit der symbolischen Adresse EKTONR sollen nur die Stellen 806 bis 809 angesprochen werden. Wie muß der Operand lauten ?

3. Mit Hilfe des Symbols EKREDIT sollen die Speicherstellen 813 bis 824 angesprochen werden.
   Wie muß der Operand lauten ?

4. Welche Stellen im Hauptspeicher (siehe Speicherbelegung) werden durch die folgenden Operanden angesprochen ?

   a) ESALDO+2(5)

   b) EKREDIT+3(2)

   c) EKREDIT-5(3)

5. Die folgenden Hauptspeicherstellen (siehe Speicherbelegung) sollen angesprochen werden. Benutzen Sie die in Klammern angegebenen Symbole zur Bildung der Operanden.

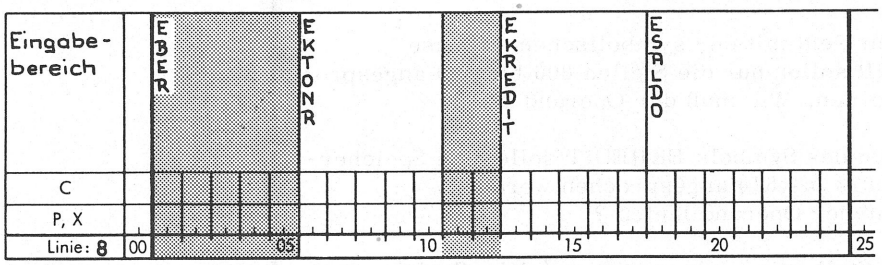
   a) 815 bis 817 (ESALDO)

   b) 815 (EKREDIT)

   c) 811 bis 812 (EBER)

   d) 801 bis 805 (EBER)

IBM System / 360
Speicherbelegung



Name	Operation	Operanc
1 8	10 14	20
E.B.E.R. . . .	D.S. . .	Ø.C.L.2.4. . .
. . . . .	D.S. . .	C.L.5. . . . .
E.K.T.O.N.R. .	D.S. . .	C.L.5. . . . .
. . . . .	D.S. . .	C.L.2. . . . .
E.K.R.E.D.I.T.	D.S. . .	C.L.5. . . . .
E.S.A.L.D.O. .	D.S. . .	C.L.7. . . . .

Abb. 2

1. EKTONR-5 (eine explizite Längenangabe wird nicht benötigt, da die implizite Länge schon mit CL5 definiert ist)

2. EKTONR(4)

3. EKREDIT(12)

4. a) 820 bis 824

   b) 816 bis 817

   c) 808 bis 810

5. a) ESALDO-3(3)

   b) EKREDIT+2(1)

   c) EBER+10(2)

   d) EBER(5)



Band 4, Seite 23 (MVC, 2. Teil)

IBM System / 360
Speicherbelegung

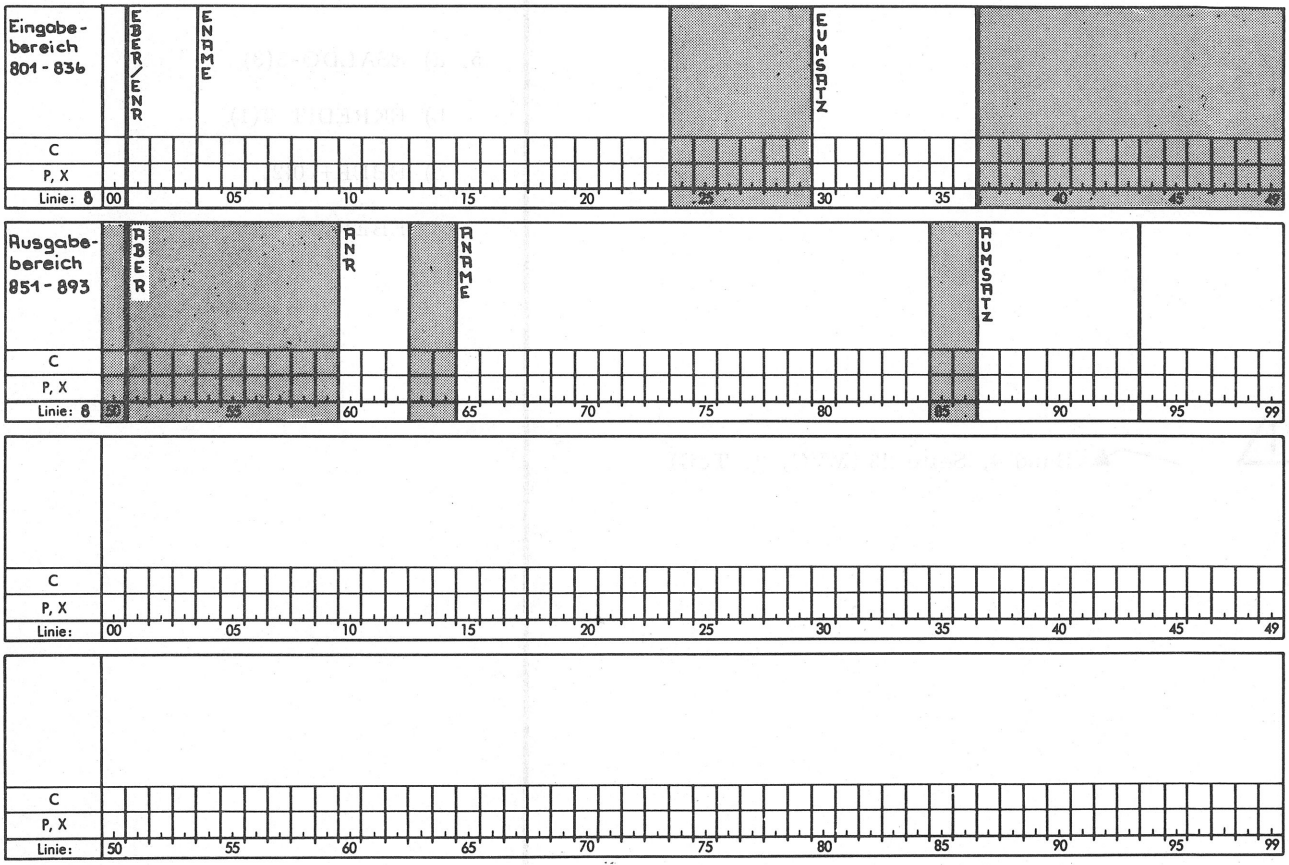


Abb. 1

FRAGEN ZUM KAPITEL III

Prüfen Sie nun bitte anhand der folgenden Fragen, ob Sie den Lernstoff dieses Kapitels verstanden haben.

Beantworten Sie zunächst alle Fragen auf einem Extrablatt (bzw. dem Programmschema). Erst nach der Beantwortung aller Fragen sollen Sie Ihre Antworten und Ergebnisse auf ihre Richtigkeit überprüfen.

1. Durch die START-Anweisung wird der Zuordnungszähler auf eine _____ eingestellt.

2. Die END-Anweisung bezeichnet für die Umwandlung das _____ des symbolischen Programms.

3. Welche der folgenden Aussagen ist falsch ?

   A Mit Hilfe der Assemblerinstruktion ORG wird der Zuordnungszähler verändert.

   B | ORG | 950 | Durch diese Anweisung wird der Zuordnungszähler auf 950 gesetzt.

   C Der Zuordnungszähler darf nur abhängig zu einem vorher definierten Symbol oder zur Sternadresse (* = momentaner Wert) gesetzt (d.h. erhöht oder verringert) werden.

4. Abb. 1 zeigt eine Speicherbelegung mit einem Eingabebereich (801 - 836) und einem Ausgabebereich (851 - 893).

   Entnehmen Sie bitte Ihren Arbeitsmitteln ein Programmschema.

   a) Definieren Sie zunächst den Eingabebereich mit seinen Feldern.

   b) Wenn Sie den Eingabebereich und seine Felder definiert haben, soll der Zuordnungszähler durch eine ORG-Anweisung relativ zu EBER auf 851 erhöht werden.

   c) Definieren Sie daran anschließend den Ausgabebereich und seine Felder.

IBM System / 360
Speicherbelegung

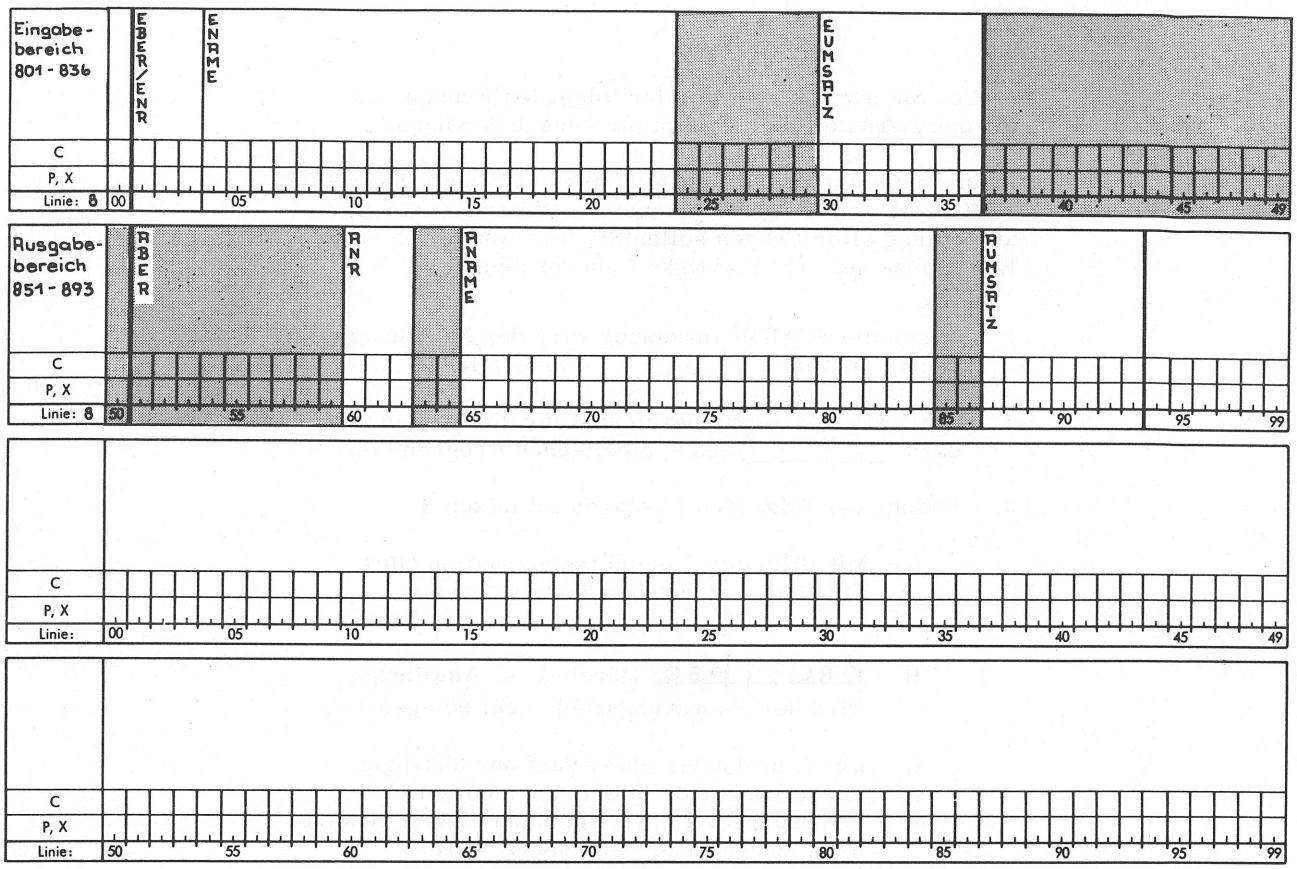


Abb. 1

5. Wie müssen die Operanden von Maschineninstruktionen lauten, die folgende Speicherstellen (siehe Speicherbelegung in Abb. 1) ansprechen ? Verwenden Sie bitte die in Klammern angegebenen Symbole:

   a) 830 - 834 (EUMSATZ)

   b) 801 - 803 (EBER)

   c) 860 - 864 (ABER)

   d) 863 - 864 (ANAME)

Überprüfen Sie nun auf der folgenden Seite Ihre Antworten und Ergebnisse.

ANTWORTEN ZUM KAPITEL III

Wenn Ihnen Fehler unterlaufen sind, dann beseitigen Sie erst alle Unklarheiten, ehe Sie fortfahren. Wiederholen Sie gegebenenfalls die Abschnitte, die Ihnen Schwierigkeiten bereiten (Seitenangaben in Klammern hinter den Antworten).

1. Durch die START-Anweisung wird der Zuordnungszähler auf eine Startadresse (oder Anfangsadresse) eingestellt.

   (Seite 85 - 89)

2. Die END-Anweisung bezeichnet für die Umwandlung das Ende des symbolischen Programms.

   (Seite 91 - 93)

3. Falsch ist:

   B ORG. 950. Durch diese Anweisung wird der Zuordnungszähler auf 950 gesetzt.

   Denn:

   Der Zuordnungszähler kann nur abhängig von einem schon vorher definierten Symbol oder abhängig von der Sternadresse gesetzt, d. h. erhöht oder verringert werden.

   (Seite 95 - 101)

IBM System / 360
Speicherbelegung

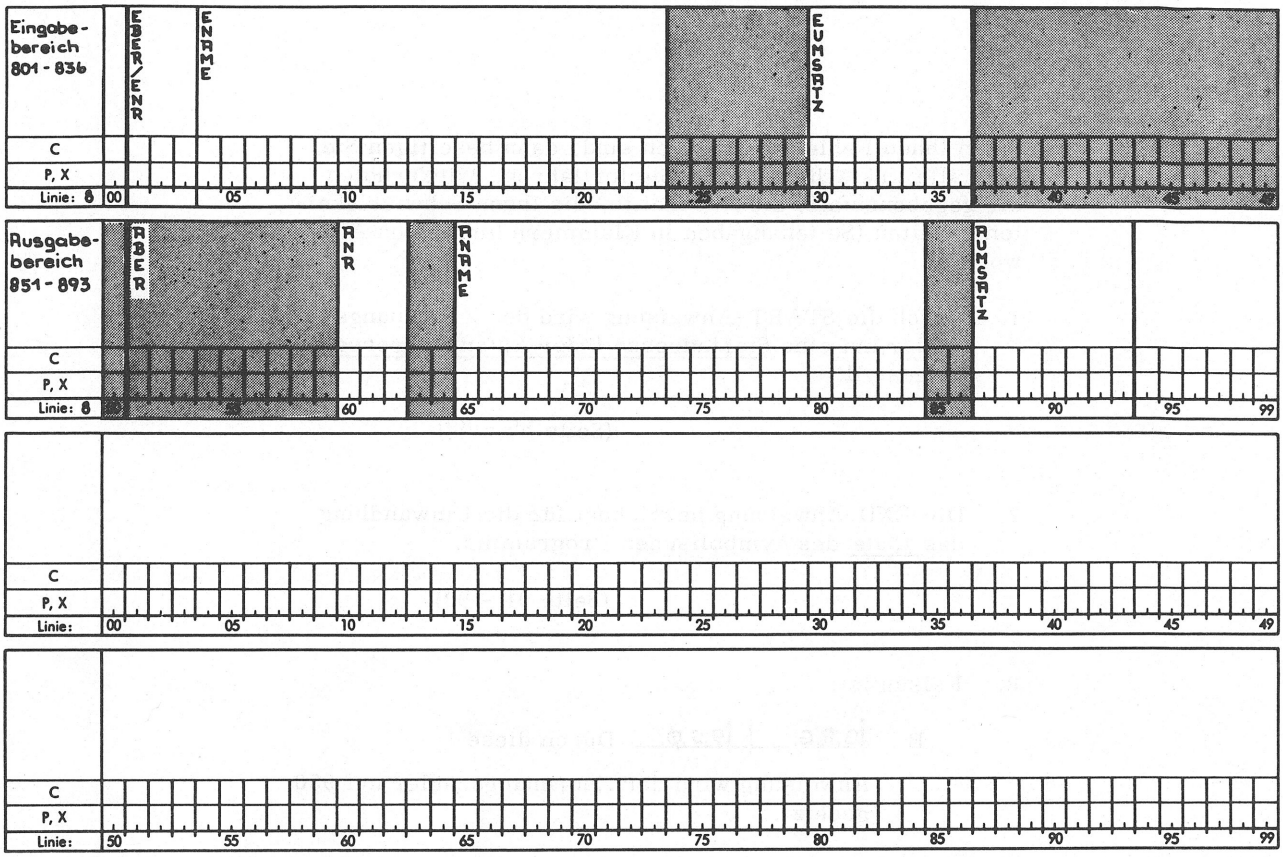


Abb. 1

4.

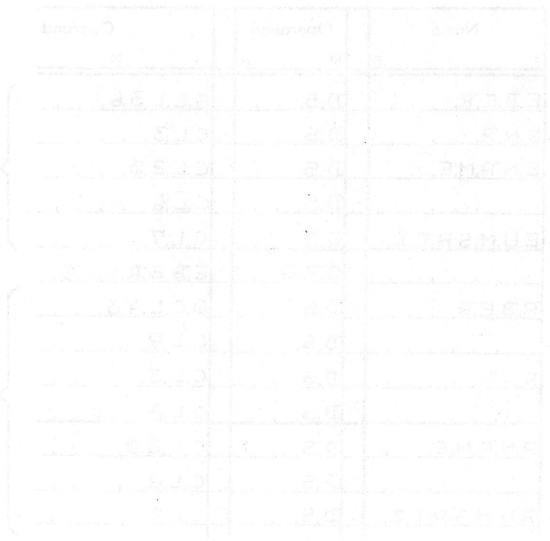
	Name (1 – 8)	Operation (10 – 14)	Operand (20)	
a)	EBER	DS	Ø,CL36	(Seite 131 - 153)
	ENR	DS	CL3	
	ENAME	DS	CL2Ø	
		DS	CL6	
	EUMSATZ	DS	CL7	
b)		ORG	EBER+5Ø	
c)	ABER	DS	ØCL43	(Seite 131 - 153)
		DS	CL9	
	ANR	DS	CL3	
		DS	CL2	
	ANAME	DS	CL2Ø	
		DS	CL2	
	AUMSATZ	DS	CL7	

5. a) EUMSATZ(5)

b) EBER(3) (Seite 155 - 171)

c) ABER+9(5)

d) ANAME-2(2)



KAPITEL IV

DIE DEFINITION DER KONSTANTEN

Lernstoff:

Die Zeichenkonstante - C

Die gepackte Konstante - P

Die Hexadezimalkonstante - X

Die Wortkonstante - F

Die Halbwortkonstante - H

Die Adresskonstante - A

Nachtrag zur DS-Anweisung

DC

DEFINIEREN KONSTANTE

(Define Constant)

Durch die Assemblerinstruktionsanweisung DC können konstante Daten definiert werden. Die durch die DC-Anweisung definierten Konstanten werden zusammen mit dem umgewandelten Programm in den Hauptspeicher geladen.

DIE ZEICHENKONSTANTE - C

Beispiel:

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–25–30)
KDATUM	DC	C'1. MAI 1975'

1. Der symbolische OP-Schlüssel für das Definieren einer Konstanten lautet DC.
2. Das Symbol im Namenfeld der DC-Instruktion (z.B. KDATUM) ist der Name der Konstanten. Dieser symbolische Name bezeichnet die Adresse der Konstanten im Hauptspeicher. Das Namenfeld kann auch unbenutzt bleiben.
3. Der Operand gibt an:
   a) die Art der Konstanten (C = Zeichenkonstante)
   b) den Inhalt der Konstanten in Hochkommas (z.B. '1. MAI 1975')
4. Die Zeichenkonstante darf maximal 256 Stellen groß sein.
5. Für die Zeichenkonstante selbst (z.B. 1. MAI 1975) dürfen Buchstaben, Ziffern, Sonderzeichen sowie Leerstellen verwendet werden.

Abb. 1

Wir befassen uns jetzt mit der Definition von Konstanten.

Als erstes behandeln wir die Definition von Zeichenkonstanten. - Die Zeichenkonstante dient der Ausgabe und Verarbeitung von Daten, die nicht aus Lochkarten kommen und auch nicht errechnet werden. Zeichenkonstanten werden z. B. für Vergleiche im Hauptspeicher und für Überschriften auf Listen benötigt.

Lesen Sie nun bitte zunächst den Text in Abb. 1 aufmerksam durch und wenden Sie sich dann den folgenden Lernelementen zu.

1 Für die Assemblerinstruktion "Definieren Konstante" wird der symbolische OP-Schlüssel _____ benutzt.

Der Name der Konstanten ist ein Symbol, das die _____ der Konstanten im Hauptspeicher bezeichnet.

DC

Adresse

2 Wie bei der DS-Anweisung wird auch dem Symbol im Namenfeld der DC-Anweisung eine Adresse zugeordnet.

In der Codierung der Abb. 1 ist der Name _____ die symbolische Adresse des (rechten, linken) _____ Bytes der Konstanten 1. MAI 1975 im Hauptspeicher.

KDATUM

linken

3 Wir wenden uns nun dem Operanden der DC-Anweisung in Abb. 1 zu. -

Es gibt verschiedene Arten von Konstanten. Der Buchstabe C im Operanden gibt an, daß es sich hier um eine _____ handelt.

Das Feld KDATUM erhält den Inhalt _____.

Zeichenkonstante

1. MAI 1975

DC

DEFINIEREN KONSTANTE

(Define Constant)

Durch die Assemblerinstruktionsanweisung DC können konstante Daten definiert werden. Die durch die DC-Anweisung definierten Konstanten werden zusammen mit dem umgewandelten Programm in den Hauptspeicher geladen.

DIE ZEICHENKONSTANTE - C

Beispiel:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
KDATUM	DC	C'1. MAI 1975'

1. Der symbolische OP-Schlüssel für das Definieren einer Konstanten lautet DC.
2. Das Symbol im Namenfeld der DC-Instruktion (z.B. KDATUM) ist der Name der Konstanten. Dieser symbolische Name bezeichnet die Adresse der Konstanten im Hauptspeicher. Das Namenfeld kann auch unbenutzt bleiben.
3. Der Operand gibt an:
   a) die Art der Konstanten (C = Zeichenkonstante)
   b) den Inhalt der Konstanten in Hochkommas (z.B. ' 1. MAI 1975 ')
4. Die Zeichenkonstante darf maximal 256 Stellen groß sein.
5. Für die Zeichenkonstante selbst (z.B. 1. MAI 1975) dürfen Buchstaben, Ziffern, Sonderzeichen sowie Leerstellen verwendet werden.

Abb. 1

4 Wir merken uns:

Der Konstantentyp (Konstantenart) wird bei der Zeichenkonstanten im Operanden durch den Buchstaben _____ angegeben.

C

Das C ist die Abkürzung von engl. Character (d. h. Zeichen). Es zeigt dem Umwandlungsprogramm an, daß jedes Zeichen (so auch Blanks) der in Hochkommas gesetzten Konstanten in einem Byte zu speichern ist.

Durch die DC-Anweisung in Abb. 1 wird dafür gesorgt, daß die Konstante mit der symbolischen Adresse KDATUM in __(Z)__ hintereinanderfolgenden Bytes im Hauptspeicher gespeichert wird.

11

5 Die Zeichenkonstante darf maximal __(Z)__ Stellen groß sein.

Reicht eine Zeile im Programmschema nicht aus, dann kann der Operand in der nächsten Zeile ab der Spalte 16 fortgesetzt werden. Die erste der beiden Zeilen muß in diesem Fall in der Stelle 72 durch ein beliebiges Zeichen (außer Blank) gekennzeichnet werden.

256

6

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
K,O,N,S,T,1, ,	D,C, , ,	C,',R,A,B,A,T,T, ,=, ,1,Ø,%,',

Durch diese DC-Anweisung erhält die Konstante RABATT = 10 % die symbolische Adresse _____ zugeordnet.

KONST1

Das C im Operanden teilt dem Umwandlungsprogramm mit, daß es sich bei der in Anführungszeichen gesetzten Konstanten um eine _____ handelt.

Zeichenkonstante

Jedes Zeichen (einschließlich Blank) innerhalb der Zeichenkonstanten belegt 1 Byte im Hauptspeicher.

Für die Konstante RABATT = 10 % wird also ein Feld in der Länge von __(Z)__ Bytes im Hauptspeicher reserviert.

12

IBM System / 360
Speicherbelegung

Zeichenkonstanten		KCLISTE										KC150				KC1000								
C		L	I	S	T	E		N	R	.	3	1	5	0	1	.	0	0	0	,	0	0		
P, X																								
Linie: 9	00					05					10					15					20			

Abb. 2

7 Betrachten Sie bitte Abb. 2. -

Wie die zu reservierenden Felder und Bereiche, die durch DS-Anweisungen definiert werden, tragen wir auch die zu definierenden Konstanten in das Speicherbelegungsblatt ein.

Für die Eintragung der Zeichenkonstanten ist die obere Reihe von Kästchen vorgesehen. Diese Zeile ist mit C (Zeichenkonstante) bezeichnet.

Ein Kästchen in dieser Zeile entspricht im Hauptspeicher einem _____.

Byte

8 Entnehmen Sie bitte Ihren Arbeitsmitteln ein Programmschema.

Tragen Sie die für Abb. 2 benötigten DC-Anweisungen in das Programmschema ein.

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerku (20 – 25)
K.C.L.I.S.T.E.	D.C. . .	C.'.L.I.S.T.E. N.R. . 3.'
K.C.1.5.Ø. . .	D.C. . .	C.'.1.5.Ø.' . . . . .
K.C.1.Ø.Ø.Ø. .	D.C. . .	C.'.1. . Ø.Ø.Ø.,Ø.Ø.' . .

IBM System / 360
Speicherbelegung

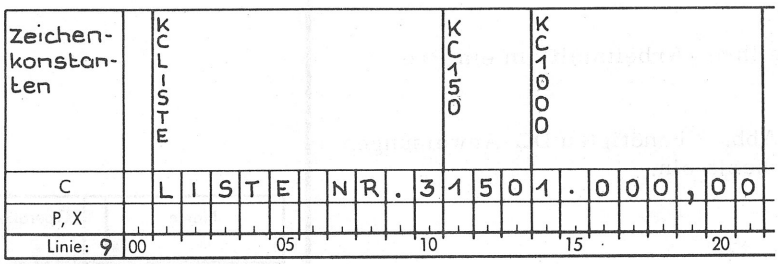


Abb. 2

9 Wie bei den DS-Anweisungen wird auch durch eine DC-Instruktion dem im Namenfeld angegebenen Symbol ein Adress- und ein Längenattribut zugeordnet.

Das Adressattribut ist die Adresse des linken Bytes der Konstanten. Das Längenattribut ergibt sich aus dem Inhalt der Konstanten.

Welche Adress- und Längenattribute werden den folgenden Symbolen zugeordnet (vgl. Abb. 2)?

	Adressattribut	Längenatrribut
KCLISTE	(Z)	(Z)
KC150	(Z)	(Z)
KC1000	(Z)	(Z)

Adressattribut	Längenattribut
901	10
911	3
914	8

Der Längenschlüssel

Bis jetzt ergab sich die Länge der Zeichenkonstanten immer aus der Anzahl der Stellen der Konstanten selbst, also aus ihrem Inhalt.

Es ist jedoch auch möglich, die Länge der Zeichenkonstanten im Operanden der DC-Anweisung mit Hilfe eines L ä n g e n s c h l ü s s e l s anzugeben.

In diesem Fall ergibt sich die Länge der Konstanten immer aus dem Längenschlüssel.

10

Name	Operation	Operand und Be
1 8	10 14	20 25
K.O.N.S.T. . .	D.C. . .	C.L.4.'A.B.C.D.'. .

Der Längenschlüssel L4 bewirkt, daß die Konstante __(Z)__ Bytes im Hauptspeicher belegt.

4

Aus der Konstanten selbst ergäbe sich die Länge von __(Z)__ Bytes.

4

In diesem Fall ist also die Angabe L4 (erforderlich, nicht erforderlich) _____.

nicht erforderlich

11 Ist die Anzahl der Zeichen in der Konstanten größer als die im Längenschlüssel angegebene Länge, so werden die rechten Bytes weggelassen.

K.O.N.S.T. . . | D.C. . . | C.L.3.'A.B.C.D.E.F.'.

Die Angabe der Länge 3 durch L3 in dem Operanden dieser DC-Anweisung bewirkt, daß nur die Buchstaben _____ der angegebenen Konstanten gespeichert werden.

ABC

12 Ist die Anzahl der Zeichen in der Konstanten kleiner als die im Längenschlüssel angegebene Länge, so wird die Konstante rechts durch Blanks erweitert.

K.O.N.S.T. . . | D.C. . . | C.L.8.'A.B.C.D.E.F.'.

Die Angabe der Länge 8 (L8) in dieser DC-Anweisung bewirkt, daß im Hauptspeicher ein __(Z)__ Bytes großes Feld reserviert wird.

8

Die angegebene Konstante A B C D E F wird rechts um __(Z)__ Leerstellen erweitert.

2 (A.B.C.D.E.F.b.b.)

Bedenken Sie bitte, daß die Angabe eines Längenschlüssels nur in einem der beschriebenen Fälle, nämlich der Erweiterung einer Konstanten, von praktischer Bedeutung ist.

Dies sehen wir sogleich am Beispiel einer Blank-Konstanten, die man z. B. zum Leeren eines Druckbereiches verwenden kann.

13

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
K.C.B.L.A.N.K.	D.C.	C' '
oder:		
K.C.B.L.A.N.K.	D.C.	C.L.15.' '

Durch jede der beiden DC-Anweisungen wird eine 15-stellige Blank-Konstante definiert.

Bei der ersten DC-Anweisung erfolgt die Zuordnung des Längenattributs 15 durch ____(WW)____.

Bei der zweiten DC-Anweisung erfolgt die Zuordnung des Längenattributs durch den _____ L15. Die im Operanden angegebene Konstante (' ') wird rechts um ___(ZW)___ erweitert.

Sinngemäß:
den Inhalt der Konstanten
(Anzahl der Leerstellen zwischen den Anführungszeichen)

Längenschlüssel

14 Leerstellen / Blanks

14 Definieren Sie nun bitte auf einem Programmschema Blank-Konstanten (Name:KCBLANK). Benutzen Sie für die Operanden die bequeme Schreibweise mit Längenschlüssel.

KCBLANK soll

a) 30 Bytes,
b) 59 Bytes,
c) 24 Bytes groß sein.

	Name (1–8)	Operation (10–14)	Operand u (20)
a)	K.C.B.L.A.N.K.	D.C.	C.L.30.' '
b)	K.C.B.L.A.N.K.	D.C.	C.L.59.' '
c)	K.C.B.L.A.N.K.	D.C.	C.L.24.' '

Der Duplizierfaktor

15 Wie bei DS-Anweisungen kann auch im Operanden einer DC-Anweisung ein Duplizierfaktor angegeben werden.

Name	Operation	Operan
1 8	10 14	20
K.O.N.S.T.1. .	D.C. . .	4.C.'.N.R.'. .

Der Duplizierfaktor wird als ein dezimaler Wert im Operanden vor dem C, das die Konstantenart bezeichnet, angegeben.

Durch den Operanden 4C'NR' in der obigen DC-Anweisung wird dem Umwandlungsprogramm mitgeteilt, daß die Konstante NR _____ hintereinander gespeichert werden soll.

viermal (N.R.N.R.N.R.N.R.)

16 Wie müssen die Operanden lauten, wenn

a) die Konstante 7 zehnmal,

b) die Konstante DM fünfmal,

c) die Konstante ABCD dreimal hintereinander gespeichert werden soll?

a) 10C '7'

b) 5C ' DM '

c) 3C ' ABCD '

17 Wir haben bereits bei der Behandlung der DS-Anweisung festgestellt, daß die Angabe eines Duplizierfaktors das Adress- und Längenattribut n i c h t b e e i n f l u ß t. - Genauso verhält es sich mit der DC-Anweisung.

Name	Operation	Operan
1 8	10 14	20
K.O.N.S.T.1. .	D.C. . .	4.C.'.N.R.'. .

Durch diese Anweisung wird der symbolischen Adresse KONST1 das Längenattribut (Z) zugeordnet.

2

DAS FORMAT DES OPERANDEN FÜR DAS DEFINIEREN
EINER ZEICHENKONSTANTEN

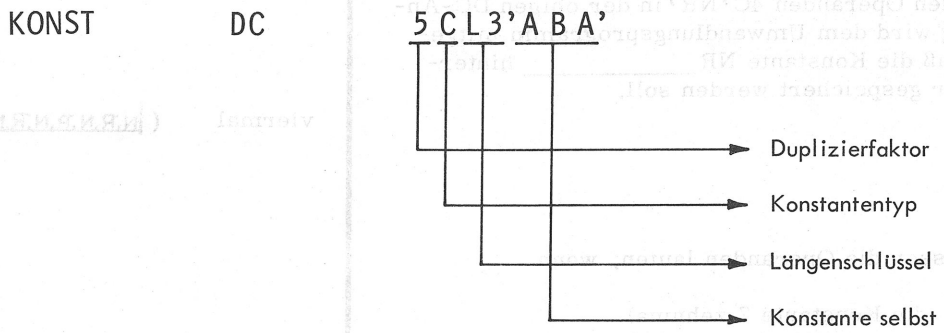


Abb. 3

18 Betrachten Sie bitte Abb. 3. -

Die Abbildung zeigt eine DC-Anweisung für eine Zeichenkonstante. Der Operand dieser DC-Anweisung setzt sich aus verschiedenen Teilen zusammen. Wir besprechen diese verschiedenen Teile nun noch einmal zusammenfassend.

Die erste Stelle des Operanden in Abb. 3 enthält den _____(WZ)_____.

Duplizierfaktor 5

19 Der Duplizierfaktor ist ein dezimaler Direktwert. Ein Duplizierfaktor kann, muß jedoch nicht benutzt werden.

Der Duplizierfaktor 5 in Abb. 3 zeigt an, daß die Konstante ABA im Hauptspeicher _____ hintereinander gespeichert werden soll.

fünfmal

20 Der Konstantentyp (die Art der Konstanten) wird in dem Operanden der Abb. 3 durch den Buchstaben C angegeben.

Dieses C zeigt an, daß es sich bei der zu speichernden Konstanten um eine _____ handelt.

Die Bezeichnung der Konstantenart darf nie fehlen!

Zeichenkonstante

21 In dem Operanden der Abb. 3 ist auch ein Längenschlüssel angegeben.

L3 gibt die Anzahl der Bytes an, die im Hauptspeicher für die definierte Zeichenkonstante benötigt werden.

Könnte der Längenschlüssel L3 im Operanden der Abb. 3 auch fehlen? (Begründen Sie Ihre Antwort.)

Sinngemäß:
Ja, der Längenschlüssel könnte im Falle der Abb. 3 fehlen. L3 ist überflüssig, da durch die Konstante selbst (ABA) die Anzahl der benötigten Bytes im Hauptspeicher angegeben wird.

DAS FORMAT DES OPERANDEN FÜR DAS DEFINIEREN
EINER ZEICHENKONSTANTEN

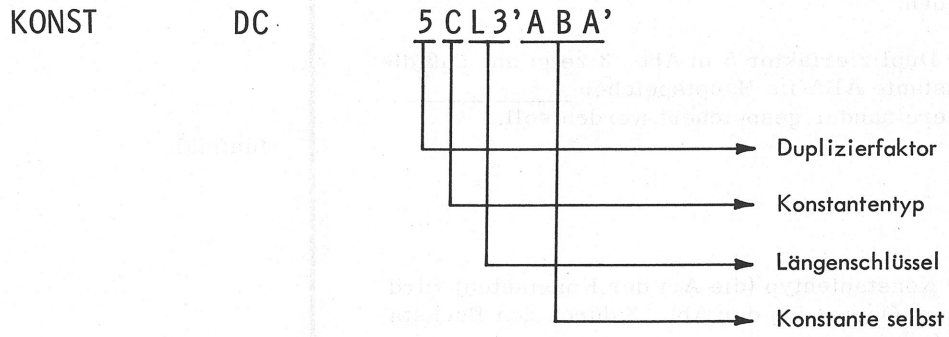


Abb. 3

22 Die Konstante selbst wird in _____ gesetzt.

Das C kennzeichnet die Konstante ABA als eine Zeichenkonstante. Das bedeutet, daß jedes Zeichen zwischen den Hochkommas gemäß dem EBCD-Code in _____(ZW)___ gespeichert wird.

Hochkommas

1 Byte

Abschließend muß zum Thema "Zeichenkonstante" noch eine Besonderheit vermerkt werden. Diese Besonderheit betrifft 2 Zeichen:

a) Apostroph (' innerhalb der Zeichenkonstanten),

b) Ampersand (sprich: ämpersänd) &, das Und-Zeichen.

Werden Apostroph (') oder Ampersand (&) als Zeichen in einer Konstanten, also zwischen den Hochkommas, benötigt, so müssen diese Zeichen d o p p e l t dargestellt werden.

z. B.:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
K.O.N.S.T.1. .	D.C. . .	C.'.W.I.E. .G.E.H.T.'.'.S.?.',
K.O.N.S.T.2. .	D.C. . .	C.'.M.A.I.E.R. .&.& .C.O. .'.

Umgewandelt lauten die hier definierten Konstanten jedoch:

WIE GEHT'S?
MAIER & CO.

Sowohl KONST1 als auch KONST2 erhalten das Längenattribut 11, obwohl die beiden Konstanten im Operanden der DC-Anweisung jeweils 12 Stellen einnehmen.

a) Zeichenkonstanten

| | | |

Übungen

Entnehmen Sie bitte Ihren Arbeitsmitteln ein Programmschema.

a) Definieren Sie die in der Speicherbelegung (Abb. 4) dargestellten Konstanten ZEIKON1 bis ZEIKON3 ohne Benutzung des Längenschlüssels.

b) In den Stellen 1050 bis 1075 der Speicherbelegung in Abb. 4 sind Blank-Konstanten dargestellt. Definieren Sie diese Konstanten bitte mit Hilfe des Längenschlüssels.

c) Mit Hilfe des Duplizierfaktors sind die Konstanten KONC1 und KONC2 (siehe Speicherbelegung in Abb. 4) zu definieren.

Überprüfen Sie nun Ihre Codierung auf der folgenden Seite.

	Name (1 … 8)	Operation (10 … 14)	Operand und Bemerkungen (20 … 40)
a)	Z.E.I.K.O.N.1.	D.C.	C'.L.O.H.N.A.B.R.E.C.H.N.U.N.G. F.E.B.R.. 77'
	Z.E.I.K.O.N.2.	D.C.	C'.1.Ø.Ø.Ø.Ø.'
	Z.E.I.K.O.N.3.	D.C.	C'.A.L.P.H.A. &. B.E.T.A.'
b)	K.C.B.L.1.	D.C.	C.L.6.' '
	K.C.B.L.2.	D.C.	C.L.2Ø.' '
c)	K.O.N.C.1.	D.C.	4.C.'A.B.'
	K.O.N.C.2.	D.C.	5.C.'1.Ø.Ø.'

Band 4, Seite 79 (CLC)

DIE GEPACKTE KONSTANTE - P

Beispiel:

Name	Operation	Operand
1 8	10 14	20
K.O.N.S.T.P. .	D.C. . .	P.'.+.1.5.Ø.'. .

1. Der Typenschlüssel P im Operanden kennzeichnet die in Hochkommas gesetzte Konstante als eine gepackte Konstante.

2. Die Konstante selbst (in Hochkommas) wird als Dezimalzahl mit oder ohne Vorzeichen geschrieben. Ist kein Vorzeichen vorhanden, dann wird die Konstante als positive Zahl behandelt.

3. Die maximale Länge einer gepackten Konstanten darf 16 Bytes betragen.

4. Duplizierfaktor und Längenschlüssel können verwendet werden.

Abb. 1

Wir behandeln jetzt die gepackte Konstante. - Die gepackte Konstante wird auch gepackte Dezimalkonstante genannt, denn mit ihrer Hilfe werden gepackte Dezimalzahlen definiert und im Hauptspeicher gespeichert.

Gepackte Konstanten werden benötigt

a) für das Rechnen mit konstanten Daten,

b) für Vergleiche mit gepackten Daten im Hauptspeicher,

c) für das Übertragen gepackter Daten in ein Feld im Hauptspeicher.

Lesen Sie bitte zunächst den Text in Abb. 1 aufmerksam durch und wenden Sie sich dann den folgenden Lernelementen zu. -

1 Eine gepackte Konstante wird im Operanden der DC-Anweisung durch den Typenschlüssel _____ gekennzeichnet.

Der Buchstabe P im Operanden einer DC-Anweisung bezeichnet die in Hochkommas gesetzte Konstante als eine _____ Dezimalzahl.

P

gepackte

2 Die Konstante selbst wird im Operanden der DC-Anweisung in _____ gesetzt.

Sie wird als Dezimalzahl mit oder ohne _____ geschrieben.

Ist kein Vorzeichen vorhanden, dann wird die in Hochkommas gesetzte Konstante als _____ Zahl behandelt.

Hochkommas

Vorzeichen

positive

DIE GEPACKTE KONSTANTE - P

Beispiel:

Name	Operation	Operand
1 8	10 14	20
K,O,N,S,T,P, ,	D,C, , ,	P,',+,1,5,Ø,', ,

1. Der Typenschlüssel P im Operanden kennzeichnet die in Hochkommas gesetzte Konstante als eine gepackte Konstante.

2. Die Konstante selbst (in Hochkommas) wird als Dezimalzahl mit oder ohne Vorzeichen geschrieben. Ist kein Vorzeichen vorhanden, dann wird die Konstante als positive Zahl behandelt.

3. Die maximale Länge einer gepackten Konstanten darf 16 Bytes betragen.

4. Duplizierfaktor und Längenschlüssel können verwendet werden.

Abb. 1

3 Wird ein Vorzeichen beim Definieren einer gepackten Konstanten angegeben, so muß dieses Vorzeichen v o r der Dezimalzahl in den Hochkommas geschrieben werden.

Wie lauten die Operanden für die folgenden gepackten Konstanten?

a) 1000

b) +550

c) -5

a) P'1000' oder P'+1000'

b) P'+550' oder P'550'

c) P'-5'

4 Wie groß (Anzahl in Bytes) darf eine gepackte Konstante maximal sein?

16 Bytes

Name (1–8)	Operation (10–14)	Operand und Bem (20–25)
K,O,N,S,T,P,1,	D,C, . .	P,',1,Ø,Ø,Ø,Ø,', . . .
K,O,N,S,T,P,2,	D,C, . .	P,',+,1,8,Ø,Ø,Ø,', . . .
K,O,N,S,T,P,3,	D,C, . .	P,',-,5,Ø,Ø,', ...

IBM System / 360
Speicherbelegung

Gepackte Konstanten		KONSTP1					KONSTP2				KONSTP3		
C													
P, X		01	00	00	0+		18	00	0+		50	0-	
Linie:	00					05							

Abb. 2

5 Betrachten Sie bitte Abb. 2. -

Wie Sie sehen, ist auch im Speicherbelegungsblatt Platz für die Eintragung gepackter Konstanten vorgesehen. Gepackte Konstanten werden auf dem Speicherbelegungsblatt in die Kästchenreihe eingetragen, die mit P, X gekennzeichnet ist. (Über die Bedeutung des X wird noch an anderer Stelle gesprochen.)

Wie Sie aus der Speicherbelegung in Abb. 2 entnehmen können, werden bei der gepackten Konstanten jeweils __(Z)__ Ziffern in __(Z)__ Byte gespeichert.

2

1

6 Das rechte Byte enthält jeweils im (rechten, linken) _____ Halbbyte das Vorzeichen.

rechten

(ein positives Vorzeichen wird als hexadezimales C und ein negatives Vorzeichen wird als hexadezimales D gespeichert)

7 Bei einer geraden Anzahl von Ziffern (z. B. KONSTP1 = 100 000) wird das linke Halbbyte des linken Bytes mit einer Null aufgefüllt.

Entnehmen Sie bitte Ihren Arbeitsmitteln ein Speicherbelegungsblatt.

Tragen Sie ab der Stelle 01 beginnend die folgenden gepackten Konstanten ein:

a) KP1 = 1255

b) KP2 = -500377

c) KP3 = 10

		KP1 a)				KP2 b)						KP3 c)		
C														
P, X		0	1	2	5	5+	0	5	0	0	3	7	7-	0	1	0+
Linie:	00							05								

8 Definieren Sie nun bitte auf einem Programmschema die gepackten Konstanten KP1, KP2 und KP3, die Sie soeben in das Speicherbelegungsblatt eingetragen haben.

Geben Sie bitte das Vorzeichen nur an, wo es unbedingt notwendig ist!

	Name (1–8)	Operation (10–14)	Operand un (20)
a)	K.P.1.	D.C. . .	P.'1255'. . .
b)	K.P.2.	D.C. . .	P.'-5ØØ377
c)	K.P.3.	D.C. . .	P.'1Ø'. . . .

Führende Nullen, die in der Speicherbelegung auftauchen, brauchen in der Codierung natürlich nicht geschrieben werden.

9

Name (1–8)	Operation (10–14)	Operanc (20)
K.P.1.	D.C. . .	P.'1255'.

Wird kein Längenschlüssel angegeben, dann ergibt sich die Länge einer gepackten Konstanten aus der Anzahl der Bytes, welche die Konstante im Hauptspeicher belegt.

Die obige Konstante belegt im Hauptspeicher ___(Z)___ Bytes.

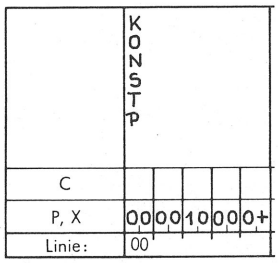
3 (Vorzeichen und führende Null der Konstanten sind mitzurechnen!)

10 Wird ein Längenschlüssel angegeben und benötigt die in Hochkommas gesetzte Konstante weniger Bytes, so wird das Feld im Hauptspeicher links mit Nullen aufgefüllt.

```
KONSTP.. | DC... | PL5'1ØØØØ'.
```

Durch L5 im Operanden dieser DC-Anweisung wird ein 5 Bytes großes Feld im Hauptspeicher reserviert.

Stellen Sie bitte auf einem Speicherbelegungsblatt dar, wie die gepackte Konstante 10 000 in dem 5 Bytes großen Feld im Hauptspeicher untergebracht ist.

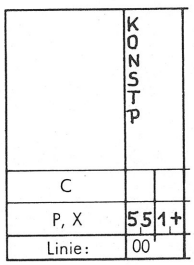


11 Benötigt die Konstante mehr Bytes als der Längenschlüssel angibt, dann fallen entsprechend der Längenangabe die linken Stellen der Konstanten weg.

```
KONSTP.. | DC... | PL2'55551'.
```

Durch diese DC-Anweisung werden für die Konstante KONSTP nur 2 Bytes im Hauptspeicher reserviert.

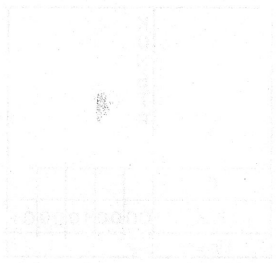
Tragen Sie bitte auf einem Speicherbelegungsblatt ein, was von der oben angegebenen gepackten Konstanten noch gespeichert wird.



12 Auch der Duplizierfaktor darf bei gepackten Konstanten verwendet werden. Der Duplizierfaktor muß im Operanden vor dem Typenschlüssel P angegeben werden.

Wie lautet der Operand, wenn die gepackte Konstante -100 fünfmal hintereinander gespeichert werden soll?

5P'-100'



Übungen

1. Entnehmen Sie bitte Ihren Arbeitsmitteln ein Speicherbelegungsblatt.

   Tragen Sie ab der Stelle 01 beginnend die folgenden gepackten Konstanten ein:

   a) KPA = 25000

   b) KPB = -15

   c) KPC = 1550

   d) KPD = -120000

   e) KPE = 0000000

2. Definieren Sie nun bitte auf einem Programmschema die gepackten Konstanten, die Sie in das Speicherbelegungsblatt eingetragen haben.

Überprüfen Sie Ihre Ergebnisse auf der folgenden Seite !

1.

		KPA a)	KPB b)	KPC c)	KPD d)	KPE e)
C						
P, X		2500 0+	01 5-	01 55 0+	01 2000 0-	0000000 0+
Linie:	00		05		10	15

2.

	Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkung (20 – 25)
a)	K.P.A.	D.C.	P,'25ØØØ.'
b)	K.P.B.	D.C.	P,'-15.'
c)	K.P.C.	D.C.	P,'155Ø.'
d)	K.P.D.	D.C.	P,'-12ØØØØ.'
e)	K.P.E.	D.C.	P,L4,'Ø.'
	oder:		
a)	K.P.A.	D.C.	P,'+25ØØØ.'
c)	K.P.C.	D.C.	P,'+155Ø.'
e)	K.P.E.	D.C.	P,'ØØØØØØØ.'

Band 4, Seite 320 (CP)

DIE HEXADEZIMALKONSTANTE - X

Beispiel:

Name	Operation	Operand
1 8	10 14	20
K.O.N.S.T.X. .	D.C. . .	X.'.B.6.F.4.'.

1. Der Typenschlüssel X im Operanden kennzeichnet die in Hochkommas gesetzte Konstante als eine Hexadezimalkonstante.

2. Die Konstante selbst wird als hexadezimal angegeben. Es werden die Hexadezimalziffern 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F verwendet. Je 2 Hexadezimalziffern werden in einem Byte gespeichert.

3. Die Hexadezimalkonstante darf maximal 256 Bytes groß sein.

4. Duplizierfaktor und Längenschlüssel dürfen verwendet werden.

Tabelle der Hexadezimalziffern und ihrer Verschlüsselung im Halbbyte:

Hexadezimalziffer	Verschlüsselung in 1 Halbbyte
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
D	1101
E	1110
F	1111

Abb. 1

Sie haben bisher die Zeichenkonstante und die gepackte Konstante kennengelernt.

Durch Zeichenkonstanten wird 1 Zeichen (Buchstabe, Ziffer oder Sonderzeichen) in einem Byte gespeichert.

Durch gepackte Konstanten werden 2 Dezimalziffern in einem Byte gespeichert. Im rechten Halbbyte des rechten Bytes befindet sich das Vorzeichen.

Wird ein Längenschlüssel angegeben und ist die in Hochkommas gesetzte Konstante kleiner als im Längenschlüssel angegeben, dann gilt:

a) für die Zeichenkonstante:
die Konstante wird rechts um Blanks erweitert,

b) für die gepackte Konstante:
die Konstante wird links um führende Nullen erweitert.

Wird ein Längenschlüssel angegeben und ist die in Hochkommas gesetzte Konstante größer als im Längenschlüssel angegeben, dann gilt:

a) für die Zeichenkonstante:
rechts fallen Stellen weg,

b) für die gepackte Konstante:
links fallen Stellen weg.

Jetzt behandeln wir die Hexadezimalkonstante. Durch diese Konstante kann man bequem jegliche Bitkombination im Hauptspeicher darstellen.

Alles was wir über den Längenschlüssel bei der gepackten Konstanten gesagt haben, gilt auch für die Hexadezimalkonstante. Wir werden hier deshalb nicht mehr im einzelnen auf den Längenschlüssel eingehen.

Lesen Sie nun bitte den Text in Abb. 1 aufmerksam durch und wenden Sie sich dann den folgenden Lernelementen zu. -

1 Eine Hexadezimalkonstante wird im Operanden durch den Typenschlüssel _____ gekennzeichnet.

Der Buchstabe X im Operanden einer DC-Anweisung teilt dem Assembler mit, daß es sich bei der in Hochkommas gesetzten Konstanten um eine _____ handelt.

X

Hexadezimalkonstante

DIE HEXADEZIMALKONSTANTE - X

Beispiel:

Name	Operation	Operand
1 8	10 14	20
K,O,N,S,T,X, ,	D,C, , ,	X,',F,6,F,4,',

1. Der Typenschlüssel X im Operanden kennzeichnet die in Hochkommas gesetzte Konstante als eine Hexadezimalkonstante.

2. Die Konstante selbst wird als hexadezimal angegeben. Es werden die Hexadezimalziffern 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F verwendet. Je 2 Hexadezimalziffern werden in einem Byte gespeichert.

3. Die Hexadezimalkonstante darf maximal 256 Bytes groß sein.

4. Duplizierfaktor und Längenschlüssel dürfen verwendet werden.

Tabelle der Hexadezimalziffern und ihrer Verschlüsselung im Halbbyte:

Hexadezimalziffer	Verschlüsselung in 1 Halbbyte
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
D	1101
E	1110
F	1111

Abb. 1

2 Die Konstante selbst wird im Operanden der DC-Anweisung in Hochkommas gesetzt.

Sie wird durch die Hexadezimalziffern __(Z)__ bis __(Z)__ angegeben.

Eine Hexadezimalkonstante darf maximal __(Z)__ Bytes groß sein.

0 (bis) F

256

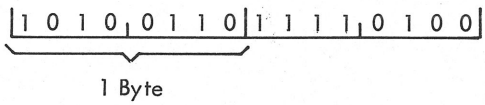
3 Jede Hexadezimalziffer der definierten Konstanten wird in einem Halbbyte binär verschlüsselt.

KONSTX in Abb. 1 belegt im Hauptspeicher also __(Z)__ Bytes.

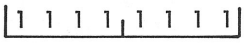
2 (für jede Hexadezimalziffer 1 Halbbyte)

4 Die in der Hexadezimalkonstanten KONSTX (Abb. 1) angegebenen Hexadezimalziffern werden im Hauptspeicher binär verschlüsselt gespeichert.

Wie steht die Hexadezimalkonstante A 6 F 4 im Hauptspeicher ? (Siehe Tabelle in Abb. 1)



5 Wie muß der Operand einer DC-Anweisung lauten, wenn durch eine Hexadezimalkonstante im Hauptspeicher ein 1 Byte großes Feld mit binären Einsen erzeugt werden soll:



X' FF'

IBM System / 360
Speicherbelegung

Hexadezimalkonstanten		KONSTX1						KONSTX2								KONSTX3		
C																		
P, X	'	0	8	R	0	4	R	3	D	D	4	4	7	0	0	F	F	R	5	F	6
Linie:	00										05										10

Abb. 2

6 Betrachten Sie bitte Abb. 2. -

Im Speicherbelegungsblatt ist Platz für die Eintragung von Hexadezimalkonstanten vorgesehen. Die Hexadezimalkonstanten (X) werden in dieselbe Kästchenreihe wie die gepackten Konstanten (P) eingetragen.

Es werden also jeweils __(Z)__ Hexadezimalziffern in __(Z)__ Byte gespeichert werden.

2

1

7 Bei einer ungeraden Anzahl von Hexadezimalziffern (z. B. '8A04A') wird das linke Halbbyte mit einer hexadezimalen Null aufgefüllt.

In der Speicherbelegung in Abb. 2 ist bei der Hexadezimalkonstanten mit dem Namen _____ das am weitesten links stehende Halbbyte auf Null gesetzt.

KONSTX1

8 Definieren Sie nun bitte auf einem Programmschema die in der Speicherbelegung in Abb. 2 dargestellten Hexadezimalkonstanten.

Name	Operation	Operand und Bemerku
1 8	10 14	20 25
K,O,N,S,T,X,1,	D,C, , ,	X,',8,A,Ø,4,A,', , , , ,
K,O,N,S,T,X,2,	D,C, , ,	X,',3,D,D,4,4,7,Ø,Ø,F,F,',
K,O,N,S,T,X,3,	D,C, , ,	X,',A,5,F,6,', , , , , ,

	Name 1 8	Operation 10 14	Oper 20
a)	K.1.	D.C. . .	C.'1.5.' . .
	. . . . .	. . .	. . . . .
b)	K.2.	D.C. . .	P.'1.5.' . .
	. . . . .	. . .	. . . . .
c)	K.3.	D.C. . .	X.'1.5.' . .

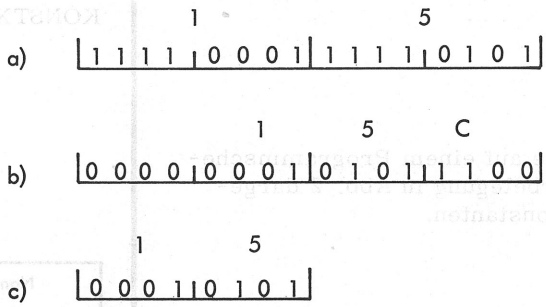


Abb. 3

Im folgenden stellen wir noch einmal die Konstantentypen C, P, X einander gegenüber. Schauen Sie sich bitte hierzu Abb. 3 an. -

9 Abb. 3 macht deutlich, welche entscheidende Rolle der Typenschlüssel im Operanden einer DC-Anweisung spielt. Obwohl für jede der Konstanten im Operanden die Konstantendaten '15' angegeben sind, bewirkt der Typenschlüssel eine jeweils verschiedene Verschlüsselung der Daten im Hauptspeicher.

Der Inhalt der Zeichenkonstanten K1 wird im Hauptspeicher in __(Z)__ Bytes gespeichert.

Jede Ziffer sowie das Vorzeichen wird bei der gepackten Konstanten K2 in einem _____ gespeichert.

Jede Hexadezimalziffer der Hexadezimalkonstanten mit dem Namen K3 wird in einem _____ gespeichert.

2

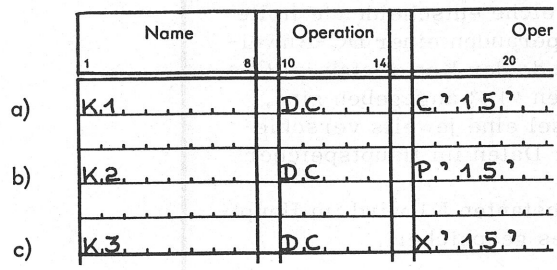
Halbbyte

Halbbyte

10 Statt der gepackten Konstanten(b) in Abb. 3 könnte auch eine Hexadezimalkonstante definiert werden, die den gleichen Speicherinhalt erzeugt.

Wie müßte der Operand einer solchen Hexadezimalkonstanten lauten?

X'15C'


Name
Operation
Oper
1
8
10
14
20
a)
K.1
D.C.
C,'1,5,'
b)
K.2
D.C.
P,'1,5,'
c)
K.3
D.C.
X,'1,5,'

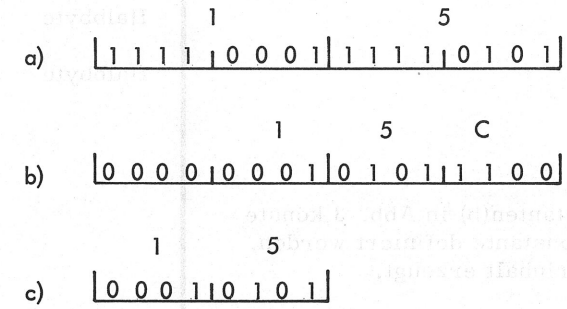

1
5
a)
1 1 1 1 0 0 0 1 1 1 1 1 0 1 0 1
1
5
C
b)
0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 0
1
5
c)
0 0 0 1 0 1 0 1

Abb. 3

11 Im Hauptspeicher soll ein 3 Bytes großes Rechenfeld reserviert werden, das folgenden Inhalt hat:

K
R
E
F
E

0	0	0	0	0	C

Tragen Sie bitte die möglichen DC-Anweisungen in ein Codierschema ein.

Name	Operation	Operand und
1 8	10 14	20 2
K.R.E.F.E. . .	D.C. . .	P.'Ø.Ø.Ø.Ø.Ø.' . .
K.R.E.F.E. . .	D.C. . .	X.'Ø.Ø.Ø.Ø.Ø.C.' .
oder : . . .	. . . .	. . . . .
K.R.E.F.E. . .	D.C. . .	P.L.3.'Ø.' . . . .
K.R.E.F.E. . .	D.C. . .	X.L.3.'C.' . . . .
. . . . .	. . . . .	. . . . .

Übungen

1. Entnehmen Sie bitte Ihren Arbeitsmitteln ein Speicherbelegungsblatt und tragen Sie ab der Stelle 01 beginnend die folgenden Hexadezimalkonstanten ein:

   a) KONX1 = 4B57CF

   b) KONX2 = FFF

   c) KONX3 = A9B1C55

   d) KONX4 = D

2. Definieren Sie nun bitte auf einem Programmschema die Hexadezimalkonstanten, die Sie in das Speicherbelegungsblatt eingetragen haben.

Überprüfen Sie Ihre Ergebnisse auf der nächsten Seite!

1.

		KONX1 a)	KONX2 b)	KONX3 c)	KONX4 d)
C					
P, X		4B57CF	0FFF	0R9B1C55	0D
Linie:	00		05		10

2.

	Name 1 8	Operation 10 14	Operand und E 20 25
a)	K,O,N,X,1, . .	D,C, . .	X,',4,B,5,7,C,F,',
b)	K,O,N,X,2, . .	D,C, . .	X,',F,F,F,', . . .
c)	K,O,N,X,3, . .	D,C, . .	X,',R,9,B,1,C,5,5,',
d)	K,O,N,X,4, . .	D,C, . .	X,',D,', ...

Entnehmen Sie nun bitte Ihren Arbeitsmitteln Merkblatt 304, Seite 1.

Band 1, Seite 171

DIE WORTKONSTANTE - F

Mit Hilfe einer Wortkonstanten können konstante Dualdaten in Wortgröße im Hauptspeicher erzeugt werden.

Beispiel:

Name	Operation	Operand
1 8	10 14	20
K.O.N.S.T.F. .	D.C. . .	F.'6,4.' . .

1. Der Typenschlüssel F kennzeichnet die in Hochkommas gesetzte Konstante für den Assembler als Wortkonstante. (F ist die Abkürzung von englisch full-word=Wort.)

2. Die Konstante selbst wird als Dezimalzahl geschrieben. Bei der Umwandlung wird diese Dezimalzahl in eine Dualzahl umgewandelt und in einem Wort (4 Bytes) gespeichert.

3. Die Dezimalzahl innerhalb der Hochkommas kann mit einem Vorzeichen versehen werden (z.B. '-194'). Wird kein Vorzeichen benutzt, so gilt die angegebene Konstante als positiv.

4. Die Wortkonstante unterliegt der Begrenzungszuordnung.

5. Duplizierfaktor und Längenschlüssel dürfen verwendet werden. Bei Verwendung eines Längenschlüssels (z.B. FL3'13') wird die Begrenzungszuordnung nicht berücksichtigt.

Abb. 1

Sie haben bereits Dualdaten und ihre Formate kennengelernt. Wir behandeln nun Dualdaten, die als Konstanten in der Ihnen bekannten Form (Halbwort, Wort) gespeichert werden.

Erinnern Sie sich bitte hierzu, wie die Formate für Dualdaten aussehen und wo das Vorzeichen in diesen Formaten steht. -

Lesen Sie bitte zunächst den Text in Abb. 1 aufmerksam durch und wenden Sie sich dann den folgenden Lernelementen zu.

1 Eine Wortkonstante wird im Operanden durch den Typenschlüssel _____ gekennzeichnet.

Der Buchstabe F im Operanden einer DC-Anweisung bezeichnet die in Hochkommas gesetzte Konstante als eine _____.

F

Wortkonstante

2 Die in Hochkommas gesetzte Konstante wird als _____ mit oder ohne Vorzeichen geschrieben.

Wird kein Vorzeichen benutzt, so gilt die Konstante als _____.

Dezimalzahl

positiv

3 Bei der Umwandlung wird die in Hochkommas gesetzte Dezimalzahl in eine _____ umgewandelt.

Dualzahl

4 Die binäre Zahl wird in einem _____ im Hauptspeicher gespeichert.

Wo steht das Vorzeichen in einem Wort und wie wird es dargestellt?

Wort

Das Vorzeichen steht im linken Bit eines Wortes und wird durch 0 (positiv) und 1 (negativ) dargestellt.

DIE WORTKONSTANTE - F

Mit Hilfe einer Wortkonstanten können konstante Dualdaten in Wortgröße im Hauptspeicher erzeugt werden.

Beispiel:

Name	Operation	Operand
1 8	10 14	20
K.O.N.S.T.F. .	D.C. . .	F.'1.6.4.'. . .

1. Der Typenschlüssel F kennzeichnet die in Hochkommas gesetzte Konstante für den Assembler als Wortkonstante. (F ist die Abkürzung von englisch full-word=Wort.)

2. Die Konstante selbst wird als Dezimalzahl geschrieben. Bei der Umwandlung wird diese Dezimalzahl in eine Dualzahl umgewandelt und in einem Wort (4 Bytes) gespeichert.

3. Die Dezimalzahl innerhalb der Hochkommas kann mit einem Vorzeichen versehen werden (z.B. '-194'). Wird kein Vorzeichen benutzt, so gilt die angegebene Konstante als positiv.

4. Die Wortkonstante unterliegt der Begrenzungszuordnung.

5. Duplizierfaktor und Längenschlüssel dürfen verwendet werden. Bei Verwendung eines Längenschlüssels (z.B. FL3'13') wird die Begrenzungszuordnung nicht berücksichtigt.

Abb. 1

5 Zeichnen Sie bitte auf ein Extrablatt, wie die in Abb. 1 definierte Wortkonstante im Hauptspeicher nach der Umwandlung binär dargestellt ist. (Die Dezimalzahl 64 lautet binär 1000000.)

(+)
|00000000000000000000000001000000|

32 Bits

(Die Dualzahl wird rechtsbündig im Wort gespeichert. Die links nicht benötigten Bits werden bis zum Vorzeichenbit mit Nullen aufgefüllt.)

6 Die folgenden Dezimalzahlen sollen als Konstanten binär in einem Wort im Hauptspeicher gespeichert werden. (Definieren Sie die Wortkonstanten auf einem Programmschema!)

a) KONF1 = 37

b) KONF2 = -194

	Name (1–8)	Operation (10–14)	Operand (20)
a)	K.O.N.F.1. . .	D.C. . .	F.'37.' . .
	oder :		
	K.O.N.F.1. . .	D.C. . .	F.'+37.' .
b)	K.O.N.F.2. . .	D.C. . .	F.'-194.'

7 Wie bei allen Konstantentypen, die wir bisher kennengelernt haben, kann auch bei der Definition einer Wortkonstanten im Operanden ein Duplizierfaktor angegeben werden.

Name (1–8)	Operation (10–14)	Operand (20)
K.F.	D.C. . .	3.F.'550.'

Durch diese DC-Anweisung werden im Hauptspeicher (Z) Felder in Wortgröße reserviert.

In jedem dieser Felder ist die Dezimalzahl 550 _____ dargestellt.

3

binär / dual

Ein Längenschlüssel kann auch bei der Wortkonstanten angegeben werden.

Ist kein Längenschlüssel angegeben - und das wird bei der Benutzung einer Wortkonstanten fast immer der Fall sein-, dann erhält die Wortkonstante eine feste Länge zugeordnet und wird an einer Wortadresse gespeichert (Begrenzungszuordnung).

Übungen

Entnehmen Sie bitte Ihren Arbeitsmitteln ein Programmschema und definieren Sie die folgenden Wortkonstanten:

a) KONF1 = 4399

b) KONF2 = -5000

c) KONF3 = 1555 (1555 soll fünfmal hintereinander in je einem Wort gespeichert werden).

	Name (1–8)	Operation (10–14)	Operand (20)
a)	K.O.N.F.1. . .	D.C. . .	F.'4.3.9.9.'.
b)	K.O.N.F.2. . .	D.C. . .	F.'-.5.Ø.Ø.Ø.'
c)	K.O.N.F.3. . .	D.C. . .	5F.'1.5.5.5.'
	. . . . .	. . . . .	. . . . .
	oder :		
a)	K.O.N.F.1. . .	D.C. . .	F.'+.4.3.9.9.'
c)	K.O.N.F.3. . .	D.C. . .	5F.'+.1.5.5.5

DIE HALBWORTKONSTANTE - H

Mit Hilfe einer Halbwortkonstanten können konstante Dualdaten in Halbwortgröße im Hauptspeicher erzeugt werden.

Beispiel:

Name	Operation	Operand
1 8	10 14	20
K,O,N,S,T,H, ,	D,C, , ,	H,',6,1,', , ,

1. Der Typenschlüssel H kennzeichnet die in Hochkommas gesetzte Konstante für den Assembler als Halbwortkonstante. (H ist die Abkürzung von englisch half-word = Halbwort.)

2. Die Konstante selbst wird als Dezimalzahl geschrieben. Bei der Umwandlung wird diese Dezimalzahl in eine Dualzahl umgewandelt und in einem Halbwort (2 Bytes) gespeichert.

3. Die Dezimalzahl innerhalb der Hochkommas kann mit einem Vorzeichen versehen werden (z.B. '-1000'). Wird kein Vorzeichen benutzt, so gilt die angegebene Konstante als positiv.

4. Die Halbwortkonstante unterliegt der Begrenzungszuordnung.

5. Duplizierfaktor und Längenschlüssel dürfen verwendet werden. Bei Verwendung eines Längenschlüssels (z.B. HL1'15') wird die Begrenzungszuordnung nicht berücksichtigt.

Abb. 1

1 Lesen Sie bitte zunächst den Text in Abb. 1 aufmerksam durch. -

Eine Halbwortkonstante wird im Operanden der DC-Anweisung durch den Typenschlüssel _____ gekennzeichnet.

Der Buchstabe H im Operanden der DC-Anweisung teilt dem Umwandlungsprogramm mit, daß es sich bei der in Hochkommas gesetzten Konstanten um eine _____ handelt.

H

Halbwortkonstante

2 Wie die Wortkonstante wird auch die Halbwortkonstante als Dezimalzahl mit oder ohne Vorzeichen im Operanden einer DC-Anweisung in Hochkommas gesetzt.

Bei der Umwandlung wird diese Dezimalzahl in eine _____ umgewandelt und zusammen mit dem Vorzeichen in einem _____ gespeichert.

Dualzahl

Halbwort

3 Zeichnen Sie bitte auf ein Extrablatt, wie die in Abb. 1 definierte Halbwortkonstante im Hauptspeicher nach der Umwandlung binär dargestellt ist. (Die Dezimalzahl 61 lautet binär 111101.)

(+)
0000000000111101

16 Bits

Sie sehen, die Halbwortkonstante entspricht im Grunde der Wortkonstanten, nur lautet der Typenschlüssel H statt F und die Länge ist ein Halbwort statt ein Wort.

In der Regel wird auch bei der Halbwortkonstanten kein Längenschlüssel angegeben, so daß die Halbwortkonstante eine feste Länge zugeordnet erhält und ab einer Halbwortadresse gespeichert wird (Begrenzungszuordnung).

Übungen

Entnehmen Sie bitte Ihren Arbeitsmitteln ein Programmschema und definieren Sie die folgenden Halbwortkonstanten:

a) KONH1 = 32767

b) KONH2 = -32768

c) KONH3 = 100 (100 soll zweimal hintereinander in je einem Halbwort gespeichert werden).

	Name (1 … 8)	Operation (10 … 14)	Operan (20 …)
a)	K.O.N.H.1. . .	D.C. . .	H.'32767'
b)	K.O.N.H.2. . .	D.C. . .	H.'-32768'
c)	K.O.N.H.3. . .	D.C. . .	2H.'100'.
	. . . . .	. . . . .	. . . . .
	oder:		
a)	K.O.N.H.1. . .	D.C. . .	H.'+32767'
c)	K.O.N.H.3. . .	D.C. . .	2H.'+100'.

DIE ADRESSKONSTANTE - A

Der Inhalt einer Adresskonstanten ist eine Hauptspeicheradresse (effektive Adresse), die dual in einem Wort gespeichert wird. Adresskonstanten werden z.B. dazu verwendet, einen Adresswert in ein Basisregister zu laden.

Beispiel:

Name	Operation	Operand u
1 8	10 14	20
A,K,O,N,S,T, ,	D,C, , ,	A,(,F,E,L,D,1,),

1. Der Typenschlüssel A im Operanden einer DC-Anweisung kennzeichnet die definierte Konstante als Adresskonstante.

2. Die zu speichernde Adresse wird im Operanden symbolisch - in Klammern gesetzt - angegeben.

3. Der in Klammern gesetzte Ausdruck kann sein:

   a) ein Symbol, z.B. (FELD1),
   b) ein zusammengesetzter Ausdruck, z.B. (FELD1+5) oder (*-125).

4. Ein Symbol, das in dem in Klammern gesetzten Ausdruck verwendet wird, muß der Name einer Instruktion oder DS-Anweisung oder der * sein.

5. Die Adresskonstante unterliegt der Begrenzungszuordnung.

6. Duplizierfaktor und Längenschlüssel sind erlaubt.

Abb. 1

1 Lesen Sie bitte zunächst den Text der Abb. 1 aufmerksam durch. -

Der Typenschlüssel _____ kennzeichnet die in Abb. 1 definierte Konstante als eine _____.

A

Adreßkonstante

2 Bei dem Definieren einer Adreßkonstanten ist zu beachten, daß die Konstante selbst als symbolischer Ausdruck im Operanden der DC-Anweisung in _____ gesetzt wird.

Klammern

3 Im Operanden steht innerhalb der Klammern ein einfacher oder zusammengesetzter Ausdruck wie z. B. FELD1 oder FELD1+5.

Es kann auch ein zusammengesetzter Ausdruck verwendet werden, der von dem augenblicklichen Wert des _____ ausgeht, z. B. *-125.

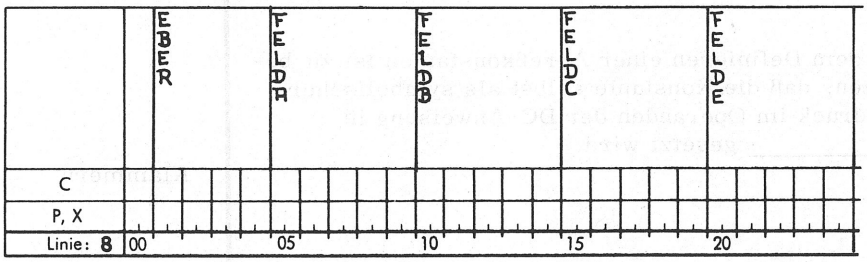
Zuordnungszählers

4 Der Ausdruck in den Klammern ist die symbolische Bezeichnung für eine Hauptspeicheradresse.

Wenn z. B. der augenblickliche Wert des Zuordnungszählers 1 000 ist, dann stellt der Ausdruck *-50 den Adresswert __(Z)__ dar.

950

IBM System / 360
Speicherbelegung



	Name (1–8)	Operation (10–14)	Operand und Bemerku (20–25)
	EBER	DS	ØCL24
		DS	CL4
	FELDA	DS	CL5
	FELDB	DS	CL5
	FELDC	DS	CL5
	FELDE	DS	CL5
a)	BKON1	DC	A(FELDA)
b)	BKON2	DC	A(FELDA-2)
c)	BKON3	DC	A(FELDB+15)

Abb. 2

5 Betrachten Sie bitte Abb. 2.

FELDA im Operanden der DC-Anweisung (a) stellt die Hauptspeicheradresse 805 dar (siehe Speicherbelegung in Abb. 2).

Welche Adresswerte werden durch die Ausdrücke FELDA-2 und FELDB+15 in den Operanden der DC-Anweisungen b) und c) dargestellt ?

b) 803

c) 825

6 Der Adresswert des in Klammern gesetzten Ausdruckes ist der Inhalt der Adresskonstanten. In Adresskonstanten werden also Hauptspeicheradressen als konstante Daten definiert.

Name	Operation	Operand und
1 8	10 14	20 2
ADRKON	DC	A(FELDC)

Welche Hauptspeicheradresse wird zum Inhalt der hier definierten Adresskonstanten ? (Siehe Speicherbelegung in Abb. 2.)

815

7 Die Adresskonstante wird bei der Umwandlung in einem Wort binär gespeichert.

Name	Operation	Operand und
1 8	10 14	20
ADRKON	DC	A(FELDC)

00000000000000000000001100101111

1 Wort = 32 Bits

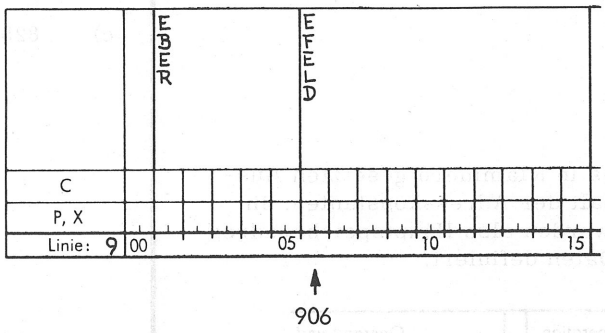
Durch die obige DC-Anweisung wird die Hauptspeicheradresse __(Z)__ in die Dualzahl __(Z)__ umgewandelt und in einem _____ gespeichert.

815

1100101111

Wort

IBM System / 360
Speicherbelegung



Name	Operation	Operan
1 8	10 14	20
E,B,E,R, . . .	D,S, . .	Ø,C,L,1,5, . .
. . . . .	D,S, . .	C,L,5, ...
E,F,E,L,D, . .	D,S, . .	C,L,1,Ø, . . .

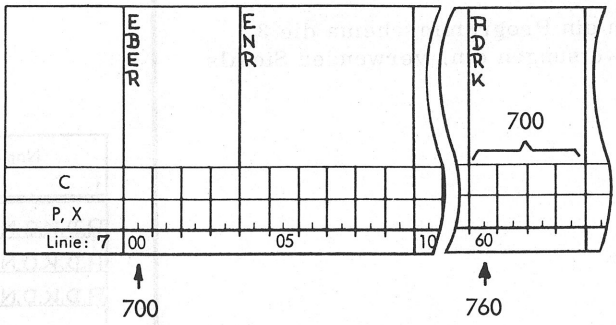
Abb. 3

8 In Abb. 3 sind Speicherbelegung und Definitionen für ein Feld im Eingabebereich dargestellt.

Die Hauptspeicheradresse 906 (EFELD) soll als Adresskonstante definiert werden. Der augenblickliche Wert des Zuordnungszählers sei 946.

Tragen Sie bitte in ein Programmschema die 3 möglichen DC-Anweisungen ein, verwenden Sie als Name ADKON.

Name	Operation	Operand und
1 8	10 14	20
A.D.K.O.N. . .	D.C. . .	A.(.E.F.E.L.D.) . .
A.D.K.O.N. . .	D.C. . .	A.(.E.B.E.R.+.5.) .
A.D.K.O.N. . .	D.C. . .	A.(.*.-.4.0.) . . .



Name	Operation	Operand
1 8	10 14	20
EBER	DS	Ø,CL15
	DS	CL4
ENR	DS	CL6
ADRK	DC	A(EBER)

Abb. 4

9 Betrachten Sie bitte Abb. 4. -

Es ist zu unterscheiden zwischen der A d r e s s e einer Adresskonstanten im Hauptspeicher und dem I n h a l t einer Adresskonstanten.

Die in Abb. 4 gezeigte Adresskonstante mit dem symbolischen Namen _____ erhält die Adresse __(Z)___ .

Der Inhalt dieser Adresskonstanten ist der Adresswert __(Z)___. Das ist der Adresswert, den das Symbol _____ im Operanden der DC-Anweisung darstellt.

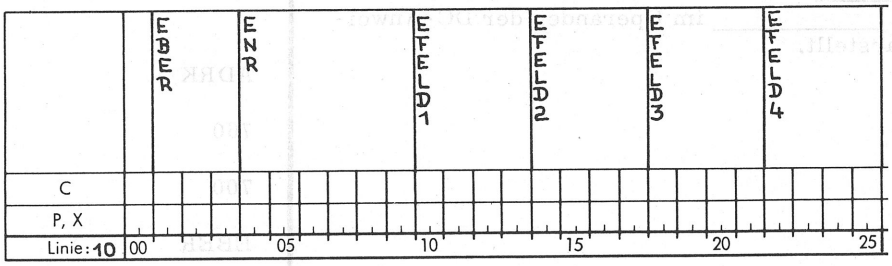
ADRK

760

700

EBER

IBM System / 360
Speicherbelegung



Name	Operation	Oper
1 8	10 14	20
E.B.E.R. . . .	D.S. . .	Ø.C.L.2.5. .
. . . . .	D.S. . .	C.L.3. . . .
E.N.R.	D.S. . .	C.L.6. . . .
E.F.E.L.D.1. .	D.S. . .	C.L.4. . . .
E.F.E.L.D.2. .	D.S. . .	C.L.4. . . .
E.F.E.L.D.3. .	D.S. . .	C.L.4. . . .
E.F.E.L.D.4. .	D.S. . .	C.L.4. . . .

Abb. 5

Übungen

Verschiedene Hauptspeicheradressen in der Speicherbelegung der Abb. 5 sollen als Adreßkonstanten definiert werden. Die Namen der Konstanten sollen ADKON1, ADKON2, ADKON3, usw. lauten.

Entnehmen Sie bitte Ihren Arbeitsmitteln ein Programmschema und definieren Sie die folgenden Hauptspeicheradressen als Adreßkonstanten. Die in Klammern gesetzten Symbole sind im Operanden der DC-Anweisung zu verwenden.

a) Adresse 1010 (EFELD1)

b) Adresse 1010 (EBER)

c) Adresse 1014 (EFELD2)

d) Adresse 1014 (EFELD4)

e) Adresse 1010 (augenblicklicher Wert des Zuordnungszählers: 1 100)

	Name (1–8)	Operation (10–14)	Operand und Be (20–25)
a)	ADKON1	DC	A(EFELD1)
b)	ADKON2	DC	A(EBER+9)
c)	ADKON3	DC	A(EFELD2)
d)	ADKON4	DC	A(EFELD4-8)
e)	ADKON5	DC	A(*-90)

Entnehmen Sie nun Ihren Arbeitsmitteln die Merkblätter 304 (Seite 2) und 305.

Merkblatt 304 (Seite 2) enthält eine Zusammenfassung zu den Konstantentypen F, H und A. In Merkblatt 305 finden Sie eine formale Darstellung des Aufbaus der Operanden von DC-Anweisungen.

DAS FORMAT DES OPERANDEN BEI DER DEFINITION

DER FELDER UND BEREICHE

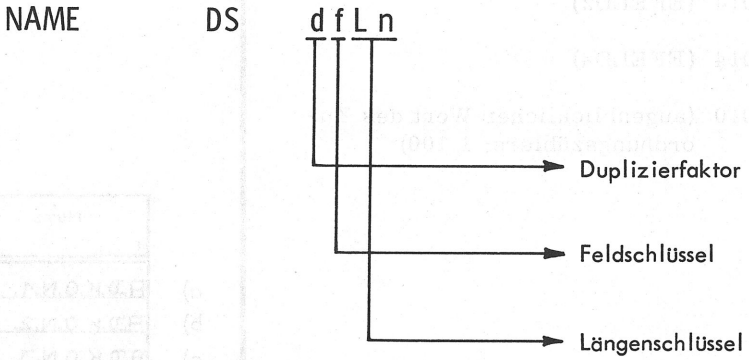


Abb. 1

Nachtrag zur DS-Anweisung

Betrachten Sie bitte Abb. 1. -

In dieser Abbildung ist das Format des Operanden einer DS-Anweisung dargestellt, wie Sie es bisher kennengelernt und angewendet haben.

Der Operand einer DS-Anweisung setzt sich aus mehreren Elementen zusammen.

Der Duplizierfaktor (d), der angegeben werden kann, ist eine Dezimalzahl, die die Anzahl der zu reservierenden Felder angibt.

Der Feldschlüssel (f) bezeichnet die Art und der Längenschlüssel (Ln) die Länge des zu reservierenden Feldes.

Sie haben in diesem Lehrgang bisher nur den Feldschlüssel C für die Definition variabel großer Felder und Bereiche kennengelernt.

Wie Sie wissen, werden für Dualdaten Formate fester Länge (Halbwort, Wort, Doppelwort) benötigt. Dementsprechend gibt es auch die Möglichkeit, Felder von Halbwort-, Wort- und Doppelwortgröße im Hauptspeicher zu reservieren.

Es stehen uns hierzu folgende Feldschlüssel zur Verfügung:

a) H für Halbworte,

b) F für Worte,

c) D für Doppelworte.

1

	Name (1 … 8)	Operation (10 … 14)	
a)	F,E,L,D,H, . .	D,S, . .	H, . .
b)	F,E,L,D,F, . .	D,S, . .	6,F, .
c)	F,E,L,D,D, . .	D,S, . .	D, . .

Durch diese Anweisungen werden im Hauptspeicher Felder mit den festen Längen von

a) einem _____,

b) _____(ZW)_____,

c) einem _____ reserviert.

a) Halbwort

b) 6 Worten

c) Doppelwort

2 Bei der Definition von Halbwort-,. Wort- und Doppelwortfeldern entfällt die Angabe eines Längenschlüssels, da diesen Formaten für Dualdaten schon eine feste Länge zugeordnet ist.

Geben Sie bitte die DS-Anweisungen an, die folgende Bedingungen erfüllen:

a) es sollen 3 Halbworte reserviert werden (Name: FELDH)

b) es soll ein Wort reserviert werden (Name: FELDF)

c) es sollen 10 Doppelworte reserviert werden (Name: FELDD)

	Name (1–8)	Operation (10–14)	(20)
a)	F.E.L.D.H. . .	D.S. . .	3,H. . .
b)	F.E.L.D.F. . .	D.S. . .	F. . . .
c)	F.E.L.D.D. . .	D.S. . .	1,Ø.D. .

3 Wir merken uns:
In den Operanden von DS-Anweisungen können folgende Feldschlüssel verwendet werden:

a) für eine variable Anzahl von Bytes der Buchstabe _____,

b) für das Halbwort der Buchstabe _____,

c) für das Wort der Buchstabe _____,

d) für das Doppelwort der Buchstabe _____.

a) C

b) H

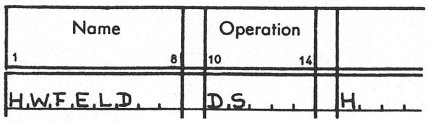
c) F

d) D

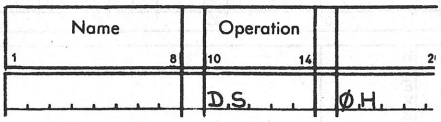
4 Ein Längenschlüssel (Ln) wird nur bei dem Feldschlüssel _____ angegeben, denn Halbwort, Wort und Doppelwort haben eine feste Länge von ___(Z)___, ___(Z)___ und ___(Z)___ Bytes.

C

2, 4 (und) 8



Werden Halbwort-, Wort- oder Doppelwortfelder definiert, so wird diesen Feldern im Hauptspeicher automatisch eine Halbwort-, Wort- bzw. Doppelwortadresse zugeordnet.

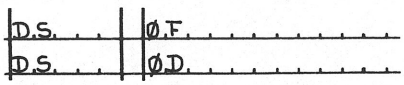


Gibt man im Operanden der DS-Anweisung als Duplizierfaktor eine 0 an, so wird lediglich der Zuordnungszähler auf die nächste Halbwort- bzw. Wort- oder Doppelwortadresse eingestellt. Es erfolgt jedoch keine Platzreservierung wie in der oberen DS-Anweisung.

Ein Symbol im Namenfeld wird für solche Anweisungen nicht benötigt.

Diese Form der DS-Anweisung (z. B. DS 0H) kann dazu benutzt werden, nachfolgenden Daten im Hauptspeicher eine Halbwort-, Wort- bzw. Doppelwortadresse zu sichern.

5 Wie lauten die DS-Anweisungen, die veranlassen, daß der Zuordnungszähler auf eine Wort- bzw. Doppelwortbegrenzung eingestellt wird (ohne Platzreservierung) ?



Schauen Sie sich nun bitte Merkblatt 306 (2 Seiten) an. -

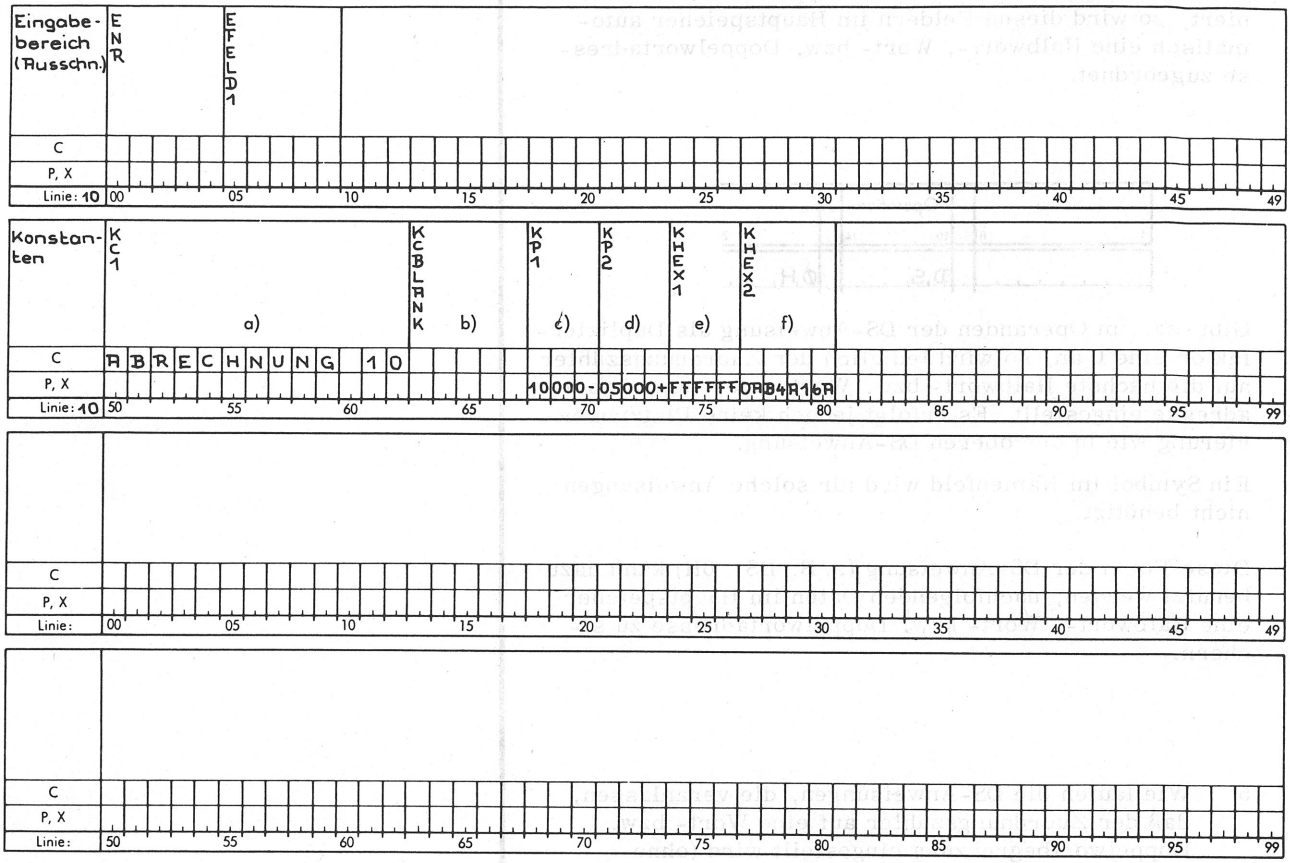

Eingabe-bereich (Ausschn.)
ENR
FELD1
C
P, X
Linie: 10
00 05 10 15 20 25 30 35 40 45 49
Konstan-ten
KC1
a)
KBEZNK
b)
KP1
c)
KP2
d)
KEX1
e)
KEX2
f)
ABRECHNUNG 10
10000-05000+FFFFFFFORB4R16R
Linie: 10
50 55 60 65 70 75 80 85 90 95 99
C
P, X
Linie:
00 05 10 15 20 25 30 35 40 45 49
C
P, X
Linie:
50 55 60 65 70 75 80 85 90 95 99

Abb. 1

FRAGEN ZUM KAPITEL IV

Beantworten Sie bitte zuerst alle Fragen, ehe Sie Ihre Antworten überprüfen. Benutzen Sie ein Extrablatt.

1. Welche der folgenden Aussagen sind richtig, welche falsch ?

   a) Jedes Zeichen einer Zeichenkonstanten wird im Hauptspeicher in einem Halbbyte verschlüsselt.

   b) Eine gepackte Konstante benötigt pro Ziffer bzw. Vorzeichen im Hauptspeicher ein Halbbyte.

   c) Das Vorzeichen einer gepackten Konstanten wird im linken Halbbyte des rechten Bytes dargestellt.

   d) Jede Ziffer einer Hexadezimalkonstanten wird in einem Halbbyte binär verschlüsselt.

2. Entnehmen Sie bitte Ihren Arbeitsmitteln ein Programmschema und definieren Sie die Konstanten in der Speicherbelegung der Abb. 1 (a - f). Die Blank-Konstante (b) soll mit Hilfe des Längenschlüssels definiert werden.

3. Die Adresse 1005 (siehe Speicherbelegung Abb. 1) soll als Adreßkonstante (Name: ADKON) definiert werden.

4. Wortkonstanten und Halbwortkonstanten haben eine feste Länge von __(Z)__ bzw. ___(Z)__ Bytes. Sie werden als _____ gespeichert. Das Vorzeichen befindet sich im ____(WW)___ eines Wortes bzw. Halbwortes.

5. Definieren Sie bitte in einem Programmschema die folgenden Wort- und Halbwortkonstanten:

a)	KF1 : 1 000	}	Wortkonstanten
b)	KF2 : - 500	}	
c)	KH1 : - 850	}	Halbwortkonstanten
d)	KH2 : 32 599	}	

ANTWORTEN ZUM KAPITEL IV

Überprüfen Sie nun Ihre Antworten. Beseitigen Sie erst alle Unklarheiten, ehe Sie fortfahren. Wiederholen Sie gegebenenfalls den Lernstoff zu einzelnen Fragen (Seitenangabe in Klammern).

1. a) Falsch. - Jedes Zeichen einer Zeichenkonstanten wird in einem Byte verschlüsselt.
(Seite 187-191)

b) Richtig.

c) Falsch. - Das Vorzeichen wird im rechten Halbbyte des rechten Bytes dargestellt.
(Seite 209-221)

d) Richtig.
(Seite 223-235)

2.

	Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)
a)	K,C,1, ...	D,C, . . .	C,',B,B,R,E,C,H,N,U,N,G, ,1,Ø,', ...
b)	K,C,B,L,B,N,K,	D,C, . . .	C,L,5,', ,', ...
c)	K,P,1, ...	D,C, . . .	P,',-,1,Ø,Ø,Ø,Ø,', oder: X,',1,Ø,Ø,Ø,Ø,D,', .
d)	K,P,2, ...	D,C, . . .	P,',5,Ø,Ø,Ø,', oder: P,',+,5,Ø,Ø,Ø,', . .
e)	K,H,E,X,1, . .	D,C, . . .	X,',F,F,F,F,F,F,', ...
f)	K,H,E,X,2, . .	D,C, . . .	X,',B,B,4,B,1,6,B,', ...

(Seite 187-235)

3.

B,D,K,O,N, . .	D,C, . . .	B,(,E,F,E,L,D,1,), .
oder		
B,D,K,O,N, . .	D,C, . . .	B,(,E,N,R,+,5,), .

(Seite 249-257)

4. Wortkonstanten und Halbwortkonstanten haben eine feste Länge von 4 bzw. 2 Bytes. Sie werden als Dualzahlen gespeichert. Das Vorzeichen befindet sich im linken Bit eines Wortes bzw. Halbwortes.
(Seite 237-247)

5.

	Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)
a)	KF1	DC	F'1000' oder F'+1000'
b)	KF2	DC	F'-500'
c)	KH1	DC	H'-850'
d)	KH2	DC	H'32599' oder: H'+32599'

(Seite 237-247)



Band 4, Seite 459 (Operationen für Dualdaten A bis LA)

KAPITEL V

DIE ASSEMBLERANWEISUNGEN USING UND DROP

Lernstoff:

Die Assembleranweisung USING

Die Assembleranweisung DROP

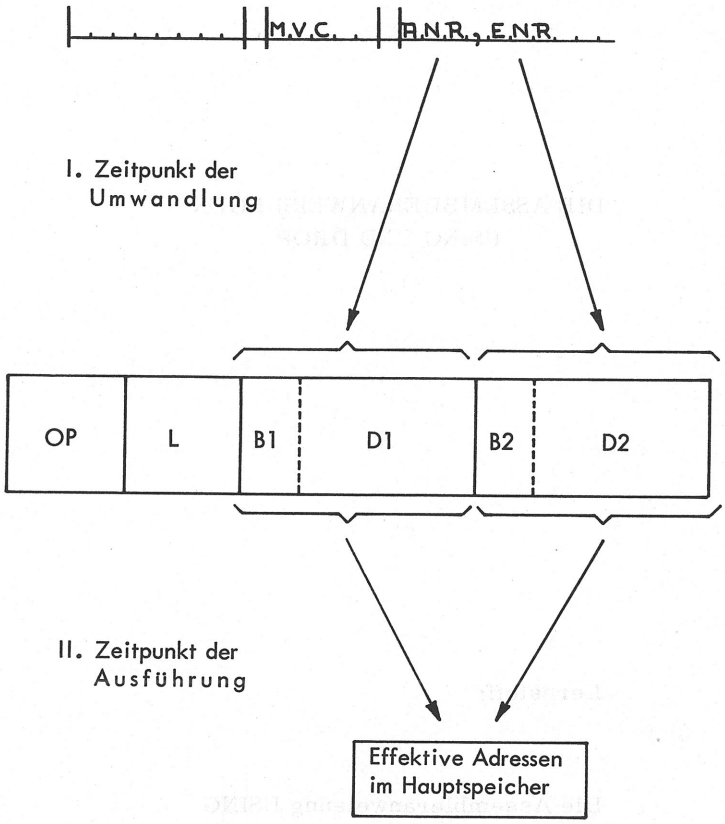

M.V.C.
A.N.R., E.N.R.
I. Zeitpunkt der Umwandlung
OP
L
B1
D1
B2
D2
II. Zeitpunkt der Ausführung
Effektive Adressen im Hauptspeicher

Abb. 1

In den Instruktionen eines Programms adressieren wir Hauptspeicherstellen symbolisch, d.h. wir vergeben für die tatsächlichen Hauptspeicheradressen symbolische Namen (z.B. ANR, ENR).

Bei der Umwandlung des Programms wird eine symbolische Adresse, die der Programmierer vergeben hat, in die Nummer eines Basisregisters (B1 bzw. B2) und die relative Adresse (D1 bzw. D2) in der echten Maschineninstruktion umgewandelt. Die effektive Hauptspeicheradresse, die z.B. durch ANR angegeben ist, wird also in der umgewandelten Maschineninstruktion in 2 Komponenten aufgegliedert:

a) in die Nummer eines Basisregisters, das die Basisadresse enthält und

b) in die relative Adresse.

Betrachten Sie bitte Abb. 1.-

Für die Adressierung der Felder im Hauptspeicher des Systems /360 sind zwei Zeitpunkte von Bedeutung:

a) der Zeitpunkt der Umwandlung,

b) der Zeitpunkt der Ausführung des Programms.

Für den Zeitpunkt der Umwandlung des symbolischen Programms müssen wir dem Umwandlungsprogramm mitteilen, welches Register als Basisregister verwendet werden soll und welcher Wert als Basisadresse für die Errechnung der relativen Adressen zu benutzen ist. Nur so ist es dem Umwandlungsprogramm möglich, z.B. für ANR in der echten Maschineninstruktion die Nummer eines Basisregisters und die errechnete relative Adresse anzugeben.

Diese Mitteilung an das Umwandlungsprogramm übernimmt die Assembleranweisung USING, die nach der Umwandlung im Maschinenprogramm nicht mehr vorhanden ist.

Zum Zeitpunkt der Ausführung des Programms müssen aus den Angaben in den echten Instruktionen für die Verarbeitung der Daten wieder Hauptspeicheradressen (effektive Adressen) errechnet werden. Hierzu muß nun dafür gesorgt sein, daß tatsächlich die für die Errechnung der relativen Adressen angegebene Basisadresse auch in dem entsprechenden Basisregister steht. Denn durch die Assembleranweisung USING werden lediglich Angaben für die Umwandlung gemacht, die angegebene Basisadresse steht jedoch noch nicht in dem zu benutzenden Basisregister.

Um das Basisregister mit der entsprechenden Basisadresse zu versehen, benötigen wir eine Maschineninstruktion, die Sie noch kennenlernen werden.

Wir wenden uns nun der USING-Anweisung zu, die dem Assembler-Umwandlungsprogramm Angaben zum Basisregister und zur Basisadresse zur Verfügung stellt.

USING

VERWENDEN BASISREGISTER

(Use base address register)

Die USING-Anweisung wird dazu benutzt, für die Umwandlung ein Register als Basisregister zu kennzeichnen und eine Basisadresse anzugeben.

Beispiel:

Name	Operation	Op
1 8	10 14	20
. . . . .	U.S.I.N.G	*,8. . .

1. Der symbolische Operationsschlüssel für die Assembleranweisung "Verwenden Basisregister " lautet USING.
2. Der 1. Operand der USING-Anweisung gibt in symbolischer Form den Wert an, der als Basisadresse zur Errechnung der relativen Adressen zu verwenden ist (z.B.*, d.h. momentaner Stand des Zuordnungszählers).
3. Der 2. Operand bezeichnet ein allgemeines Register (z.B. 8), das als Basisregister verwendet werden soll.
4. Das Namenfeld bleibt in einer USING-Anweisung unbenutzt.
5. Die USING-Anweisung kann in einem Programm mehrmals benutzt werden.

Abb. 2

1 Lesen Sie bitte zunächst den Text in Abb. 2 aufmerksam durch. -

Der symbolische Operationsschlüssel, mit Hilfe dessen ein Basisregister und eine Basisadresse für das Assembler-Umwandlungsprogramm bereitgestellt werden, lautet _____.

USING

2 Der 1. Operand einer USING-Anweisung gibt die _____ an, die für die Errechnung der relativen Adressen benötigt wird.

Basisadresse

3 Der 2. Operand einer USING-Anweisung ist ein dezimaler Direktwert, der ein allgemeines Register bezeichnet, das als _____ benutzt werden soll.

Basisregister

4 Betrachten Sie die Codierung der Abb. 2. -

Der 1. Operand in dieser USING-Anweisung ist ein Stern (*). Dieser Stern ist ein Symbol, das den augenblicklichen Stand des Zuordnungszählers angibt.

Wie Sie bereits wissen, zeigt der Zuordnungszähler, der den Feldern, Konstanten und Instruktionen im Hauptspeicher nacheinanderfolgende Hauptspeicherstellen zuordnet, immer die Adresse der nächsten verfügbaren Hauptspeicherstelle an.

Der augenblickliche Stand oder Wert des Zuordnungszählers ist also die als nächstes verfügbare _____ im Hauptspeicher.

Adresse

5 Der augenblickliche Stand des Zuordnungszählers wird symbolisch durch einen _____ bezeichnet.

Stern (das Symbol * wird auch als Sternadresse bezeichnet)

USING

VERWENDEN BASISREGISTER

(Use base address register)

Die USING-Anweisung wird dazu benutzt, für die Umwandlung ein Register als Basisregister zu kennzeichnen und eine Basisadresse anzugeben.

Beispiel:

Name	Operation	Op
1 8	10 14	20
	U,S,I,N,G	*,8

1. Der symbolische Operationsschlüssel für die Assembleranweisung "Verwenden Basisregister " lautet USING.
2. Der 1. Operand der USING-Anweisung gibt in symbolischer Form den Wert an, der als Basisadresse zur Errechnung der relativen Adressen zu verwenden ist (z.B.*, d.h. momentaner Stand des Zuordnungszählers).
3. Der 2. Operand bezeichnet ein allgemeines Register (z.B. 8), das als Basisregister verwendet werden soll.
4. Das Namenfeld bleibt in einer USING-Anweisung unbenutzt.
5. Die USING-Anweisung kann in einem Programm mehrmals benutzt werden.

Abb. 2

6 Der Stern im 1. Operanden der USING-Anweisung in Abb. 2 teilt dem Assembler-Umwandlungsprogramm also mit, daß der augenblickliche Wert, den der Zuordnungszähler gerade anzeigt, als _____ verwendet werden soll.

Basisadresse

7 Wenn der augenblickliche Stand des Zuordnungszählers 1050 ist, dann bewirkt der Stern im 1. Operanden der USING-Anweisung, daß die Basisadresse __(Z)__ zur Errechnung der relativen Adressen verwendet wird.

1050

8 Der 2. Operand der USING-Anweisung in Abb. 2 gibt an, daß das allgemeine Register __(Z)__ als _____ zu benutzen ist.

8

Basisregister

9 Das Register 8 hat hiermit eine besondere Funktion erhalten. Es kann nun im gleichen Programm nicht mehr zum Rechnen mit Dualdaten benutzt werden, da es zum Basisregister erklärt ist und für die Adreßbildung benötigt wird.

Wir fassen zusammen:

| . . . . . | | U.S.I.N.G | | *, 8 . . . |

Durch diese USING-Anweisung wird dem Assembler-Umwandlungsprogramm mitgeteilt,

a) daß Register __(Z)__ als _____ benutzt werden soll und

b) daß der augenblickliche Stand des Zuordnungszählers als _____ für die Errechnung der relativen Adressen dienen soll.

8

Basisregister

Basisadresse

10

```
|        | |USING| |*,9,  |
```

Kommentieren Sie bitte diese USING-Anweisung selbst.

Sinngemäß:
Diese Anweisung teilt dem Umwandlungsprogramm mit, daß Register 9 als Basisregister benutzt werden soll. Als Basisadresse soll der augenblickliche Wert des Zuordnungszählers verwendet werden.

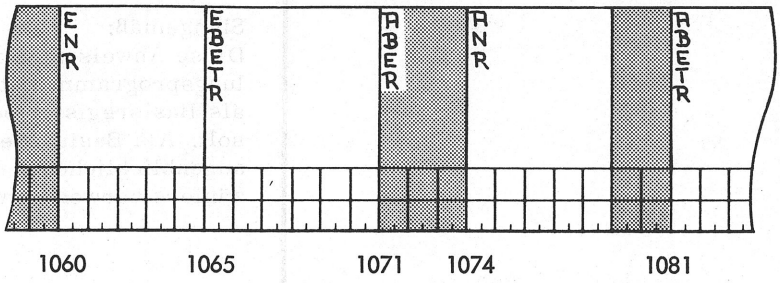
11 Wir merken uns:
In der USING-Anweisung wird für die Umwandlung angegeben, welches Register als _____ benutzt werden soll und welcher Wert für die Errechnung der relativen Adressen als _____ zu verwenden ist.

Basisregister

Basisadresse

Beachten Sie bitte:

Register 0 darf nicht als Basisregister angegeben werden.



Stand des Zuordnungszählers:

	Name (1–8)	Operation (10–14)	Operand und Be (20–25)
1000		U.S.I.N.G	*,,1Ø
1060	E.N.R.	D.S.	C.L.5
1065	E.B.E.T.R.	D.S.	C.L.6
1071	A.B.E.R.	D.S.	Ø.C.L.2.8
		D.S.	C.L.3
1074	A.N.R.	D.S.	C.L.5
		D.S.	C.L.2
1081	A.B.E.T.R.	D.S.	C.L.6
		M.V.C.	A.N.R.,E.N.R.

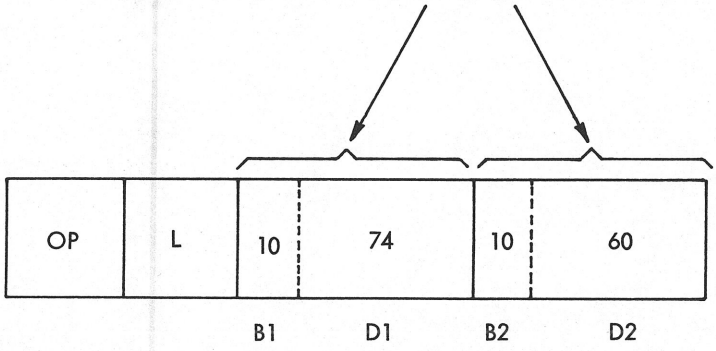


Abb. 3

Die folgenden Lernelemente in Verbindung mit Abb. 3 verdeutlichen die Notwendigkeit der USING-Anweisung für das Errechnen der relativen Adressen.

12 Betrachten Sie bitte Abb. 3. -

In der MVC-Instruktion der Abb. 3 hat der Programmierer die symbolischen Adressen ANR und ENR angegeben.

In der echten Maschineninstruktion werden diese symbolischen Adressen nach der Umwandlung jeweils durch das Basisregister und die relative Adresse dargestellt.

Dreierlei benötigt das Assembler-Umwandlungsprogramm, um eine symbolische Adresse (z. B. ANR) in echter Form (B1, D1) darstellen zu können:

a) die effektive Adresse (siehe Speicherbelegung), die durch den Zuordnungszähler gegeben ist,

b) die Angabe des zu verwendenden Basisregisters,

c) die Angabe der zu benutzenden Basisadresse.

Die effektiven Adressen werden mit Hilfe des Zuordnungszählers durch die Definition der Felder und Bereiche angegeben.

Das zu verwendende Basisregister und die für die Errechnung der relativen Adressen benötigte Basisadresse werden durch die _____ -Anweisung angegeben.

USING

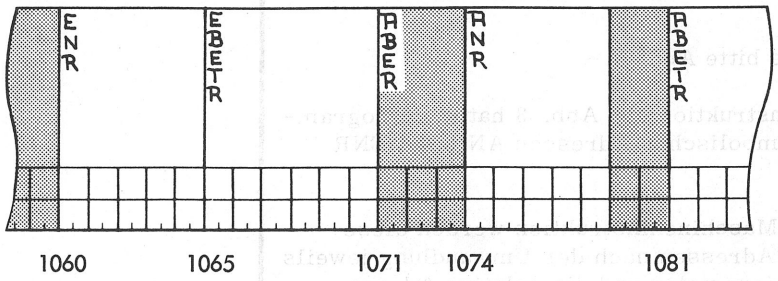
13 Nachdem also dem Umwandlungsprogramm die effektiven Adressen der Felder, das zu verwendende Basisregister und die zu benutzende Basisadresse bekannt sind, können ANR und ENR in der Form B1, D1 und B2, D2 in der echten Maschineninstruktion dargestellt werden.

In Abb. 3 wird ANR in der umgewandelten Instruktion durch die Angabe des Basisregisters __(Z)__ und der _____ Adresse __(Z)__ dargestellt.

10

relativen

74



Stand des Zuordnungszählers:

	Name (1–8)	Operation (10–14)	Operand und Be (20–25)
		↓	
1000		U,S,I,N,G	*,,1,Ø,
		↓	
1060	E,N,R,	D,S,	C,L,5,
1065	E,B,E,T,R,	D,S,	C,L,6,
1071	A,B,E,R,	D,S,	Ø,C,L,2,8,
		D,S,	C,L,3,
1074	A,N,R,	D,S,	C,L,5,
		D,S,	C,L,2,
1081	A,B,E,T,R,	D,S,	C,L,6,
		↓	
		M,V,C,	A,N,R,,E,N,R,

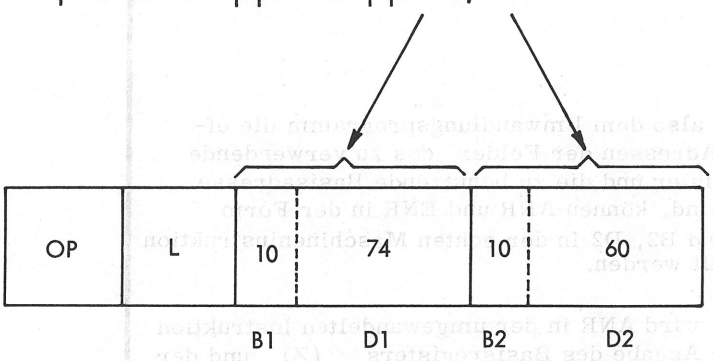


Abb. 3

14 Die relative Adresse ist die Differenz aus der effektiven Adresse __(Z)__ und der Basisadresse __(Z)__ (siehe Abb. 3).

1074

1000 (angenommener Wert des Zuordnungszählers)

15 ENR wird in echter Form durch die Angabe des __(WZ)__ und der _____ Adresse __(Z)__ dargestellt.

Basisregisters 10

relativen

60

16 Wie ergibt sich die relative Adresse 60 für ENR ?

Effektive Adresse 1060 minus Basisadresse 1000

17 Welche relativen Adressen würden sich aufgrund der Angaben in Abb. 3 ergeben für:

a) EBETR

b) ABER

c) ABETR

a) 65

b) 71

c) 81

Damit alle symbolischen Adressen eines Programms in der echten Form (B1, D1 und B2) dargestellt werden können, müssen gleich zu Anfang des Programms durch USING ein Basisregister und eine Basisadresse angegeben werden.

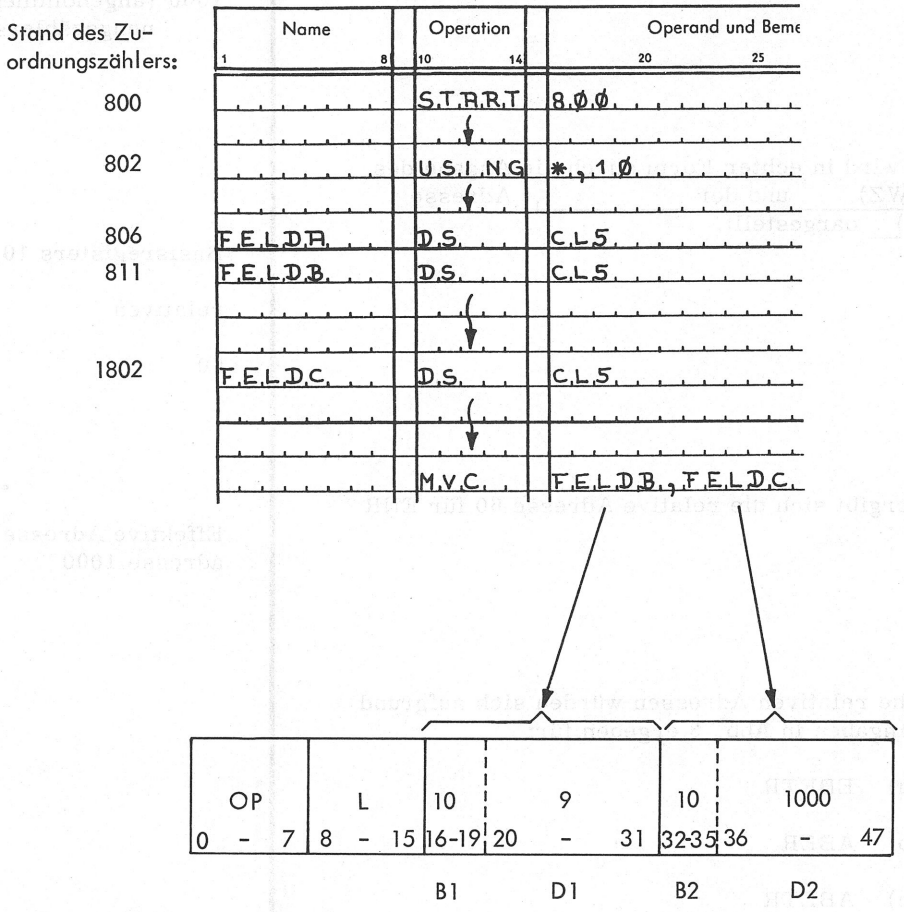

Stand des Zu-
ordnungszählers:
Name
Operation
Operand und Beme
1
8
10
14
20
25
800
S,T,R,R,T
8,Ø,Ø,
802
U,S,I,N,G
*,,1,Ø,
806
F,E,L,D,A,
D,S,
C,L,5,
811
F,E,L,D,B,
D,S,
C,L,5,
1802
F,E,L,D,C,
D,S,
C,L,5,
M,V,C,
F,E,L,D,B,,F,E,L,D,C,
OP
L
10
9
10
1000
0 - 7
8 - 15
16-19
20 - 31
32-35
36 - 47
B1
D1
B2
D2

Abb. 4

18 Betrachten Sie bitte Abb. 4. -

Durch die START-Anweisung wird der Zuordnungszähler auf 800 eingestellt. Nach der START-Anweisung ist in Abb. 4 Platz freigehalten für die Instruktion, die zum Einbringen der Basisadresse in das Basisregister benötigt wird. Diese Instruktion, die in Abb. 4 noch nicht aufgeführt ist, benötigt im Hauptspeicher 2 Bytes, nämlich die Speicherstellen 800 bis 801.

Der Zuordnungszähler, der immer die nächste verfügbare Hauptspeicheradresse anzeigt, steht also nach dieser 2 Bytes großen Instruktion auf __(Z)__.

802

19 An dieser Stelle wird nun dem Assembler mitgeteilt, daß Register __(Z)__ als _____ benutzt und der augenblickliche Wert des Zuordnungszählers (802) als _____ verwendet werden soll.

10

Basisregister

Basisadresse

20 Für das Errechnen der relativen Adresse von FELDA (Abb. 4) ist dem Assembler nun durch die USING - Anweisung die Basisadresse 802 und durch den weitergestellten Zuordnungszähler die effektive Adresse 806 (siehe Abb. 4) gegeben.

Für FELDA wird somit bei der Umwandlung die __(WW)__ __(Z)__ errechnet.

relative Adresse 4

21 Nach den Angaben in Abb. 4 ergibt sich für FELDB die relative Adresse __(Z)__ und für die FELDC die relative Adresse __(Z)__.

9

1000

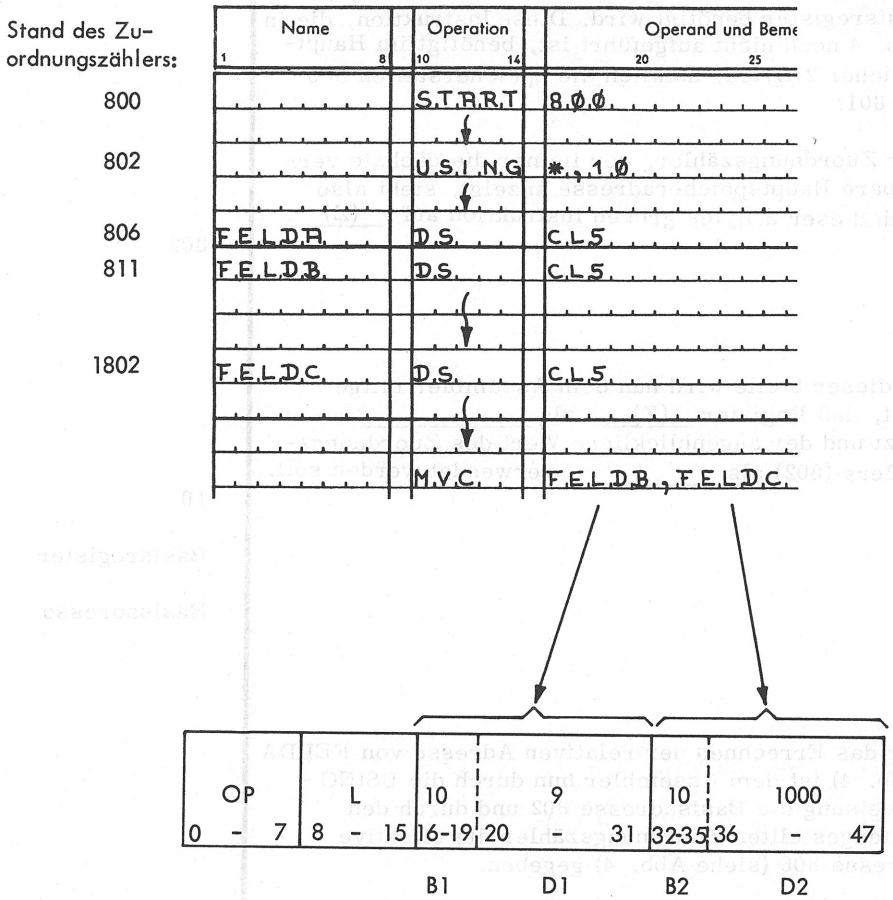

Stand des Zuordnungszählers:
Name
Operation
Operand und Beme
1
8
10
14
20
25
800
802
806
811
1802
S,T,R,R,T
8,Ø,Ø,
U,S,I N,G
*,,1,Ø,
F,E,L,D,B,
D,S,
C,L,5,
F,E,L,D,B,
D,S,
C,L,5,
F,E,L,D,C,
D,S,
C,L,5,
M,V,C,
F,E,L,D,B,,F,E,L,D,C,
OP
L
10
9
10
1000
0 - 7
8 - 15
16-19
20 - 31
32-35
36 - 47
B1
D1
B2
D2

Abb. 4

22 Aus Abb. 4 ersehen Sie, wie die Adressen nach der Umwandlung in der echten Maschineninstruktion dargestellt sind.

Für die Darstellung einer relativen Adresse in der umgewandelten Maschineninstruktion stehen jeweils 12 Bits zur Verfügung. In diesen 12 Bits können relative Adressen von 0 bis 4095 dual dargestellt werden.

Die relative Adresse für ein Feld in einer Maschineninstruktion kann also maximal __(Z)__ betragen.

4095

23 Kann die relative Adresse eines Feldes in 12 Bits dual dargestellt werden, wenn die Basisadresse 802 und die effektive Adresse 4 900 beträgt ?

Nein

(Bei einer Basisadresse 802 und einer effektiven Adresse 4900 ergäbe sich eine relative Adresse von 4098. Diese kann aber nicht in 12 Bits dual dargestellt werden).

24 Sie sehen, die Differenz zwischen effektiver Adresse und Basisadresse darf nie größer sein als __(Z)__.

4095

Heißt das nun, daß ein Programm nicht mehr als 4095 Bytes umfassen darf, damit wir immer gültige relative Adressen erhalten ? - Nein.

Wenn ein Programm größer als 4095 Bytes ist, was sehr oft der Fall sein wird, dann müssen mehrere Basisregister und Basisadressen für die Adressierung des Hauptspeichers angegeben werden. - Mit diesem Problem, nämlich der Angabe mehrerer Basisregister und Basisadressen, befassen wir uns jedoch nicht in diesem Lehrgang.

Wir fassen nun noch einmal in wenigen Lernelementen zusammen, was Sie von der USING-Anweisung wissen müssen.

25 In der USING-Anweisung wird für die Umwandlung angegeben, welches Register als _____ benutzt werden soll und welcher Wert als _____ für die Errechnung der ____(WW)____ zu verwenden ist.

Basisregister

Basisadresse

relativen Adressen

26 Der 1. Operand einer USING-Anweisung bezeichnet den Wert, der als _____ benutzt werden soll.

Der 2. Operand gibt an, welches Register als _____ zu verwenden ist.

Basisadresse

Basisregister

27 | . . . . . | | U.S.I.N.G | | *.,.13. . . |

In dem 1. Operanden dieser USING-Anweisung wird dem Assembler mitgeteilt, daß der augenblickliche Stand des _____ als _____ zur Errechnung der ____(WW)____ verwendet werden soll.

Im 2. Operanden wird angegeben, daß ____(WZ)____ als _____ zu benutzen ist.

Zuordnungszählers

Basisadresse

relativen Adressen

Register 13

Basisregister

28 Die USING-Anweisung ist für die Adressierung des Hauptspeichers im System /360 unbedingt notwendig !

Denn ohne die Angabe eines Basisregisters und einer Basisadresse könnte bei der Umwandlung in die echten Maschineninstruktionen kein _____ eingesetzt und keine ___(WW)___ errechnet werden.

Basisregister

relative Adresse

Stand des Zuordnungszählers:	Name (1–8)	Operation (10–14)	Operand und Bemerku (20–25)
1048		S,T,A,R,T	1,Ø,4,8,
1050		U,S,I,N,G	*,,8,
1065	E,B,E,R,	D,S,	Ø,C,L,5,Ø,
		D,S,	C,L,4,
1069	E,N,R,	D,S,	C,L,6,
1075	E,N,A,M,E,	D,S,	C,L,1,Ø,
1115	A,B,E,R,	D,S,	Ø,C,L,3,Ø,
		D,S,	C,L,5,
1120	A,N,R,	D,S,	C,L,6,
1126	A,N,A,M,E,	D,S,	C,L,1,Ø,
		M,V,C,	A,N,R,,E,N,R,
		M,V,C,	A,N,A,M,E,,E,N,A,M,E,

Abb. 5

Übungen

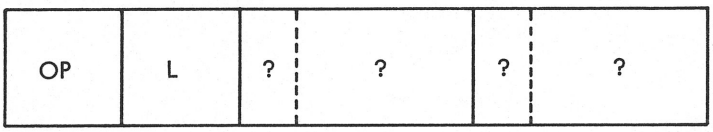
1. Wie muß die USING-Anweisung lauten, wenn Register 5 als Basisregister zu benutzen und der augenblickliche Wert des Zuordnungszählers als Basisadresse zu verwenden ist ?

2. Betrachten Sie bitte die Codierung in Abb. 5. -

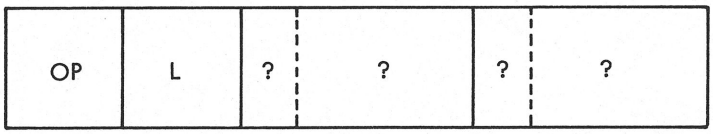
Wie werden die symbolischen Adressen ANR, ENR und ANAME, ENAME der beiden MVC-Instruktionen nach der Umwandlung dargestellt ?

Vervollständigen Sie bitte die beiden umgewandelten Maschineninstruktionen:

a) | . . . . . | M.V.C. . | A.N.R.,E.N.R. . . .

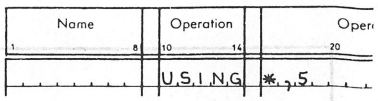


b) | . . . . . | M.V.C. . | A.N.A.M.E.,E.N.A.M.E. .

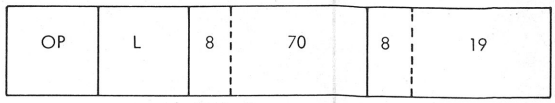


Überprüfen Sie nun bitte Ihre Ergebnisse auf der folgenden Seite !

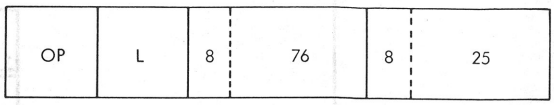
1.



2.a)



b)



Durch die Assembleranweisung USING haben wir dem Assembler lediglich für die Umwandlung mitgeteilt, welches Basisregister und welche Basisadresse für die Darstellung der Adressen in der umgewandelten Maschineninstruktion benutzt werden sollen. Damit steht jedoch die Basisadresse noch n i c h t in dem angegebenen Basisregister.

Wir müssen also durch eine besondere Instruktion noch dafür sorgen, daß der Wert, den wir in der USING-Anweisung als Basisadresse angeben, auch in das entsprechende Register, das als Basisregister benutzt wird, hineingebracht wird. Wir müssen die Basisadresse, die wir in der USING-Anweisung angeben, in das Basisregister l a d e n.

Das L a d e n eines Basisregisters mit der Basisadresse, das zu der USING-Anweisung also noch hinzukommen muß, lernen Sie bei der Besprechung der BALR-Instruktion kennen.



Band 4, Seite 432 (BALR)

DROP

AUFGEBEN BASISREGISTER

(Drop Base Address Register)

Beispiel:

Name	Operation	Op
1 8	10 14	20
. . . . .	D.R.O.P.	8. . . . .

1. Der symbolische Operationsschlüssel für die Assembleranweisung "Aufgeben Basisregister" lautet DROP.
2. Im Operandenfeld wird das Register angegeben, das nicht mehr als Basisregister verwendet werden soll.
3. Das Namenfeld bleibt unbenutzt.
4. Die Assembleranweisung DROP kann an jeder Stelle eines Programms benutzt werden.
5. Ein Basisregister, das durch eine DROP-Anweisung aufgegeben wurde, kann jederzeit durch USING wieder als solches eingesetzt werden.

Abb. 1

Durch die USING-Anweisung wird ein Register als Basisregister gekennzeichnet. - Die DROP-Anweisung bewirkt, daß ein bisher als Basisregister verfügbares Register nicht mehr als Basisregister verwendet wird.

Es soll an dieser Stelle nicht besprochen werden, in welchen Fällen der Anwendung die DROP-Anweisung zu benutzen ist. Das würde den Rahmen dieses Lehrganges sprengen. Wir können an dieser Stelle nur aufzeigen, daß es grundsätzlich möglich ist, ein durch USING eingesetztes Basisregister auch wieder aufzugeben und wie dies mit Hilfe der DROP-Anweisung geschehen muß.

Lesen Sie hierzu bitte zunächst den Text in Abb. 1 aufmerksam durch und wenden Sie sich dann den folgenden Lernelementen zu.

1 Dem Assembler wird ein allgemeines Register durch die USING-Anweisung als Basisregister zur Verfügung gestellt.

Ein solches durch USING zur Verfügung gestelltes Basisregister kann aber durch den Programmierer wieder aufgegeben werden. Das geschieht durch die Assembleranweisung _____.

DROP

2 Die DROP-Anweisung bewirkt also, daß ein vorher durch USING angegebenes Basisregister nun nicht mehr als _____ verwendet wird.

Basisregister

3 Im Operandenfeld der DROP-Anweisung wird angegeben, welches Register nicht mehr als _____ verwendet werden soll.

Basisregister

4 Durch die Codierung in Abb. 1 wird dem Assembler mitgeteilt, daß Register __(Z)__ nicht mehr als Basisregister zur Verfügung steht.

8

DROP

AUFGEBEN BASISREGISTER

(Drop Base Address Register)

Beispiel:

Name	Operation	Op
1 8	10 14	20
	D.R.O.P.	8.

1. Der symbolische Operationsschlüssel für die Assembleranweisung "Aufgeben Basisregister" lautet DROP.
2. Im Operandenfeld wird das Register angegeben, das nicht mehr als Basisregister verwendet werden soll.
3. Das Namenfeld bleibt unbenutzt.
4. Die Assembleranweisung DROP kann an jeder Stelle eines Programms benutzt werden.
5. Ein Basisregister, das durch eine DROP-Anweisung aufgegeben wurde, kann jederzeit durch USING wieder als solches eingesetzt werden.

Abb. 1

5 In einer DROP-Anweisung bleibt das Namenfeld _____.

Eine DROP-Anweisung kann an einer (bestimmten, beliebigen) _____ Stelle im Programm verwendet werden.

unbenutzt

beliebigen

6 Wie muß die Assembler-Anweisung lauten, wenn in einem Programm fortan das Basisregister 10 nicht mehr zur Adreßbildung benutzt werden soll ?

Name	Operation	Op
1 8	10 14	20
	DROP	10

7 Wir merken uns:
Es ist jederzeit möglich, ein vorher durch USING zur Verfügung gestelltes Basisregister wieder aufzugeben. Das geschieht durch die Assembler-Anweisung _____. -

	DROP	14

Diese Anweisung bewirkt, daß fortan im Programm bei der Bildung der Hauptspeicheradressen Register __(Z)__ nicht mehr als _____ benutzt wird. -

Wurde ein Basisregister durch DROP aufgegeben, so kann es jederzeit durch eine _____ -Anweisung wieder eingesetzt werden.

DROP

14

Basisregister

USING

FRAGEN ZUM KAPITEL V

Prüfen Sie nun bitte, ob Sie den Lernstoff dieses Kapitels beherrschen. Beanworten Sie erst alle Fragen auf einem Extrablatt bzw. Programmschema, bevor Sie Ihre Antworten überprüfen.

1. Der symbolische Operationsschlüssel für die Assembleranweisung "Verwenden Basisregister" lautet _____.

2. | . . . . . | USING | *,9. .

   Was wird dem Assembler

   a) durch den 2. Operanden (9),

   b) durch den 1. Operanden (*) dieser Anweisung mitgeteilt ?

3. Relative Adressen können die Werte __(Z)__ bis __(Z)__ haben.

4. Dem Assembler soll das Register 11 als Basisregister zur Verfügung gestellt werden. Die Basisadresse soll der momentane Stand des Zuordnungszählers sein. - Wie lautet die erforderliche Anweisung ?

5. Ein durch USING eingesetztes Basisregister kann mit Hilfe einer _____ -Anweisung wieder aufgegeben werden.

ANTWORTEN ZUM KAPITEL V

Ehe Sie in diesem Lehrgang fortfahren, sollten Sie erst alle aufgetretenen Unklarheiten beseitigen. Wiederholen Sie die in Klammern angegebenen Seiten dieses Kapitels, wenn Sie in Ihren Antworten Fehler vorfinden.

1. Der symbolische Operationsschlüssel für die Assembleranweisung "Verwenden Basisregister" lautet USING.

   (Seite 276)

2. Sinngemäß:

   Dem Assembler wird mitgeteilt,

   a) daß Register 9 als Basisregister verwendet werden soll,

   b) daß der augenblickliche Wert des Zuordnungszählers als Basisadresse zu benutzen ist.

   (Seite 277 - 281)

3. Relative Adressen können die Werte 0 bis 4095 haben.

   (Seite 289)

4. | | |

KAPITEL VI

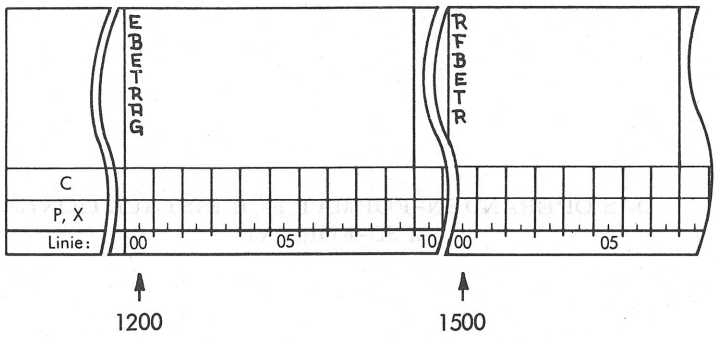
DAS OPERANDEN-FORMAT FÜR INSTRUKTIONEN IM SS-FORMAT

Lernstoff:

Die symbolische Adressierung.

Die explizite Adressierung. - D1 (L1, B1) bzw. D2 (L2, B2).

Besonderheit: die explizite Adressierung im 2. Operanden einer logischen SS-Instruktion. - D2 (B2).



	Name	Operation	Operand und Bemerkungen
	1 8	10 14	20 25 30 35
a)		A,P,	R,F,B,E,T,R,,,E,B,E,T,R,A,G,
b)		A,P,	R,F,B,E,T,R,,,E,B,E,T,R,A,G,+,4,(,6,)

Basisregister: 11 Basisadresse: 1000

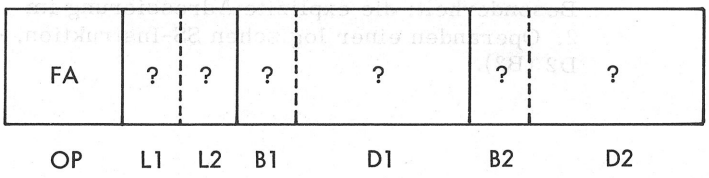


Abb. 1

Dieses abschließende Kapitel zu Band 3 gibt Ihnen einen Überblick über den Aufbau der Operanden einer Instruktion im SS-Format.

Es soll aufgezeigt werden, welche Möglichkeiten Sie als Programmierer haben, um im Laufe des Programms Felder im Hauptspeicher anzusprechen.

Betrachten Sie bitte Abb. 1. -

In dieser Abbildung ist der Ausschnitt einer Speicherbelegung mit den Feldern EBETRAG (Länge 10) und RFBETR (Länge 8) gezeigt. Danach folgen zwei Instruktionen, durch deren Operanden die in der Speicherbelegung dargestellten Felder angesprochen werden. Es ist die Frage gestellt, wie die Operanden und ihre Länge in der echten Instruktion stehen werden.

Wir wollen uns zunächst die Operanden der in Abb. 1 gezeigten Codierung näher anschauen. - In der Codierung der Abb. 1 sind die Operanden in einer Form angegeben, die Sie alle schon aus diesem Lehrgang kennen und auch schon selbst benutzt haben.

Die einfachste Form eines Operanden in einer Speicher-Speicher-Instruktion ist das Symbol ohne Adressrechnung und explizite Längenangabe.

z. B.: RFBETR
EBETRAG

Einen solchen Operanden bezeichnet man auch als einen einfachen Ausdruck.

Neben dem einfachen Ausdruck, z. B. RFBETR, kann man als Operanden in einer SS-Instruktion auch den zusammengesetzten Ausdruck verwenden.
Solche zusammengesetzte Ausdrücke wären

z. B.: RFBETR+1
EBETRAG+4

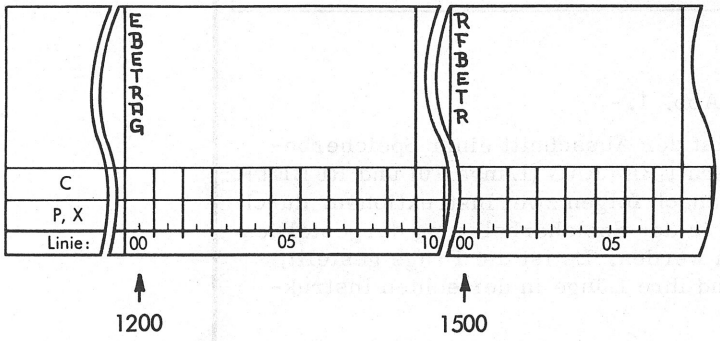
Zusammengesetzte Ausdrücke sind Kombinationen von Symbolen und Direktwerten, die miteinander arithmetisch verknüpft sind (Adressrechnung).

Schließlich gibt es noch die Ihnen schon bekannte Möglichkeit, einfache oder zusammengesetzte Ausdrücke mit einem Unterfeld zu versehen.

z. B.: RFBETR(5)
EBETRAG+4(6)

Ein Unterfeld in einem Operanden wird immer in Klammern gesetzt. In unseren Beispielen sind die Unterfelder die in Klammern angegebenen expliziten Längen 5 und 6.

Alle die hier angeführten Beispiele zu den möglichen Formen der Operanden einer SS-Instruktion sind Ihnen schon bekannt. Sie wissen auch, daß die in Operanden verwendeten Symbole natürlich definiert sein müssen, damit Ihnen ein Adress- und ein Längenattribut zugeordnet werden kann.



	Name	Operation	Operand und Bemerkungen
	1 8	10 14	20 25 30 35
a)		A.P.	R.F.B.E.T.R.,E.B.E.T.R.A.G.
b)		A.P.	R.F.B.E.T.R.,E.B.E.T.R.A.G.+4.(6.)

Basisregister: 11 Basisadresse: 1000

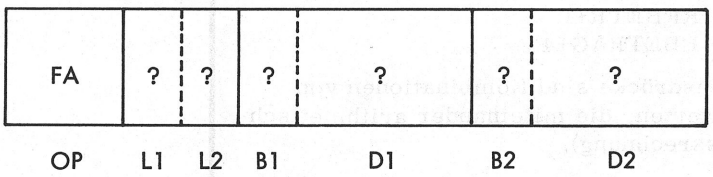
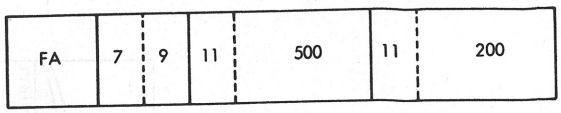


Abb. 1

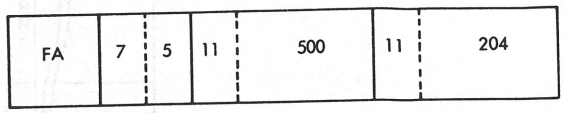
1. Wir wenden uns nun den noch offenen Fragen in Abb. 1 zu. -

Geben Sie bitte auf einem Extrablatt an, wie die Instruktionen in Abb. 1 in der echten Form aussehen. Der Operationsschlüssel ist bereits vorgegeben.

a)



b)

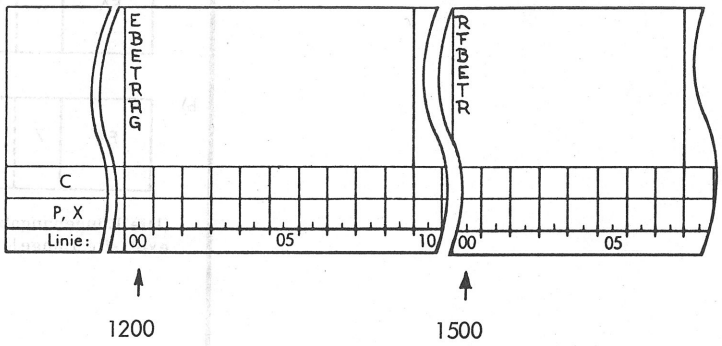


(Die Längenangabe ist um 1 kleiner als die definierte bzw. explizite Länge!)

Wir kommen nun zu einer Möglichkeit der Angabe von Operanden einer SS-Instruktion, die Sie in diesem Lehrgang noch nicht kennengelernt haben.

Ebenso wie die Länge kann auch die A d r e s s e eines Operanden e x p l i z i t angegeben werden. Bei der expliziten Angabe einer Adresse im Operanden einer SS-Instruktion müssen das Basisregister und die relative Adresse ausdrücklich im Operanden selbst angegeben werden.

Diese Form der Angabe eines Operanden im Hauptspeicher benutzt man z. B. für die Adressmodifikation. (In Aufgabe 14 werden wir uns damit ausführlich befassen.)



Basisregister: 11 Basisadresse: 1000

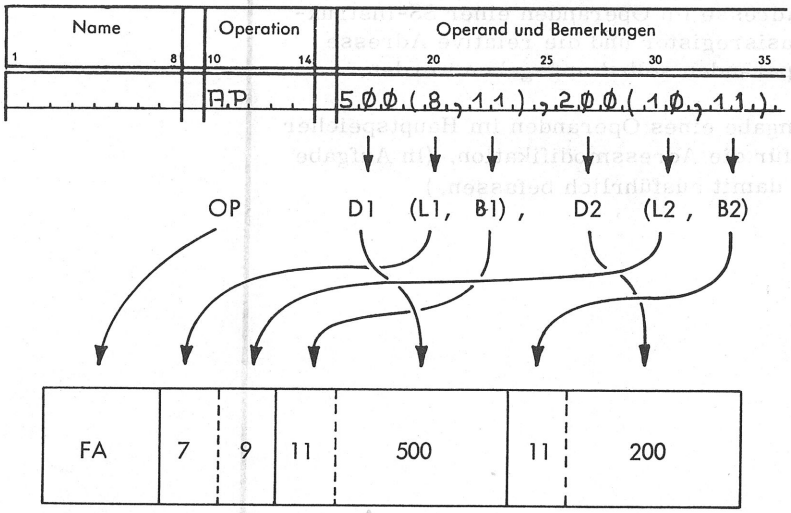


Abb. 2

Die explizite Adressierung der Operanden einer SS-Instruktion

2 Betrachten Sie bitte Abb. 2.-

Die in Abb. 2 dargestellte SS-Instruktion AP hat 2 Operanden, die durch ein Komma voneinander getrennt sind. Jeder der Operanden besitzt 2 Unterfelder, die in Klammern eingeschlossen und wiederum durch Kommas voneinander getrennt sind.

Der 1. Operand der Instruktion in Abb. 2 lautet _____, der 2. Operand _____.

500(8, 11)

200(10, 11)

3 Wir betrachten nun den 1. Operanden der Instruktion in Abb. 2 im einzelnen.

500(8, 11)

Der Direktwert vor der Klammer gibt die ____(WW)____ des 1. Operanden an.

relative Adresse

4 Die beiden Unterfelder des 1. Operanden sind in _____ gesetzt.

Das erste Unterfeld (8) gibt die _____ des 1. Operanden an. Das zweite Unterfeld (11) bezeichnet das _____, das die Basisadresse für die Adreßbildung des 1. Operanden enthält.

Klammern

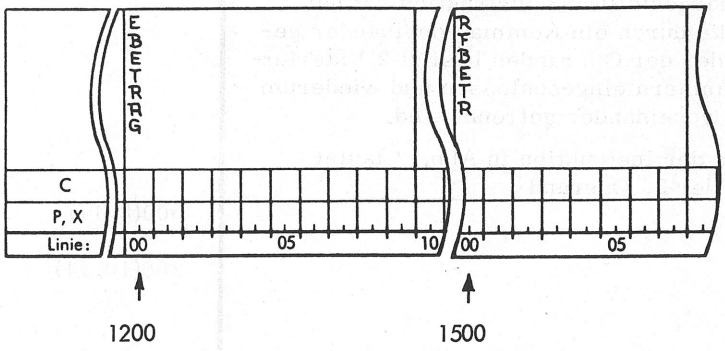
Länge

Basisregister

5 Wie Sie bereits gesehen haben, müssen Unterfelder in Klammern gesetzt werden.

Die Trennung von Unterfeldern innerhalb einer Klammer muß durch ein _____ erfolgen.

Komma



Basisregister: 11 Basisadresse: 1000

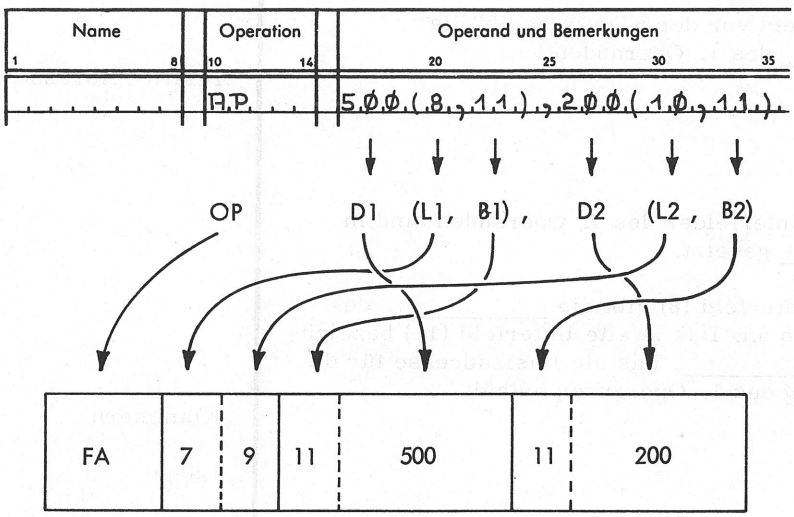


Abb. 2

6 In dem 2. Operanden der SS-Instruktion in Abb. 2 sind folgende Angaben für die Adresse und Länge eines Feldes im Hauptspeicher gemacht:

a) die relative Adresse __(Z)__,

b) die Länge __(Z)__,

c) das Basisregister __(Z)__.

200

10

11

7 Um die Adresse und die Länge eines Feldes im Hauptspeicher explizit im Operanden einer SS-Instruktion festlegen zu können, müssen also Angaben gemacht werden

a) zur ___(WW)___,

b) zur _____,

c) zum _____.

relativen Adresse

Länge

Basisregister

8 0(5, 9)

Was ist in diesem Operanden im einzelnen angegeben ?

a) Die ___(WW)___ __(Z)__,

b) die ___(WZ)___,

c) das ___(WZ)___.

relative Adresse 0

Länge 5

Basisregister 9

9 In den Operanden einer PACK-Instruktion sollen Adresse und Länge explizit angegeben werden.

Der 1. Operand soll die Länge 6, die relative Adresse 10 und das Basisregister 8 angeben, der 2. Operand die Länge 10, die relative Adresse 45 und das Basisregister 8.

Wie muß die Instruktion lauten ?

PACK 1Ø(6,8),45(1Ø,8).

10 Wenn die relative Adresse und das Basisregister, in dem die Basisadresse enthalten ist, sowie die Länge explizit in den Operanden einer SS-Instruktion angegeben sind, dann entfallen für diese Operanden die DS-Anweisungen, und die USING-Anweisung hat keinen Einfluß auf diese Operanden. In solchen Operanden muß also auch immer die Länge explizit angegeben werden, da diese nicht mehr aus einer DS-Anweisung entnommen werden kann.

Alles was der Assembler für die Umwandlung einer solchen Instruktion in ihre echte Form benötigt, ist in den Operanden selbst explizit angegeben:

PACK 1Ø(6,8),45(1Ø,8).

Wie sieht diese PACK-Instruktion nach der Umwandlung in ihrer echten Form aus ? (Die echte Form des OP-Schlüssels PACK lautet F2).

F2	5	9	8	10	8	45

(Beachten Sie bitte, daß in der echten Instruktion die Länge um 1 kleiner ist!)

DIE MÖGLICHEN OPERANDEN-FORMATE FÜR INSTRUKTIONEN IM SS-FORMAT

Arithmetische Operationen

1. Operand:	2. Operand:
S1	S2
S1(L1)	S2(L2)
D1(L1,B1)	D2(L2,B2)

Anmerkung:

S bedeutet die symbolische Adresse (ein einfacher oder zusammengesetzter Ausdruck) eines Feldes im Hauptspeicher.
L bezeichnet die explizite Länge eines Feldes im Hauptspeicher.
D bedeutet die relative Adresse eines Feldes im Hauptspeicher, explizit angegeben.
B bezeichnet ein Basisregister, das die Basisadresse enthält.

Abb. 3

Im folgenden fassen wir nun noch einmal alle Möglichkeiten der Angabe eines Operanden in einer SS-Instruktion systematisch zusammen.

11 Betrachten Sie bitte Abb. 3. -

In der Form S1 bzw. S2 wird eine symbolische Adresse im 1. bzw. 2. Operanden angegeben. Diese symbolische Adresse kann ein _____ oder _____ Ausdruck sein (z. B. EFELD oder EFELD+5).

einfacher

zusammengesetzter

12 In der Form S1(L1) bzw. S2(L2) wird ein Feld im Hauptspeicher ebenfalls symbolisch adressiert, jedoch wird die _____ des Feldes _____ angegeben.

Länge

explizit

13 In der Form D1(L1, B1) bzw. D2(L2, B2) wird in den Operanden alles explizit angegeben, sowohl die Adresse, nämlich durch die ____(WW)____ und das _____, als auch die _____ des Feldes im Hauptspeicher.

relative Adresse

Basisregister

Länge

14 Für die beiden Operanden einer SS-Instruktion brauchen nicht die sich entsprechenden Formen gewählt werden. Die folgende Instruktion ist durchaus erlaubt:

Name (1–8)	Operation (10–14)	Operand und Bemerkung (20–25)
	PACK	REFE,Ø(7,11)

In dieser arithmetischen Instruktion im SS-Format ist der 1. Operand in der Form _____ (siehe Abb. 3), der 2. Operand in der Form _____ geschrieben.

S1

D2(L2, B2)

DIE MÖGLICHEN OPERANDEN-FORMATE FÜR
INSTRUKTIONEN IM SS-FORMAT

Arithmetische Operationen

1. Operand:	2. Operand:
S1	S2
S1(L1)	S2(L2)
D1(L1,B1)	D2(L2,B2)

Logische Operationen

1. Operand:	2. Operand:
S1	S2
S1(L)	-
D1(L,B1)	D2(B2)

Anmerkung:

S bedeutet die symbolische Adresse (ein einfacher oder zusammengesetzter Ausdruck) eines Feldes im Hauptspeicher.
L bezeichnet die explizite Länge eines Feldes im Hauptspeicher.
D bedeutet die relative Adresse eines Feldes im Hauptspeicher, explizit angegeben.
B bezeichnet ein Basisregister, das die Basisadresse enthält.

Abb. 4

15 0(7, 11)

Was ist in diesem Operanden von links beginnend im einzelnen angegeben ?

Relative Adresse 0

Länge 7

Basisregister 11

16 Betrachten Sie bitte Abb. 4. -

Die seitherige Abbildung wurde um die Formate der Operanden bei logischen SS-Instruktionen erweitert.

Arithmetische und logische Operationen im SS-Format unterscheiden sich lediglich bezüglich der Längenangaben. Bei arithmetischen Operationen gibt es zwei Längenangaben (für den 1. und 2. Operanden). Bei logischen Operationen dieses Formates gibt es nur e i n e Längenangabe.

Deshalb ist in Abb. 4 statt L1 bei den arithmetischen nur L bei den logischen Operationen angegeben.
Bei diesen Operationen ist nur eine Längenangabe explizit möglich, nämlich die des (1., 2.) _____ Operanden.

1.

17 Für die Angabe des 2. Operanden einer logischen Operation im SS-Format gibt es nur 2 Möglichkeiten, da hier die Längenangabe entfällt.

Der 2. Operand einer logischen SS-Instruktion kann in der Form _____ oder _____ geschrieben werden (siehe Abb. 4).

S2

D2(B2)

18

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 3
.....	M.V.C. .	A.B.E.R.(.5.).,1.5.(.8.). .

Welchen Operandenformaten entsprechen die beiden Operanden dieser logischen SS-Instruktion?

Der Operand ABER(5) entspricht dem Format S1(L).

Der Operand 15(8) entspricht dem Format D2(B2).

DIE MÖGLICHEN OPERANDEN-FORMATE FÜR INSTRUKTIONEN IM SS-FORMAT

Arithmetische Operationen

1. Operand:	2. Operand:
S1	S2
S1(L1)	S2(L2)
D1(L1,B1)	D2(L2,B2)

Logische Operationen

1. Operand:	2. Operand:
S1	S2
S1(L)	"
D1(L,B1)	D2(B2)

Anmerkung:

S bedeutet die symbolische Adresse (ein einfacher oder zusammengesetzter Ausdruck) eines Feldes im Hauptspeicher.
L bezeichnet die explizite Länge eines Feldes im Hauptspeicher.
D bedeutet die relative Adresse eines Feldes im Hauptspeicher, explizit angegeben.
B bezeichnet ein Basisregister, das die Basisadresse enthält.

Abb. 4

19

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
. . . . .	M.V.C. .	8.Ø.(5.,8.),4.5.(8.) . . .

Geben Sie bitte auf einem Extrablatt an, wie diese Instruktion (SS-Format) nach der Umwandlung aussieht. (Der Operationsschlüssel für MVC lautet D2.)

D2	4	8	80	8	45

20 Wir fassen zusammen:

Für das Schreiben der Operanden in a r i t h m e t i s c h e n SS-Instruktionen gibt es 3 Möglichkeiten:

a) Angabe einer symbolischen Adresse, Länge implizit. - Format S1 bzw. S2.

b) Angabe einer symbolischen Adresse, Länge explizit. - Format _____ bzw. _____.

c) Angabe einer expliziten Adresse und einer expliziten Länge. - Format _____ bzw. _____.

S1(L1) (bzw) S2(L2)

D1(L1,B1) (bzw.) D2(L2,B2)

21 Bei l o g i s c h e n SS-Instruktionen müssen wir wegen der Längenangabe, die nur beim 1. Operanden möglich ist, unterscheiden:

Der 1. Operand kann geschrieben werden

a) als symbolische Adresse, Länge implizit, - Format _____,

b) als symbolische Adresse, Länge explizit, - Format _____,

c) als explizite Adresse, Länge ebenfalls explizit, - Format _____.

Der 2. Operand kann geschrieben werden

a) als symbolische Adresse, - Format _____,

b) als explizite Adresse, - Format _____.

S1

S1(L)

D1(L, B1)

S2

D2(B2)

Auch bei den Instruktionen aller anderen Formate können die Operanden im Hauptspeicher explizit angegeben werden. Weiteres hierzu erfahren Sie im Folgelehrgang.

FRAGEN ZUM KAPITEL VI

Beantworten Sie bitte erst alle Fragen zum Kapitel VI auf einem Extrablatt, ehe Sie Ihre Antworten überprüfen.

1. Ein Feld im Hauptspeicher wird in den Operanden einer SS-Instruktion in der Regel symbolisch adressiert, nämlich mittels eines _____ oder _____ Ausdruckes.

2. Wird neben einer symbolischen Adresse eine explizite Länge angegeben, so wird die Längenangabe als Unterfeld in _____ gesetzt.

3. Die Beschreibung der nachfolgenden Operanden ist durcheinander geraten. - Ordnen Sie bitte jedem der Operanden die richtige Beschreibung zu. Geben Sie nur die Buchstaben an.

A	ABER-3	-	A einfacher Ausdruck, Länge implizit.
B	ENR(3)	-	B zusammengesetzter Ausdruck, Länge explizit.
C	EBER	-	C einfacher Ausdruck, Länge explizit.
D	EBER+5(5)	-	D zusammengesetzter Ausdruck, Länge implizit.

4. Bei der expliziten Adressierung müssen in den Operanden einer arithmetischen SS-Instruktion explizit angegeben werden:

   a) die ___(WW)___ und

   b) als Unterfelder die _____ und das zu benutzende _____.

5. Wie müssen die Operanden geschrieben werden, in denen explizit die folgenden Angaben gemacht werden sollen:

   a) Basisregister 9, relative Adresse 25, Länge 5.

   b) relative Adresse 0, Länge 10, Basisregister 15.

6. Da im 2. Operanden einer logischen SS-Instruktion keine explizite Längenangabe gemacht werden darf, werden hier bei expliziter Adressierung nur die __(WW)__ und das _____ angegeben.

7. Wie muß ein solcher Operand lauten, wenn die relative Adresse 0 und das Basisregister 15 explizit angegeben werden sollen ?

ANTWORTEN ZUM KAPITEL VI

Berichtigen Sie eventuell aufgetretene Fehler und beseitigen Sie Unklarheiten ! Wiederholen Sie gegebenenfalls die in Klammern angegebenen Seiten bzw. Lernelemente, soweit Ihre Antworten falsch sind.

1. Ein Feld im Hauptspeicher wird in den Operanden einer SS-Instruktion in der Regel symbolisch adressiert, nämlich mittels eines einfachen oder zusammengesetzten Ausdruckes.

   (Seite 309)

2. Wird neben einer symbolischen Adresse eine explizite Länge angegeben, so wird die Längenangabe als Unterfeld in Klammern gesetzt.

   (Seite 309)

3. A - D

   B - C

   C - A

   D - B

   (Seite 309)

4. a) die relative Adresse und

   b) als Unterfelder die Länge und das zu benutzende Basisregister.

   (LE 2 - 10)

5. a) 25(5, 9)

   b) 0(10, 15)

(LE 2 - 10)

6. Da im 2. Operanden einer logischen SS-Instruktion keine explizite Längenangabe gemacht werden darf, werden hier bei expliziter Adressierung nur die relative Adresse und das Basisregister angegeben.

(LE 16 - 18)

7. 0(15)

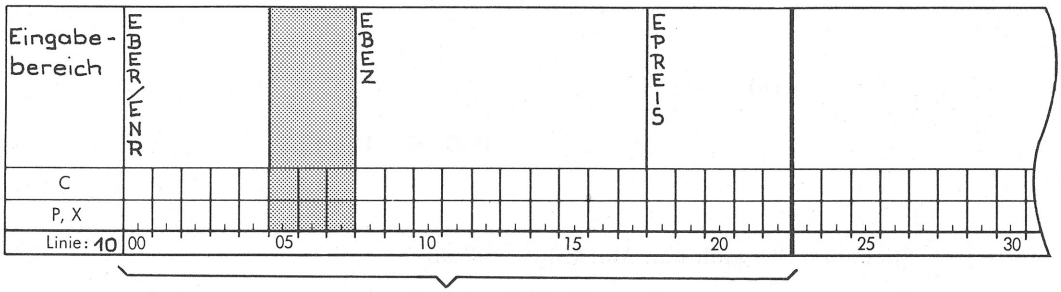
(LE 16 - 18)

Schauen Sie sich nun Merkblatt 307 an. -



Band 1, Seite 439

IBM System / 360
Speicherbelegung



Eingabebereich

Abb. 1

FRAGEN ZUM BAND 3

Die folgenden Fragen beziehen sich auf den Lernstoff des Bandes 3. - Schreiben Sie Ihre Antworten bitte auf ein Extrablatt bzw. Programmschema.

Wenn Sie alle Fragen beantwortet haben, finden Sie die Lösungen auf der nächsten Seite.

1. Das in der Assemblersprache geschriebene Programm ist ein _____ Programm, das vor der Verarbeitung der Daten in ein _____ umgewandelt werden muß.

2. Die Umwandlung des symbolischen Programmes in das Maschinenprogramm erfolgt im System /360 und wird durch das _____ gesteuert.

3. Den Maschineninstruktionen, Konstanten, Feldern und Bereichen des symbolischen Programmes werden bei der Umwandlung mit Hilfe des _____ nacheinanderfolgende Hauptspeicherstellen zugewiesen.

4.

Name	Operation	Opera
1 8	10 14	20
. . . . .	S.T.A.R.T	1.Ø.Ø.Ø. . .

Durch diese Anweisung wird der Zuordnungszähler auf die ___(WZ)_____ eingestellt.

5. Entnehmen Sie bitte Ihren Arbeitsmitteln ein Programmschema und definieren Sie den in Abb. 1 dargestellten Eingabebereich.

IBM System / 360
Speicherbelegung

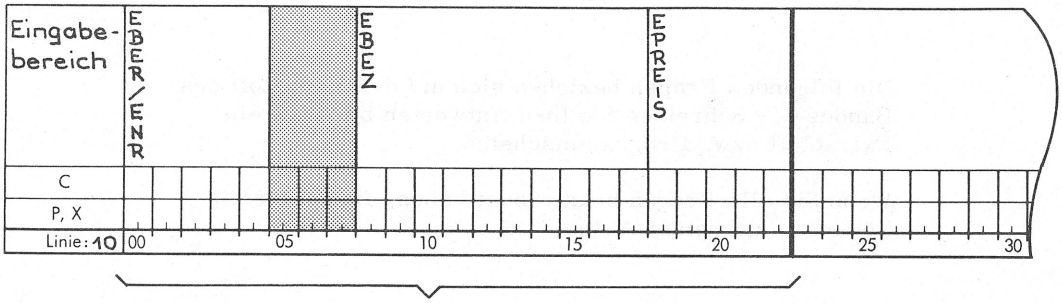


Abb. 1

IBM System / 360
Speicherbelegung

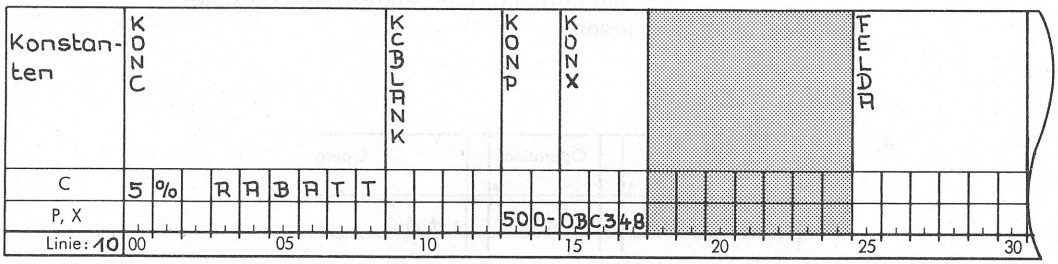


Abb. 2

6. Anschließend an die Definitionen sollen Sie nun mittels einer ORG-Anweisung dafür sorgen, daß der Zuordnungszähler auf die Adresse 1030 verweist (also weitergestellt wird). Siehe Abb. 1. - Es gibt mehrere Möglichkeiten.

7. Wie müssen die Operanden lauten, die die folgenden Speicherstellen (siehe Speicherbelegung Abb. 1) ansprechen ? Verwenden Sie die in Klammern angegebenen Symbole.

a) 1000 - 1007 (EBER)

b) 1005 - 1007 (EBEZ)

c) 1017 (EBEZ)

8. Welche Konstantentypen werden durch die folgenden Typenschlüssel angegeben ?

C - _____

P - _____

X - _____

F - _____

H - _____

A - _____

9. Entnehmen Sie bitte Ihren Arbeitsmitteln ein Programmschema. -

a) Definieren Sie bitte die in Abb. 2 dargestellten Konstanten. Benutzen Sie die in den Namen angedeuteten Konstantentypen.

b) Die Adresse von FELDA (Abb. 2) soll als Adreßkonstante (ADKON) definiert werden.

10.

Name	Operation	Oper
1 8	10 14	20
. . . . .	U.S.I.N.G	*,1.5. . .

a) Was wird dem Assembler durch diese USING-Anweisung mitgeteilt ?

b) Der augenblickliche Wert des Zuordnungszählers sei 2000. Welche Hauptspeicheradressen können dann mit Hilfe der Basisadresse und der relativen Adressen angesprochen werden ?

ANTWORTEN ZUM BAND 3

Wenn Ihnen Fehler unterlaufen sind, sollten Sie zu den Fragen des betreffenden Kapitels (in Klammern) zurückschlagen und diese durcharbeiten. Kommt es auch dort zu Fehlern, dann wiederholen Sie die betreffenden Lernelemente. - Fahren Sie erst dann in diesem Lehrgang fort, wenn Sie sicher sind, daß Sie den Lernstoff dieses Bandes beherrschen.

1. Das in der Assemblersprache geschriebene Programm ist ein symbolisches Programm, das vor der Verarbeitung der Daten in ein Maschinenprogramm umgewandelt werden muß.

   (Fragen zum Kapitel I, Seite 43)

2. Die Umwandlung des symbolischen Programmes in das Maschinenprogramm erfolgt im System /360 und wird durch das Assembler-Umwandlungsprogramm gesteuert.

   (Fragen zum Kapitel I, Seite 43)

3. Den Maschineninstruktionen, Konstanten, Feldern und Bereichen des symbolischen Programmes werden bei der Umwandlung mit Hilfe des Zuordnungszählers nacheinanderfolgende Hauptspeicherstellen zugewiesen.

   (Fragen zum Kapitel III, Seite 177-179)

4. Durch diese Anweisung wird der Zuordnungszähler auf die Anfangsadresse 1000 eingestellt.

   (Fragen zum Kapitel III, Seite 177-179)

5.

Name 1 8	Operation 10 14	Operanc 20
E.B.E.R. . . .	D.S. . .	Ø.C.L.2.3. . .
E.N.R.	D.S. . .	C.L.5. . . . .
. . . . .	D.S. . .	C.L.3. . . . .
E.B.E.Z. . . .	D.S. . .	C.L.1.Ø. . . .
E.P.R.E.I.S. .	D.S. . .	C.L.5. . . . .

(Fragen zum Kapitel III, Seite 177-179)

6.

Name 1 8	Operation 10 14	Operand und Be 20 25
:	: . . .	. . . . .
. . . . .	O.R.G. .	E.B.E.R.+.3.Ø. . . .
oder : . . .	O.R.G. .	E.N.R.+3Ø. . . . .
oder : . . .	O.R.G. .	E.B.E.Z.+.2.2. . . .
oder : . . .	O.R.G. .	E.P.R.E.I.S.+.1.2. .
oder : . . .	O.R.G. .	*+7 . . . . .

(Fragen zum Kapitel III, Seite 177-179)

7. a) EBER(8)
   b) EBEZ-3(3)
   c) EBEZ+9(1)

(Fragen zum Kapitel III, Seite 177-179)

8. C - Zeichenkonstante
   P - Gepackte Konstante
   X - Hexadezimalkonstante
   F - Wortkonstante
   H - Halbwortkonstante
   A - Adreßkonstante

(Fragen zum Kapitel IV, Seite 265-267)

9.

	Name (1 – 8)	Operation (10 – 14)	Operand und Bemerku (20 – 25)
a)	K,O,N,C, . . .	D,C, . . .	C,',5,%, ,R,A,B,A,T,T,',
	K,C,B,L,A,N,K,	D,C, . . .	C,L,4,', ', ...
	oder:		
	K,C,B,L,A,N,K,	D,C, . . .	C,', . . . ', ...
	K,O,N,P, . . .	D,C, . . .	P,' - 5,Ø,Ø,' . . . . .
	K,O,N,X, . . .	D,C, . . .	X,' B,C,3,4,8,' . . . . .
	. . . . .	. . . . .	. . . . .
b)	A,D,K,O,N, . .	D,C, . . .	A,(,F,E,L,D,A,), ...

(Fragen zum Kapitel IV, Seite 265-267)

10. a) Sinngemäß:

Dem Assembler wird mitgeteilt, daß Register 15 als Basisregister und der momentane Wert des Zuordnungszählers als Basisadresse für die Adreßbildung benutzt werden sollen.

b) Es können die Hauptspeicheradressen 2000 bis 6095 angesprochen werden (relative Adressen 0 bis 4095).

(Fragen zum Kapitel V, Seite 303)