

**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

Programmierte in der IBM 360 Assembler-Sprache
(AAS-PU)

Eine Programmiersprache

Text 2

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 Instruk- tionen 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

Auftrags-Nr.				Lohn- oder Materialkosten				K A						
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

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 klar erkannt und fest umrissen sein.

ERSTELLEN EINER LISTE DER LOHN- UND MATERIALKOSTEN

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

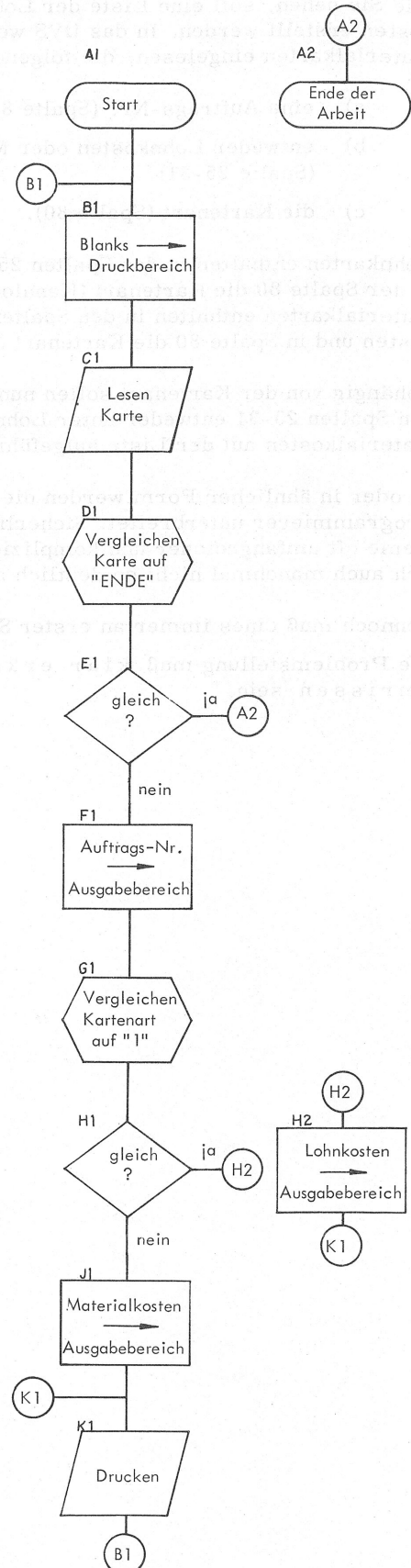
Auftrags-Nr.	Lohnkosten	Materialkosten
00513	943,50	2.051,03
00513		758,81
00513	1.236,39	
01024	1.098,75	382,56
01024	885,86	1.340,19
01024	1.125,70	
03152		826,63
03152		92,84
03152	974,23	
03604	4.247,14	10.387,36
03604		

Druckstellen

4 - 8

11 - 19

22 - 30



- 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.

- 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. -

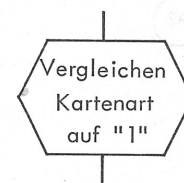
Wie nennt man diese Blöcke (Symbole der Zeichenschablone)?

Anmerkung:

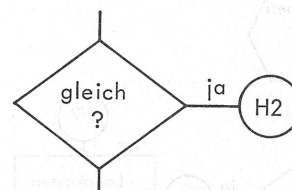
Die Symbole des Programmablaufplans sind im Merkblatt 301 in Ihren Arbeitsmitteln zusammengestellt.

Programmablaufplan

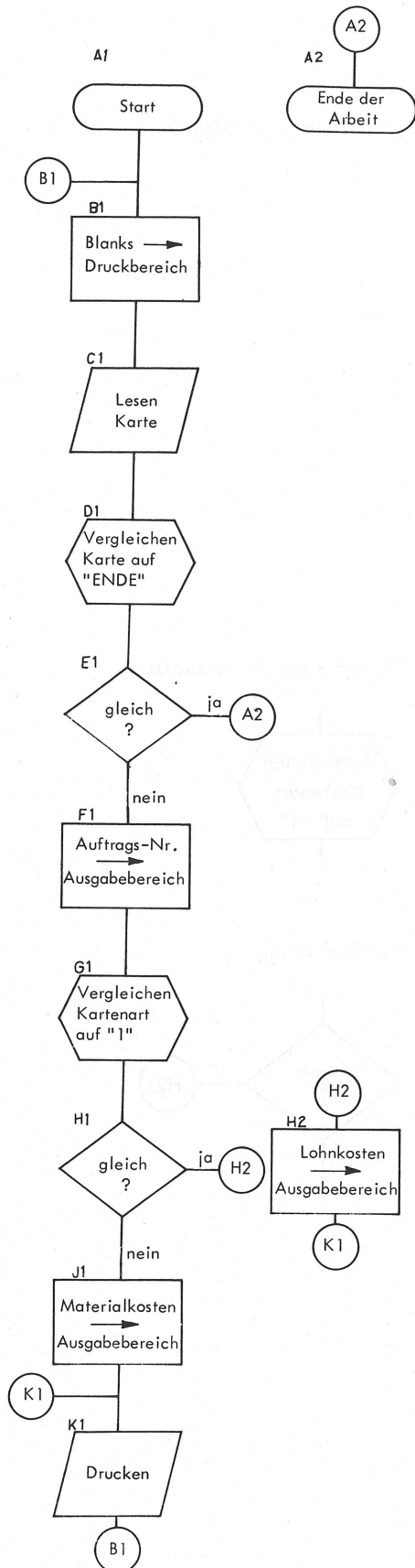
Programm-Modifikation



Entscheidung



PROGRAMMABLAUFPLAN



PROGRAMM

Name	Operation	Operand und Bemerkungen
1	8	10 14 20 25 30
	START	1,000,00
ANFANG	BALR	8,0
	USING	*,8
* PROGRAMMABLAUF		
B1	MVC	ABER,KCBLANK
	GET	KARTE,EBER
	CLC	EBER(4),KCENDE
	BE	R2
	MVC	RNR,ENR
	CLC	EKR,KC1
	BE	H2
	MVC	EMRT-1,(10),SCHABL
	PRCK	EMRT,EMRT
	ED	EMRT-1,(10),EMRT+3
K1	PUT	LISTE,ABER
	B	B1
H2	MVC	ELOHN-1,(10),SCHABL
	PRCK	ELOHN,ELOHN
	ED	ELOHN-1,(10),ELOHN+3
	B	K1
R2		Ende der Arbeit
* DEFINITION		

- 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 symbolischen _____ 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

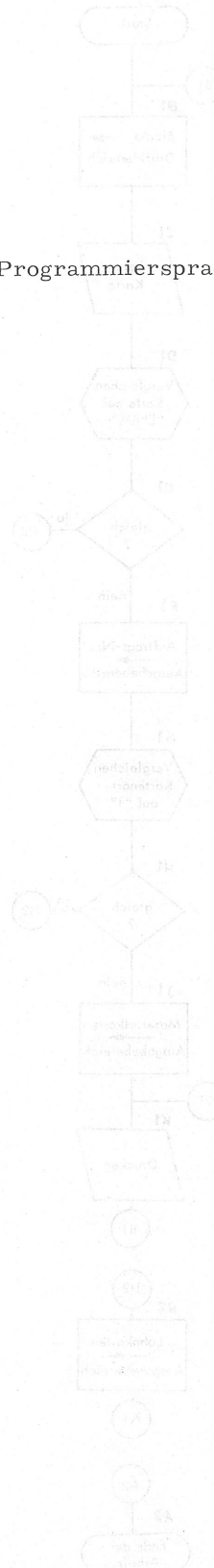
- 8 Ein Programm, das in einer symbolischen Programmiersprache geschrieben ist, bezeichnet man als ein _____ Programm.

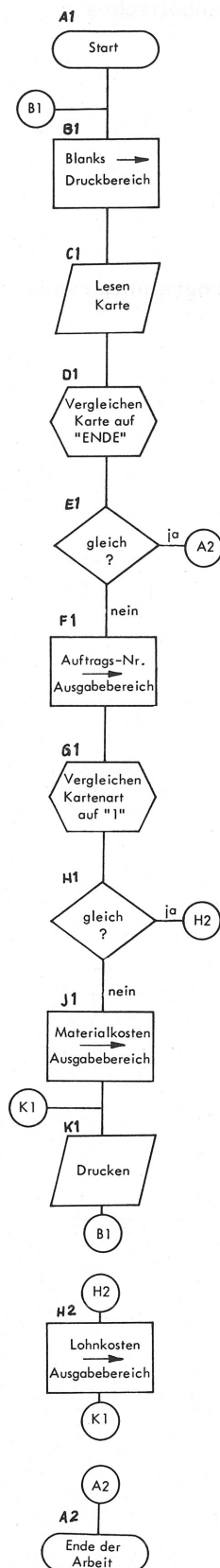
- 9 In diesem Lehrgang befassen wir uns mit dem Schreiben symbolischer Programme in der Assemblersprache.

Die Assemblersprache ist eine symbolische _____.

symbolisches

Programmiersprache





Name	Operation	Operand und Bemerkungen
1	8	10 14 20 25 30 35
	START	1000

B1. MVC. RBER, KCBLANK.

GET. KARTE, EBER

CLC. EBER(4), KCENDE

BE. R2

MVC. RNR, ENR

CLC. EKR, KC1

BE. H2

MVC. RMBT-1(10), SCHBBL
PRCK. EMAT, EMAT
ED. RMBT-1(10), EMAT+3

K1. PUT. LISTE, RBER

B. B1

H2. MVC. RLOHN-1(10), SCHBBL
PRCK. ELOHN, ELOHN
ED. RLOHN-1(10), ELOHN+3

B. K1

R2. Ende der Arbeit

- 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 Codieren des Programmablaufplans.

Die Codierung 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 Anweisungen in der Assemblersprache übertragen wird.

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

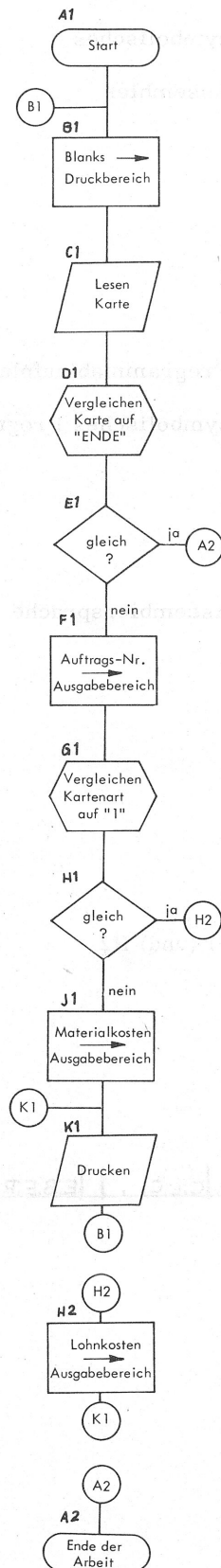
J1 (und) H2

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

DIE CODIERUNG DES PROGRAMMABLAUPLANS IN DER ASSEMBLERSPRACHE



Name	Operation	Operand und Bemerkungen
1	8	10 14 20 25 30 35
	START	10000

B1 M.V.C. RBER, KCBLANK

GET KRITE, EBER

CLC EBER(4), KCENDE

BE R2

M.V.C. RNR, ENR

CLC EKR, KC1

BE H2

M.V.C. RMBT-1(10), SCHABL
PACK EMAT, EMAT
ED RMBT-1(10), EMAT+3

K1 PUT LISTE, RBER

B B1

H2 M.V.C. RLOHN-1(10), SCHABL
PACK ELOHN, ELOHN
ED RLOHN-1(10), ELOHN+3

B K1

R2 Ende der Arbeit

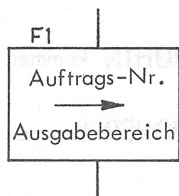
- 15 Eine Anweisung, in der der Programmierer sagt, was und mit welchen Daten etwas getan werden soll, bezeichnet man auch als eine Instruktion.

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

- 16 Abb. 4 zeigt:

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

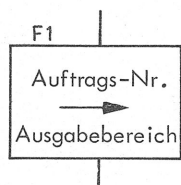
- 17



Die Eintragung im Block F1 bedeutet:

"Übertrage _____ nach _____".

- 18



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

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) _____.

Instruktion

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

Auftrags-Nr.

Ausgabebereich

MVC, ANR (und) ENR

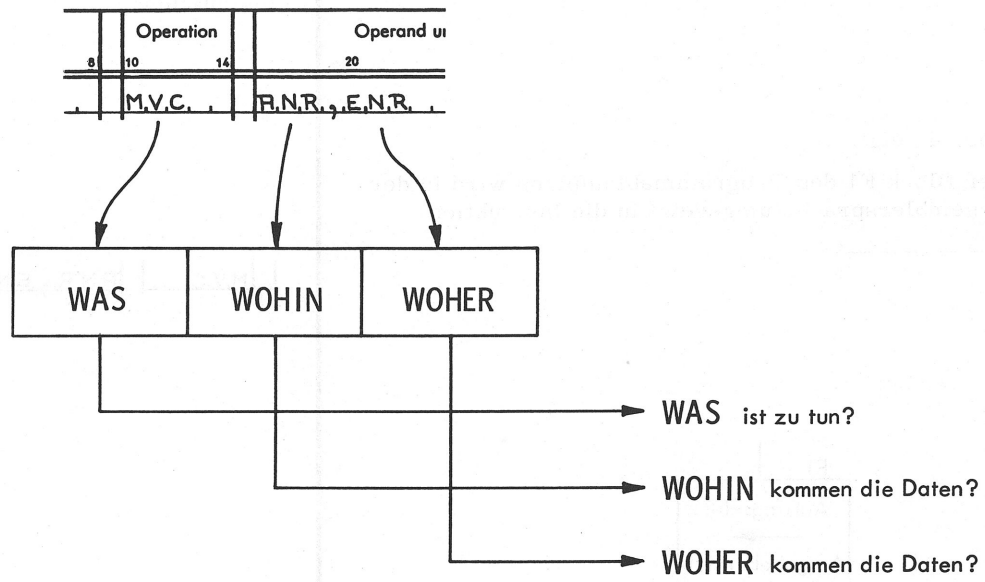


Abb. 5

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

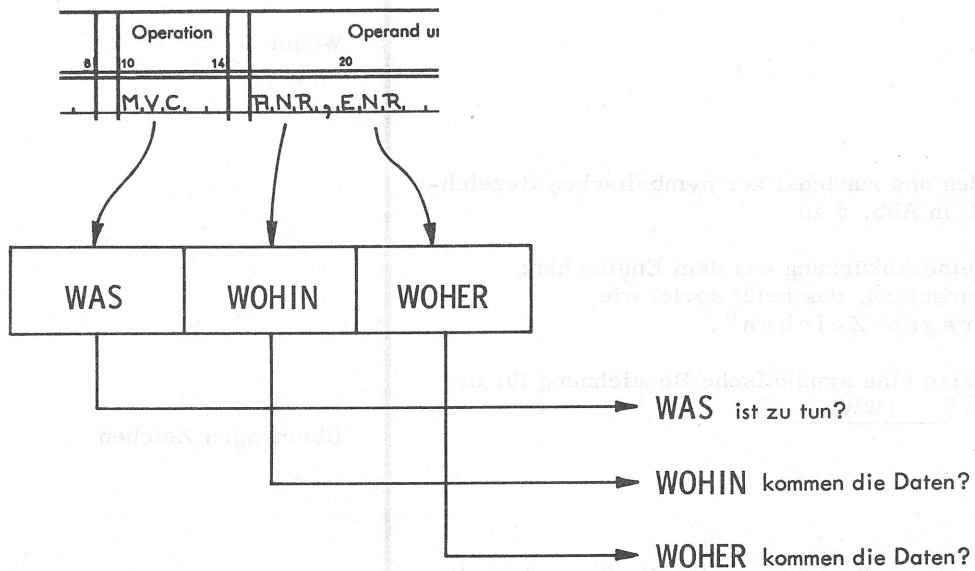


Abb. 5

- 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

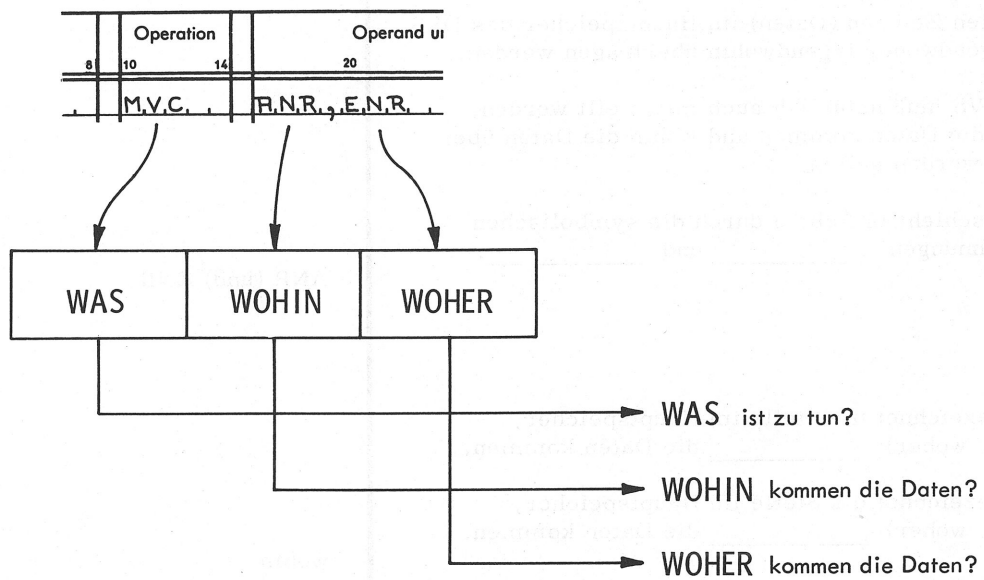


Abb. 5

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 eine Sprache, nämlich die Sprache, die der Konzeption des Systems entspricht, die Maschinensprache.

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 umgewandelt werden, damit die Instruktionen vom DVS verstanden werden können. Ein Programm, das in die Maschinensprache umgewandelt ist, bezeichnet man als Maschinenprogramm.

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

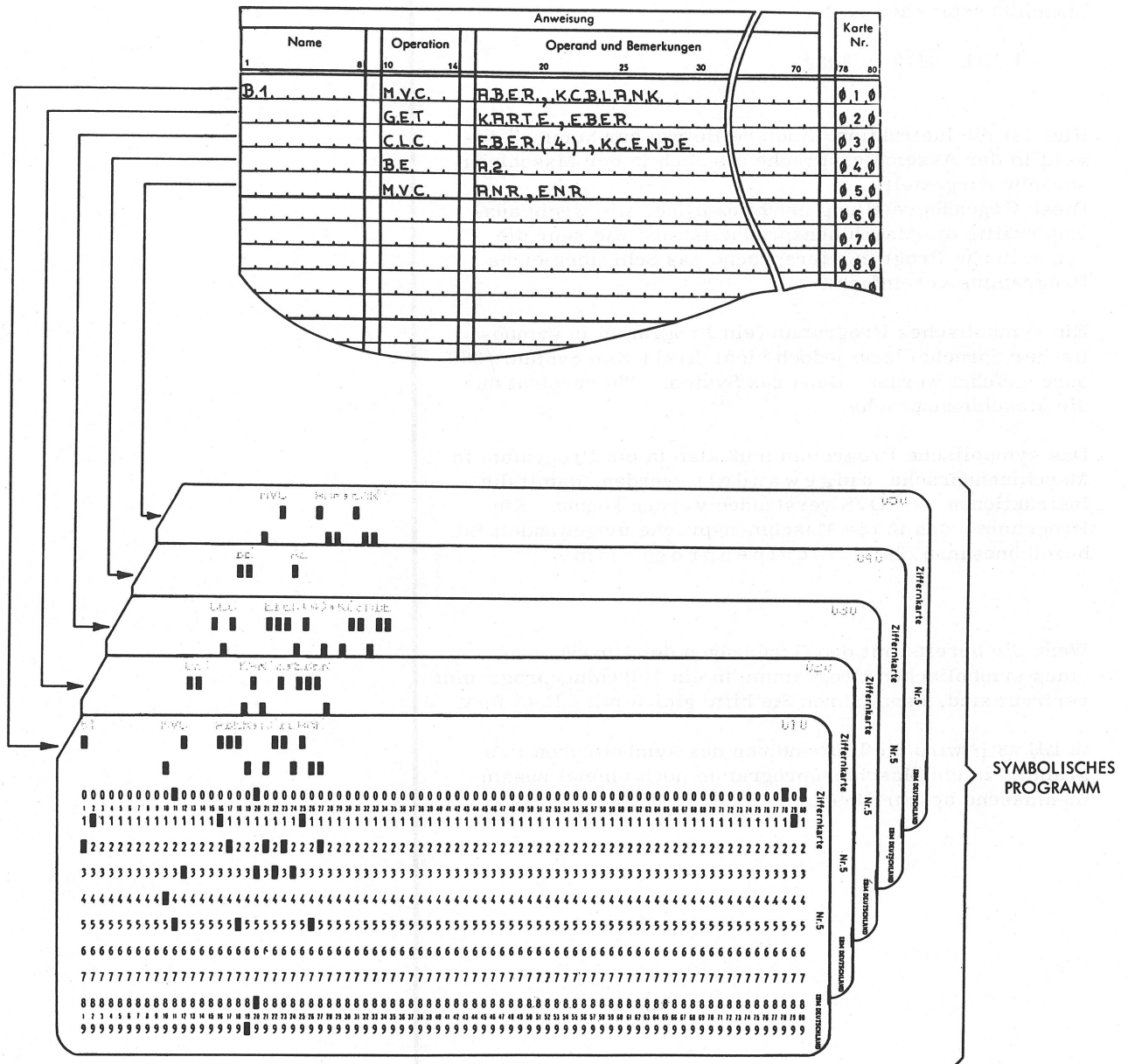


Abb. 6

- 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 Maschinenprogramm.

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 Eine 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

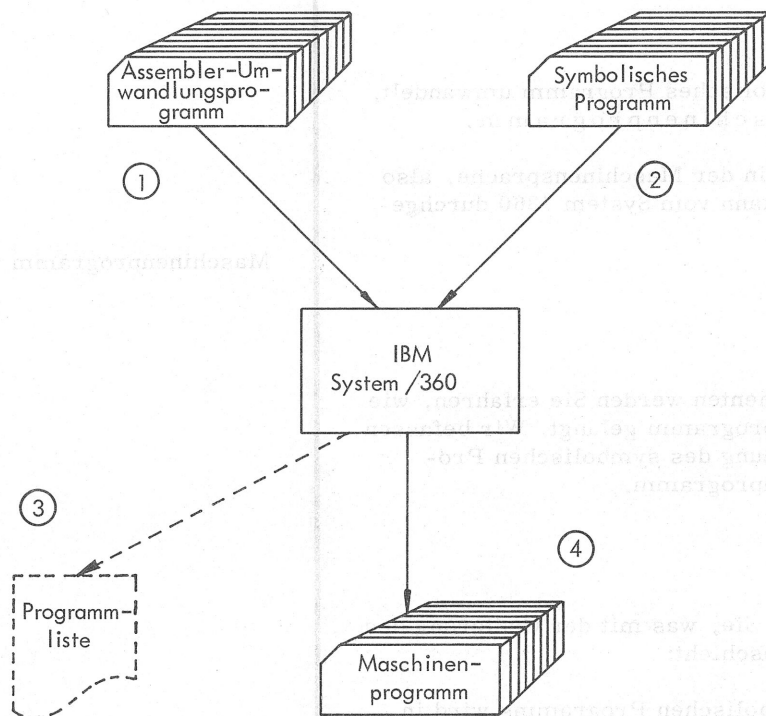


Abb. 7

- 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

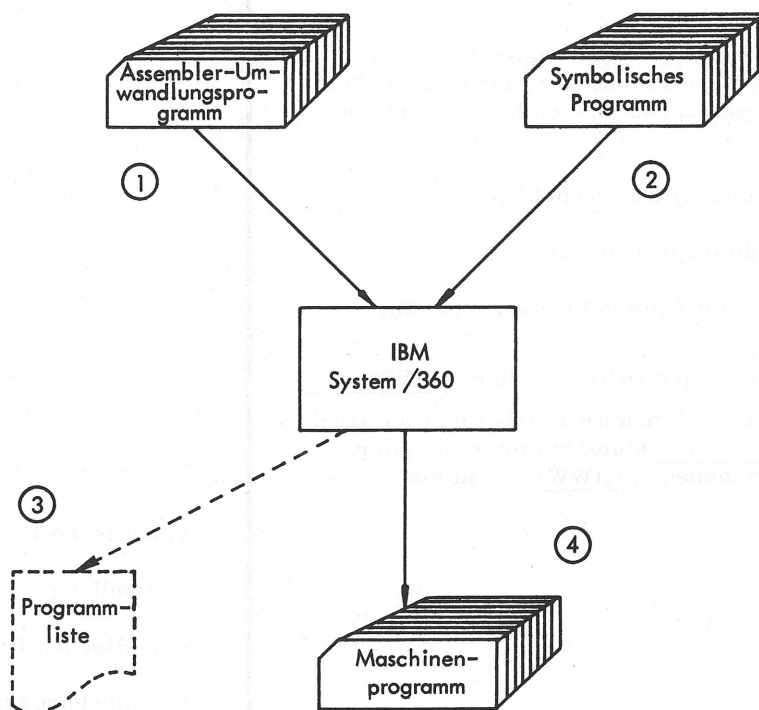


Abb. 7

- 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 Maschinsprache.

- 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

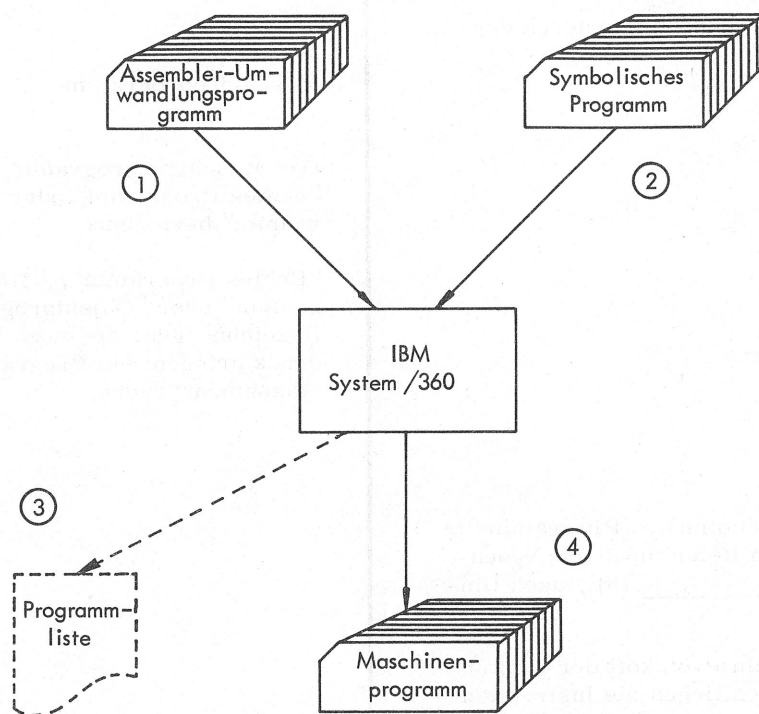


Abb. 7

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

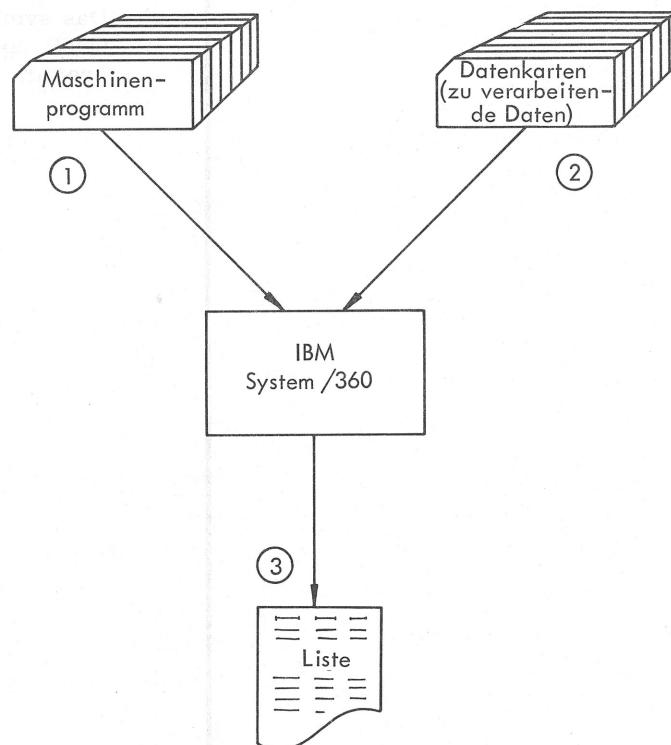


Abb. 8

- 39 Wir wenden uns jetzt der Verarbeitung 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

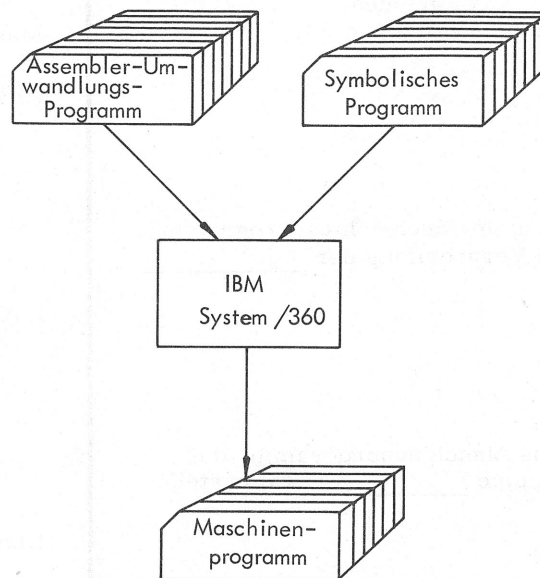
- 42 Beantworten Sie bitte erst beide Fragen, bevor Sie Ihre Antworten überprüfen.

- a) Welches Programm bewirkt die Verarbeitung der Daten?
- b) Was geschieht mit dem Maschinenprogramm, bevor Daten verarbeitet werden?

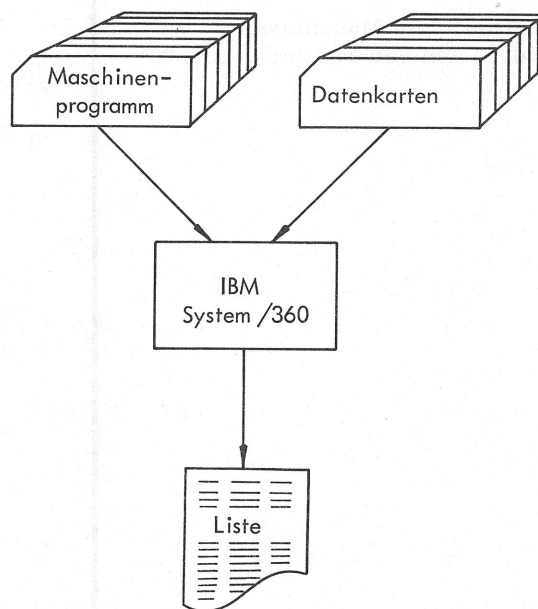
a) Das Maschinenprogramm

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

I UMWANDLUNG



II VERARBEITUNG



- 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

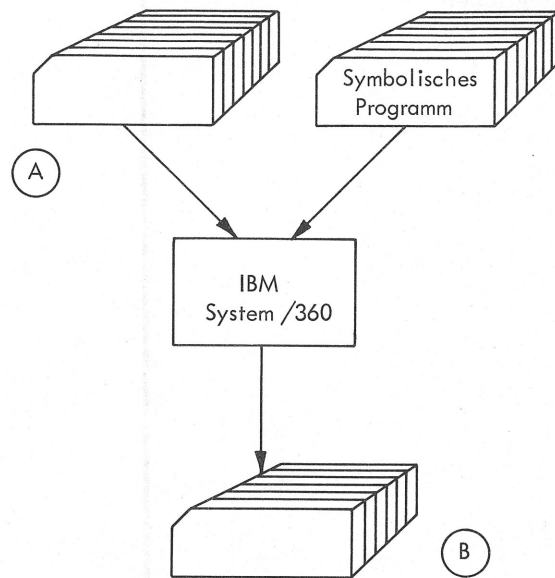
- 47 Ist das Maschinenprogramm durch die Umwandlung einmal erstellt, kann es immer wieder für die Verarbeitung der Daten verwendet werden.

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

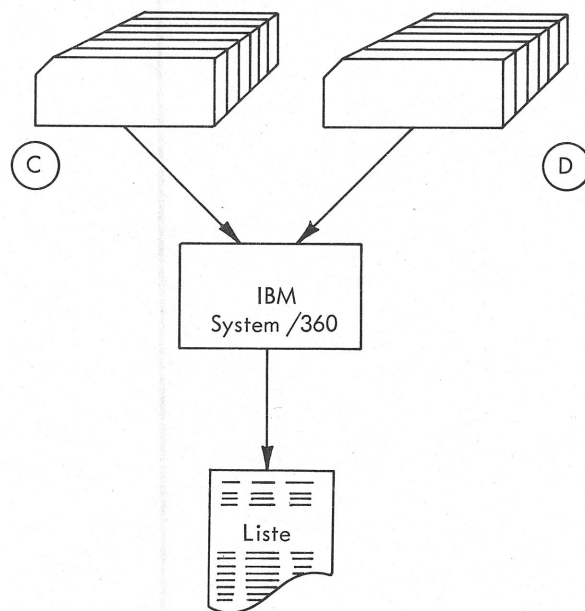
Hauptspeicher

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



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 Assembler-sprache. Wenn man ein Programm in der Assembler-sprache schreibt, dann müssen einige formale Regeln 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 Programmschema.

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 durchnummeriert.

Die einzelnen Stellen einer Zeile entsprechen den _____ einer Lochkarte.

Spalten

- 2 Pro Zeile ist nur eine Anweisung erlaubt, jedoch dürfen nach Belieben Zeilen freigelassen werden.

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

1 Lochkarte

- 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

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		Operation		Operand und Bemerkungen		
	1	8	10	14	20	25	30
a)			C.L.C.		ENR.,K.O.N.S.T.		
b)			P.R.C.K.		BETR.R.G.,BETR.R.G.		
c)			P.R.C.K.KO		STEN.,K.O.S.T.E.N.		
d)			R.P.		S.U.M.M.E.,BETR.R.G.		
e)			R.P.		G.E.K.O.S.T.E.N.,K.O.S.T.E.N.		
f)			M.V.C.		R.M.E.N.G.,M.E.N.G.E.N.		
g)			M.V.C.		R.P.R.E.I.S.,P.R.E.I.S.E.		
h)			E.N.D.		R.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

- 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 . | PRCK | BEIRRG, BEIRRG

- 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	Operation	Operand und Bemerkungen
1	8	10 14 20 25 30
	:	
WEITER.	P.R.C.K.	BETRAG., BETRAG.
	P.R.C.K.	KOSTEN., KOSTEN.
	R.P.	SUMME., BETRAG.
	R.P.	GEKOSTEN., KOSTEN.
B.2.	M.V.C.	MENGE., MENGEN.
	M.V.C.	M.PREIS., 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.	P.C.K.	BETRAG, BETRAG.
	P.C.K.	KOSTEN, KOSTEN.
	R.P.	SUMME, BETRAG.
	R.P.	GEKOSTEN, KOSTEN.
B.Z.	M.V.C.	MENG, MENGEN.
	M.V.C.	PREIS, PREISE.
	:	
	B.	WEITER.

Abb. 7

12 Gültige Symbole sind z.B.:

- a) K.T.O.N.R.
- b) H.B.T.4.5.6.
- c) N.
- d) H.2.
- e) B.E.R.E.I.C.H.E

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) B.E.T.R. 4.8.
- b) 2.7.5.3.E.T.R.
- c) P.E.R.S.O.N.F.L.N.R.
- d) P.E.R.S..N.R.

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		Operation		Operand und Bemerkungen													
1	8	10	14	20	25	30	35	40	45	50	55	60	65	70	78	80	
*DER STERN, IN		STELLE 1, KENNZEICHNET, EINE BEMERKUNGSZEILE.														0.1.0	
* BEMERKUNGEN, DÜRFEN NUR BIS STELLE 71, REICHEN UND KOENNEN, IN DER FOL-																0.2.0	
* GENDEN ZEILE FORTGESETZT WERDEN. JEDE BENÖTIGTE ZEILE IST HIERBEI																0.3.0	
* DURCH EINEN STERN ZU KENNZEICHNEN.																0.4.0	
																0.5.0	
*BEMERKUNGEN KOENNEN AUCH RECHTS VON DEN OPERANDEN GESCHRIEBEN WERDEN,																0.6.0	
*SIE MÜSSEN JEDOCH MINDESTENS DURCH 1. LEERSTELLE VON DIESEN GETRENNT																0.7.0	
*SEIN. DIE BEMERKUNGEN RECHTS VON DEN OPERANDEN DÜRFEN KEINEN * IN																0.8.0	
*STELLE 1 ERHALTEN. ZUM BEISPIEL:																0.9.0	
																1.0.0	
																1.1.0	
		CLC		ENR, KCENDE. (VERGLEICH MIT KONSTANTE "ENDE").													1.2.0
R1		MVC		FELD1, FELD2. UEBERTRAGEN.													1.3.0
																1.4.0	
*ALLE ZEICHEN, ALSO BUCHSTABEN, ZIFFERN UND SONDERZEICHEN, DÜRFEN FÜR																1.5.0	
*BEMERKUNGEN BENUTZT WERDEN. AUCH LEERSTELLEN SIND ERLAUBT.																1.6.0	
																1.7.0	
* WENDEN SIE SICH NUN BITTE LERNELEMENT 14 ZU!																1.8.0	
																1.9.0	

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 Programm-
liste (auch Umwandlungsliste genannt), die bei der Um-
wandlung 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 Programm-
schemas gekennzeichnet sein.

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

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

Stern

1

71

Stern

Anweisung															Karte Nr.	
Name		Operation		Operand und Bemerkungen												
1	5	10	14	20	25	30	35	40	45	50	55	60	65	70	78	80
a)	PROGRAMM. ZUR			ERSTELLUNG. DER. LISTE. 'ZUGRENZE'.												010
																020
b)	B2		RP	SUMME, BETRAGBEGINN. EINES. NEUEN. PROGRAMMABSCHNITTS. INNER-												030
c)	HALB. PROGRAMM		R18.													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

Programm Nr. 15		Lochinweise							Problem	<table><tr><td>L</td><td>1</td><td>5</td></tr><tr><td>73</td><td></td><td>75</td></tr></table>	L	1	5	73		75
L	1	5														
73		75														
Erstellen der Liste Lohn- und Materialkosten		Zeichen	7		*											
Programmierer	Alfons Hermes	Datum	10.3.68	Lochung	5-8	11-4-8										
								</								

[illegible]

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.

[illegible]

68

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



Programm Nr. 13	Lochhinweise	Problem R13
Kostenrechnung	Zeichen	Seite 015 von 5
Programmierer Edi Pulcher	Datum 4.4.68	Lochung

Anweisung																Karte Nr.
Name		Operation	Operand und Bemerkungen													
1	8	10	14	20	25	30	35	40	45	50	55	60	65	70	78	80
R2		MVC		RNR, ENR												010
		RP		SUMME, BETRAG												020
		RP		GESKOST, KOSTEN												030
		:														040
		:														050
		:														060
		:														070
		:														080
		:														090
		:														100
		:														110
		:														120
		:														130
		CLC		FELDB, KONST.100												140
		BH		R2												150
		END		ANFANG												160
																170
																180
																190
																200
		PRCK		BETRAG, BETRAG												011
		PRCK		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. KON1~~00~~).

Die Numerierung der Zeilen erfolgt nicht fortlaufend mit 1 beginnend, die Zeilen sind vielmehr mit 10, 20, 30, 40 usw. bis 200 durchnummeriert. 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.

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 Kartenummerierung 10, (Z) , (Z) , 20, 30, usw.

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 .

PR.C.K.	BETRAG, BETRAG.
PR.C.K.	KOSTEN, KOSTEN.

11

12

Programmschema

[illegible]

72

- 23 Wesentlich für die Umwandlung des symbolischen Programms ist das, was in die Anweisungsfelder (1-8), (10-14) und (WW) (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) .

- 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.

Name

Operation

Operand und Bemerkungen

9 (und) 15

Umwandlungsliste/Programmliste



Programm Nr. 12		Lochhinweise		Problem <u>012</u>	
Monatliche Dispositionsliste		Zeichen	<u>1</u> <u>8</u>		
Programmierer Johann Alois Kohr	Datum 11.11.68	Lochung	5-8 12		Seite <u>012</u> von <u>4</u>

Anweisung																Karte Nr.
Name		Operation	Operand und Bemerkungen													
1	8	10	14	20	25	30	35	40	45	50	55	60	65	70	78	80
*, VERARBEITUNG			DER DAUERAUFTRAGE													010
																020
B4		MVC	RNR, ENR													030
																040
B5		CLC	BETRAG, KONST. 1000. EINE KONSTANTE 1000. WIRD MIT DEM BE-													050
TRAG VERGLICHEN																060
																070
BETA		PACK	FIELD, ALPHA8													080
																090
1RB		RP	SUMME, BETRAG													100
																110
GBMMB		RP	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 KONST1000 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 Maschineninstruktionen - 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 Maschineninstruktionen-Anweisungen, die vom Programmierer symbolisch angegeben werden, werden durch die Umwandlung in echte Instruktionen (Instruktionen in Maschinsprache) umgewandelt.

Die Assemblerinstruktionen - 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 Maschineninstruktionen-Anweisungen sind die Assemblerinstruktionen-Anweisungen im Maschinenprogramm (echten Programm) nicht mehr enthalten.

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

START

PROGRAMMANFANG

(Start program)

Beispiel einer Instruktion:

Name		Operation	Operation
1	8	10	20
		START	8.00

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 ein 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:

Name								Operation				Operation											
1								8				10				14				20			
								START				8.00											

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

Name		Operation	Operation
1	8	10	20
		START	6.00

600

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								Operation						Operand					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
a)										S	T	A	R	T		9	6	0		
b)										S	T	A	R	T		7	2	0		
c)										S	T	A	R	T		1	0	4	0	



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.

- 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		800
ANFANG		BRLR		9,0
		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.

Ende

letzte Anweisung

ANFANG | BRLR | 9,0

Übungen

Geben Sie bitte folgende END-Anweisungen an:

- Die END-Anweisung soll auf eine erste auszuführende Instruktion mit dem Namen BEGINN verweisen.
- Die als erste auszuführende Instruktion hat den Namen A3.

	Name		Operation		Operand und Beme	
	1	8	10	14	20	25
a)			END		BEGINN	
b)			END		A3	

ORG

SETZEN ZUORDNUNGSSZÄHLER

(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
		O.R.G.		EBER	
		O.R.G.		EBER+300	
		O.R.G.		*-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		Operand	
1	8	10	14	20	25
		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.		BER-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 Bemerkung	
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 ZUORDNUNG SZÄHLER

(Set location counter)

Beispiele für Instruktionen:

Name		Operation	Operand und Be
1	8	10	14
		20	25
		O.R.G.	E.B.E.R.
		O.R.G.	E.B.E.R.+3.00.
		O.R.G.	*-5.00.

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								Operation								Operand							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
a)											O	R	G					*	+	1	0	0		
b)											O	R	G					*	-	5	0			

Übungen

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

- Der Zuordnungszähler soll auf den Wert des
Symbols AREA1 gesetzt werden.
- Der Zuordnungszähler soll, bezogen auf ein
Symbol FELDA, um 80 vermindert werden.
- Der Zuordnungszähler soll, bezogen auf ein
Symbol EBER, um 45 erhöht werden.
- Der momentane Wert des Zuordnungszählers
soll um 180 erhöht werden.
- Der momentane Wert des Zuordnungszählers
soll um 185 vermindert werden.

Name		Operation	Operand und Beme	
1	8	10	14	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.

Wieviele 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

Name		Operation		Operand und Bemerkungen		
1	8	10	14	20	25	30
P.E.R.S.N.R.		D.S.		C.L.5.		

104

- 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.

1	8	10	14	2
P.E.R.S.N.R.		D.S.		C.L.5.

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 und							
1		8		10		14		20															
ARTBEZ								DS								CL15							

ARTNR	ARTBEZ	AMENGE	EPREIS	GESPRESS
1 - 5 6	-	25 26 - 31	32 - 35	36 - 42

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	Operation	Op
1 8 10 14 20		
R.T.N.R.	DS.	CL.5.
R.T.BEZ.	DS.	CL.20.
R.MENGE.	DS.	CL.6.
E.PREIS.	DS.	CL.4.
G.E.S.PREIS.	DS.	CL.7.

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

a) Daten in der Lochkarte

Feld 1	Feld 2	Feld 3	
1 - 5	6 - 15	16 - 30	

b) Codierung

Name	Operation	Operand und
1	10	20
8	14	21
	START	1,0,0,1
FELD,1	DS	CL,5
FELD,2	DS	CL,1,0
FELD,3	DS	CL,1,5

c) Zuordnung im HSP

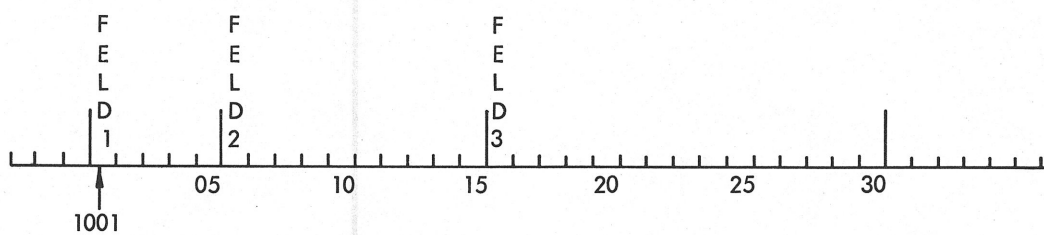


Abb. 4

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

Feld 1	Feld 2	Feld 3
1 - 5	6 - 15	16 - 30

b) Codierung

Name	Operation	Operand und
1	8	10 14 20 22
	START	1 0 0 1
FELD.1	DS	CL 5
FELD.2	DS	CL 1 0
FELD.3	DS	CL 1 5

c) Zuordnung im HSP

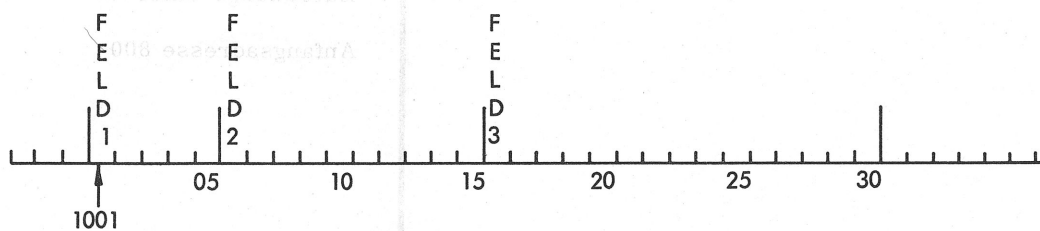


Abb. 4

- 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

Feld 1				Feld 2				Feld 3				
1	-	5	6	-	15	16	-	30				

b) Codierung

Name	Operation	Operand und
1	8	14
20	2	
	START	1001
FELD 1	DS	CL 5
FELD 2	DS	CL 10
FELD 3	DS	CL 15

c) Zuordnung im HSP

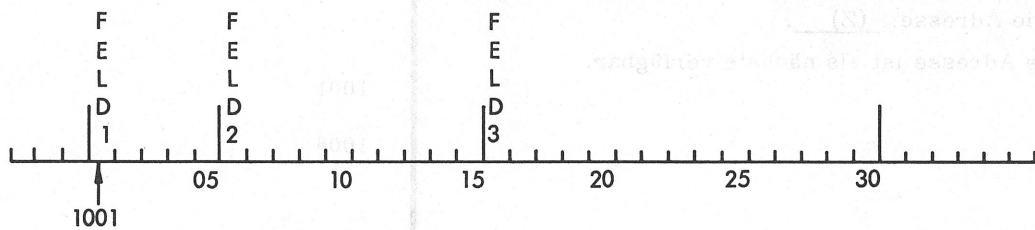


Abb. 4

12

FELD2 | D.S. | CL10 |

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	Operation	Operanc
1	8	10 14 20
.....	START	1.6.0.1
ENUMMER	D.S.	C.L.6
RNUMMER	D.S.	C.L.6
EPREIS	D.S.	C.L.8
GESPRESS	D.S.	C.L.10
.....		
.....		

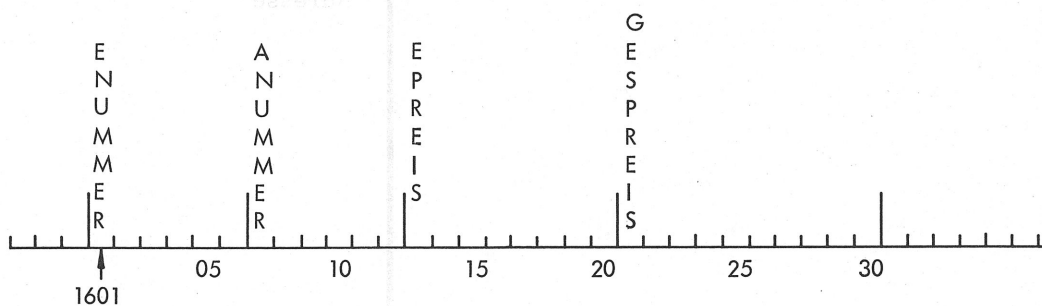


Abb. 5

Die symbolische Adressierung

Wie Sie bereits wissen, ist der Hauptspeicher von 0 beginnend aufwärts durchnummeriert. 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 GESPRESIS 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 symbolische Bezeichnungen nennt man symbolische Adressierung.

ANUMMER ist z. B. die symbolische Adresse (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 Adress- und ein Längenattribut.

GESPRESIS | DS . . . | CL10 . . .

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) .

GESPRESIS

1621

10

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

Name	Operation	Operand
1	8	10 14 20
	START	1.6.0.1.
ENUMMER	DS.	CL.6.
RNUMMER	DS.	CL.6.
EPREIS	DS.	CL.8.
GESPRESS	DS.	CL.1.0.

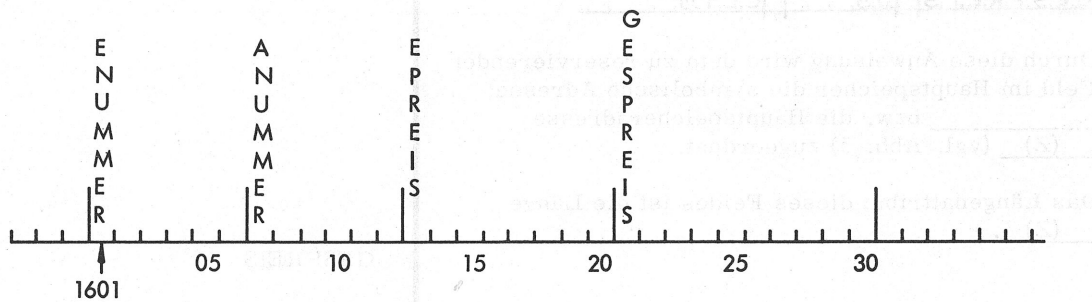


Abb. 5

- 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

Betrachten Sie bitte Abb. 6. -

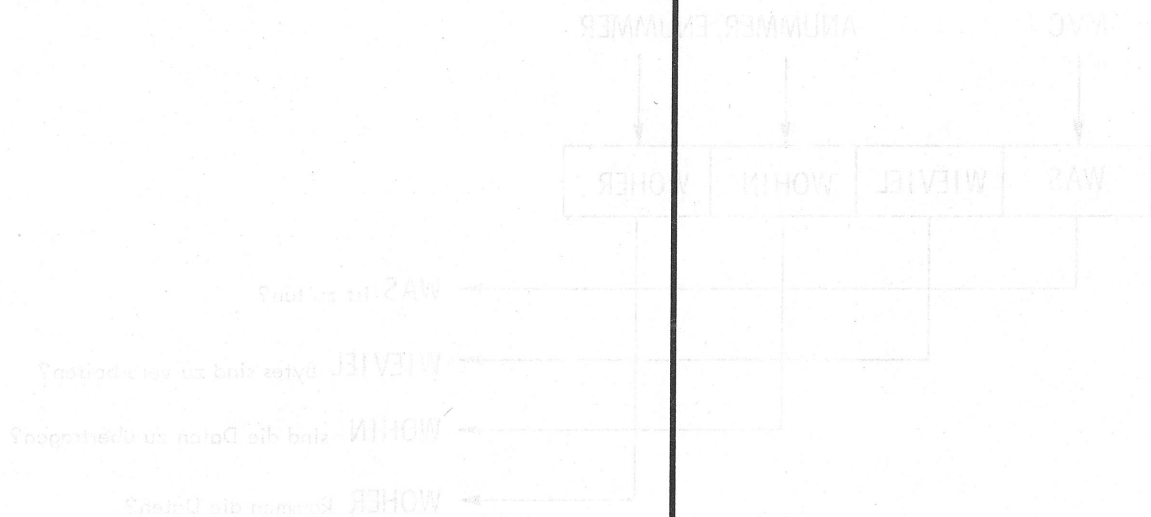
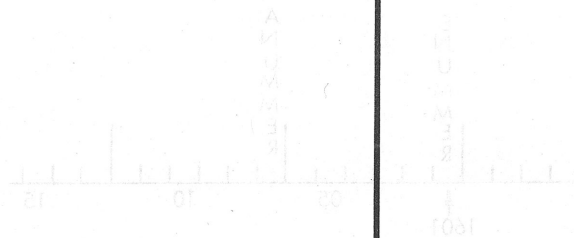
Wir hatten festgestellt:
Durch die Maschineninstruktion

MVC , ANR, ENR ,

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.R.T	1,6,0,1
E.N.U.M.M.E.R.	D.S.	C.L.b.
R.N.U.M.M.E.R.	D.S.	C.L.b.

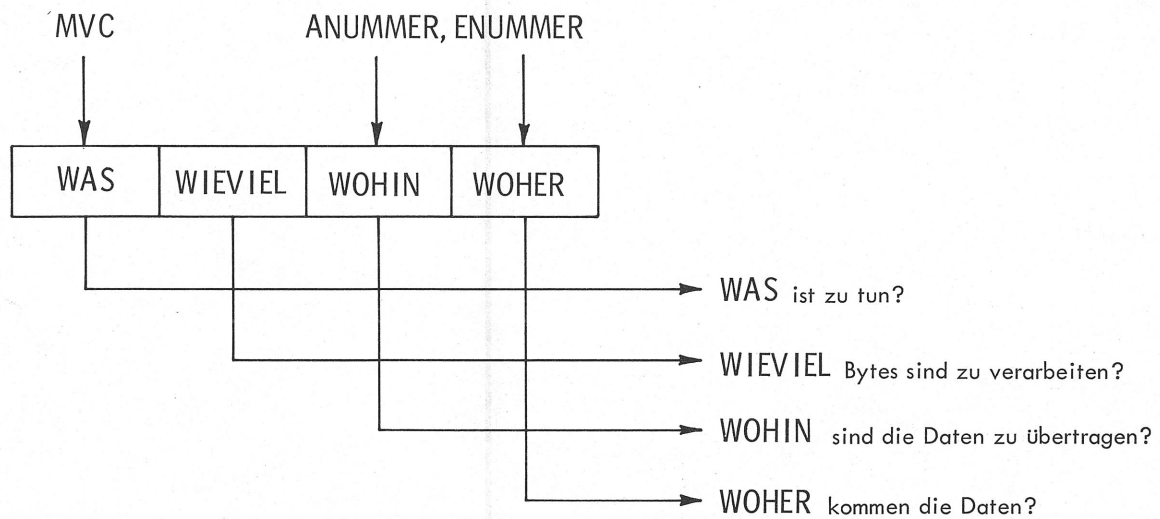
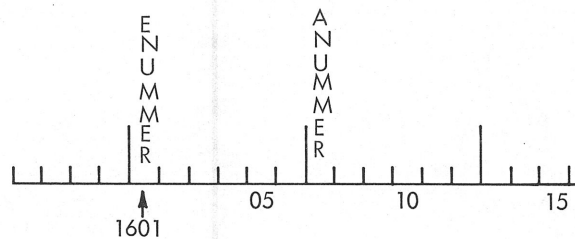


Abb. 7

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	30
		M.V.C.		R.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.R.T.	1.6.0.1.
E.N.U.M.M.E.R.	D.S.	C.L.6.
R.N.U.M.M.E.R.	D.S.	C.L.6.

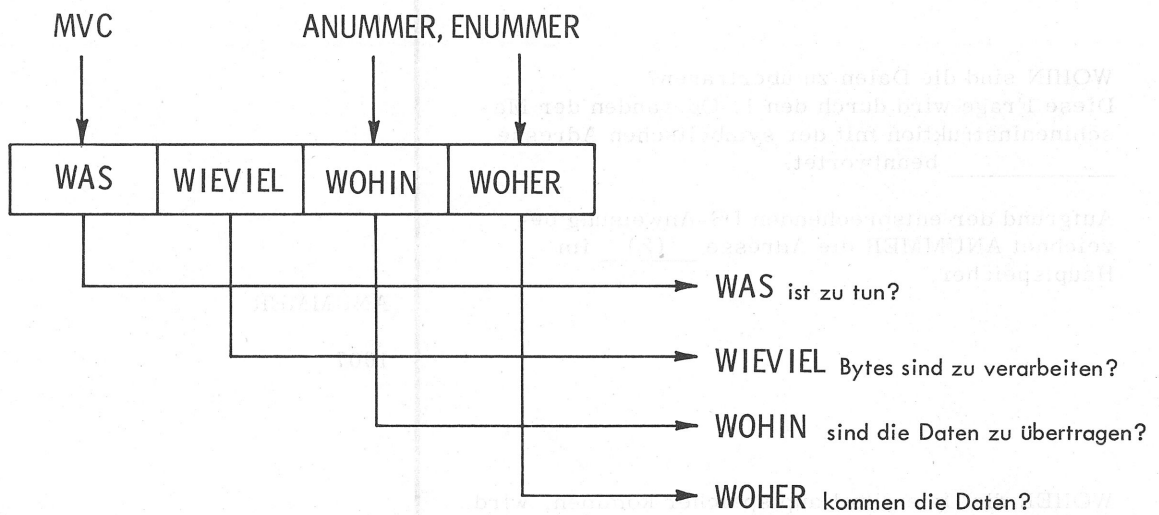
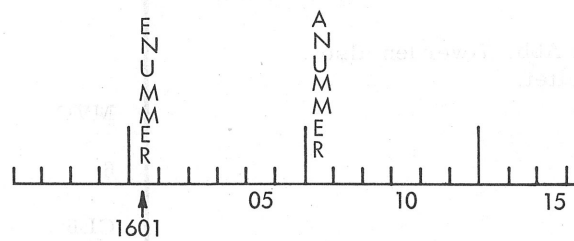


Abb. 7

- 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	16	20	24	28	32	36	40	44
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:

M.V.C. . . | A.NUMMER, E.NUMMER. . .

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

Länge

Adresse

Adresse

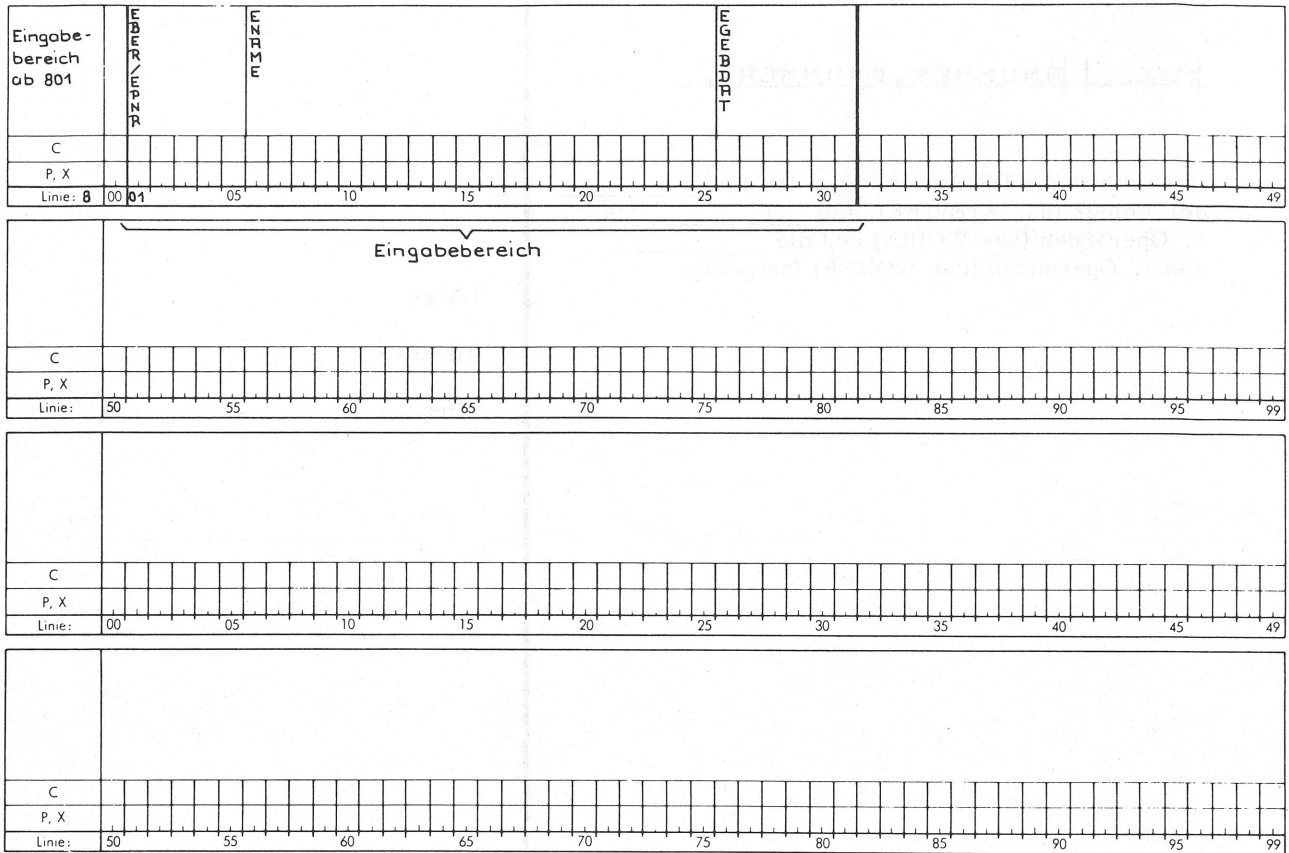


Abb. 8

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.

Personal-Nr.	Name	Geburtsdatum
1 - 5 6	-	25 26 - 31

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	EBER	EBER
	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.

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

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.

801

31

Name		Operation		Operand
1	8	10	14	20
E.B.E.R. . . .		D.S. . . .		C.L.3.1. . .

832

IBM System / 360 Speicherbelegung



EBER

134

29

Name				Operation				Of			
1	8	10	14	16	22	24	30	32	38	44	50
E	B	E	R	.	.	.	.	D	S	.	.
.	.	.	.	.	.	.	.	C	L	3	1

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	16	22	24	30	32	38	44	50
E	B	E	R	.	.	.	.	0	C	L	3
.	.	.	.	.	.	.	.	1	.	.	.

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

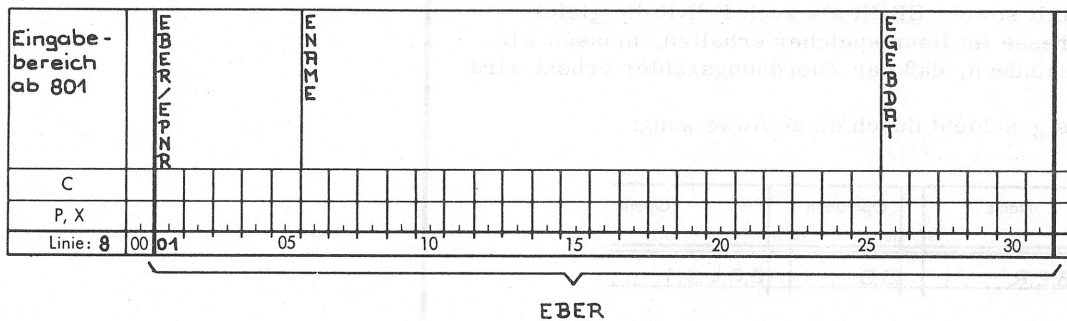


Abb. 9

- 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		8.0.1.

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

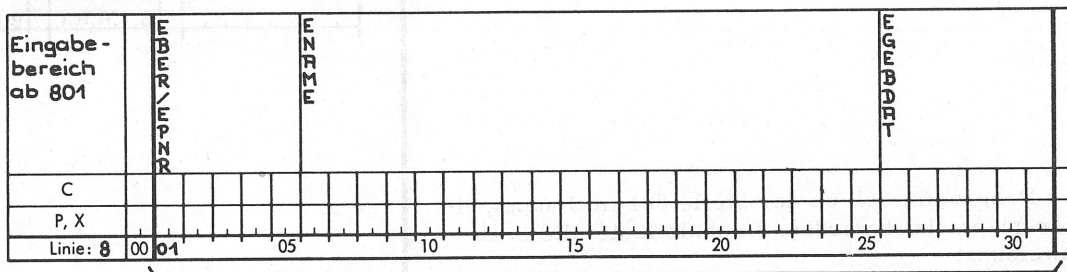
Name		Operation		Operanc
1	8	10	14	20
.		S.T.A.R.T		8.0.1.

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		8.0.1.
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



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		801
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).

Eingabe- und Ausgabe- bereich START 1001	EBER/ EKNR	ESOLL	FHBER/ FHNZ	FBER/ FNZR	FSRLO
C					
P, X					
Linie: 10	00	05	10	15	20
					25
					30

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		Operation		Operand
1	8	10	14	20
		START		1001
EBER		DS		0CL20
EKNR		DS		CL6
ESOLL		DS		CL7
EHRBEN		DS		CL7
ABER		DS		0CL13
RKNR		DS		CL6
ASRLDO		DS		CL7

Nr.								Datum				Umsatz			
1	-	4	5	-	10	11	-	16	17	-	24				

IBM System / 360
Speicherbelegung



Eingabe- bereich	EBER		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 stellenge-
recht 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 Speicher-
stellen 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

- 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				Operation				Operand					
1	8	10	14					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.R.				D.S.				C.L.2.					

Überprüfen Sie Ihre Codierung genau auf ihre Richtigkeit! Die Summe aller Feldlängen = L₀ des EBER.

	Name				Operation				Operant			
	1	8	10	14	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.R.				D.S.				3.C.L.4.			

IBM System / 360
Speicherbelegung



C P, X Linie: 8	F E L D R														
	00				05					10					15

Abb. 14

41

Name	Operation	Operand
1 8	10 14	20
FELDA	DS	CL4

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

Name	Operation	Operand
1 8	10 14	20
FELDA	DS	0CL4

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.

Name	Operation	Operand
1 8	10 14	20
FELDA	DS	3CL4

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

Die 3 im Operanden 3CL4 bezeichnet man als den Duplizierfaktor. Wie oft für FELDA ein 4 Bytes großes Feld im Hauptspeicher reserviert werden soll, wird durch den _____ angegeben.

Duplizierfaktor

- 42 Die Angabe eines Duplizierfaktors beeinflusst nicht 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?

FLD1 . . . | 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.

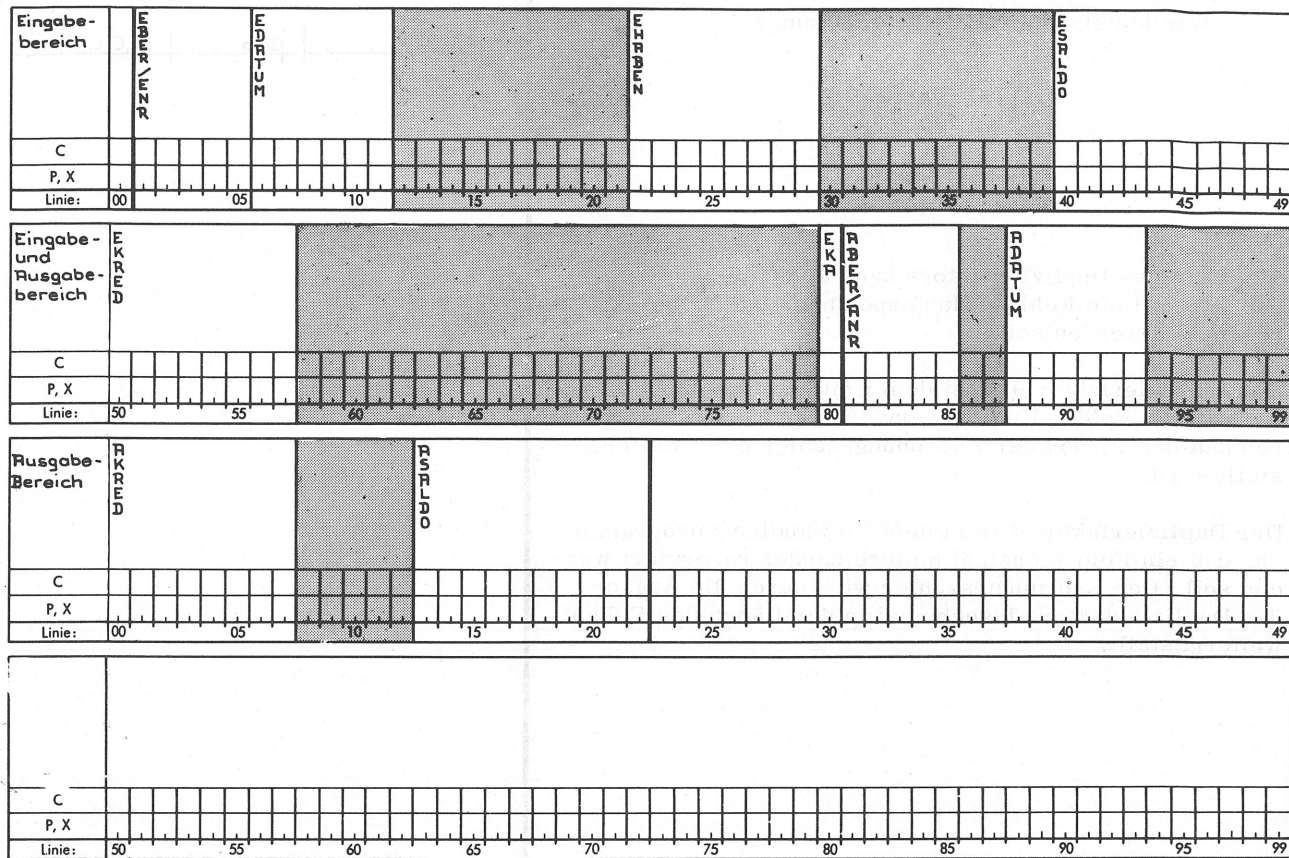


Abb. 15

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		Operation	Operand
1	8	10	14
			20
EBER		DS	0CL80
ENR		DS	CL5
EDATUM		DS	CL6
		DS	CL10
EHABEN		DS	CL8
		DS	CL10
ESALDO		DS	CL10
EKRED		DS	CL8
		DS	CL22
EKA		DS	CL1
ABER		DS	0CL42
ANR		DS	CL5
		DS	CL2
ADATUM		DS	CL6
		DS	CL6
AKRED		DS	CL8
		DS	CL5
ASALDO		DS	CL10



Band 1, Seite 27

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 $\pm$ 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				Operand			
1	2	3	4	5	6	7	8	9	10	11	12
F	E	L	D	R	.	.	.	.	D	S	.
3	C	L	4	.	.	.	.	.	.	.	.

IBM System / 360 Speicherbelegung



F															
C															
P, X															
Linie: 8															

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.

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

Wie muß der Operand lauten?

FELDA

4

FELDA+4

FELDA+8

IBM System / 360
Speicherbelegung

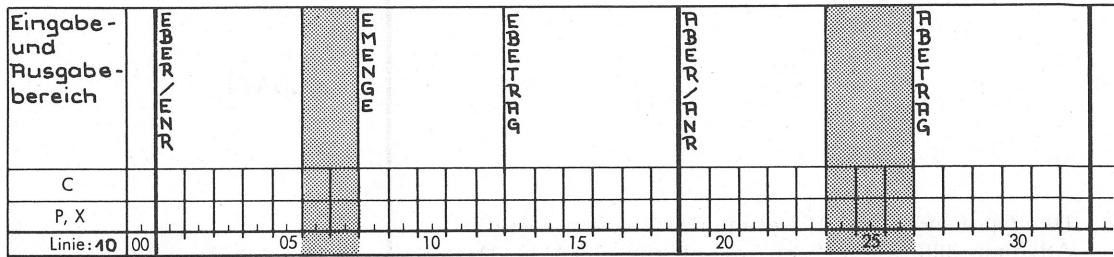


Abb. 2

- 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.

Name		Operation		Operand	
1	8	10	14	20	
F.E.L.D.P., . . .		D.S., . . .		C.L.15, . . .	

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
F.E.L.D.R.	D.S.	C.L.1.5.
	↓	
	M.V.C.	F.E.L.D.R.(1.0.), K.ONST.X

501

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 vor der impliziten (definierten) Länge.
5. Für Operanden ohne explizite Längenangabe gilt immer die definierte (implizite) Länge.

IBM System / 360
Speicherbelegung



Eingabe- und Ausgabe- bereich	EBER/ENR	EMENGE	EBETRAG	ABER/ABR	ABETRAG
C					
P, X					
Linie: 10	00	05	10	15	20
					25
					30

Name	Operation	Operand
1	8	10
EBER	D.S.	0.C.L.1.8.
ENR	D.S.	C.L.5.
	D.S.	C.L.2.
EMENGE	D.S.	C.L.5.
EBETRAG	D.S.	C.L.6.
ABER	D.S.	0.C.L.1.4.
ABR	D.S.	C.L.5.
	D.S.	C.L.3.
ABETRAG	D.S.	C.L.6.

Abb. 1

1 Betrachten Sie bitte Abb. 1. -

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

M.V.C. | ANR, ENR

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

2 EBETRAG hat die implizite Länge (Z).

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

Denn:

_____ Länge geht vor _____ Länge !

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 einen Operanden ansprechen.

Wie muß dieser Operand lauten?

5

5

6

5

explizite

impliziter

EMENGE(11)

[illegible]

Name	Operation	Operant
1	8	10
2	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

166

- 4 Durch den Operanden EBETRAG(5) werden nur die ersten 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



Eingabe- und Ausgabe- bereich	EBER ENZ	EMENGE	EBETRAG	EBER ENZ	EBETRAG
C					
P, X					
Linie: 10	00	05	10	15	20
					25
					30

Name	Operation	Operand
1	8	10
14	20	
EBER.	DS.	Ø.C.L.18.
ENR.	DS.	C.L.5.
	DS.	C.L.2.
EMENGE.	DS.	C.L.5.
EBETRAG.	DS.	C.L.6.
EBER.	DS.	Ø.C.L.14.
ENR.	DS.	C.L.5.
	DS.	C.L.3.
EBETRAG.	DS.	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



Eingabe- und Ausgabe- bereich	EBER ENZ	EMENGE	EBER ENZ	EBER ENZ	EBER ENZ
C					
P, X					
Linie: 10	00	05	10	15	20
					25
					30

Name	Operation	Operand
1	8	10
14	20	
EBER. . . .	DS. . . .	ØCL18. . .
ENR. . . .	DS. . . .	CL5. . . .
	DS. . . .	CL2. . . .
EMENGE. . .	DS. . . .	CL5. . . .
EBETRGG. . .	DS. . . .	CL6. . . .
EBER. . . .	DS. . . .	ØCL14. . .
ENR. . . .	DS. . . .	CL5. . . .
	DS. . . .	CL3. . . .
EBETRGG. . .	DS. . . .	CL6. . . .

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



Eingabe- bereich	EBER	EKT.D.N.R.	EKREDIT	ESBLDO
C				
P, X				
Linie: 8	00	05	10	15
				20
				25

Name	Operation	Operanc
1	8	10
14	20	
EBER	D.S.	0.C.L.2.4.
	D.S.	C.L.5.
EKT.D.N.R.	D.S.	C.L.5.
	D.S.	C.L.2.
EKREDIT	D.S.	C.L.5.
ESBLDO	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



Eingabe- bereich	EBER	EKTOR	EKREDIT	ESBLDO
C				
P, X				
Linie: 8	00	05	10	15
				20
				25

Name	Operation	Operanc
1	8	10
14	20	
EBER	DS	0.C.L.2.4.
	DS	C.L.5.
EKTOR	DS	C.L.5.
	DS	C.L.2.
EKREDIT	DS	C.L.5.
ESBLDO	DS	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

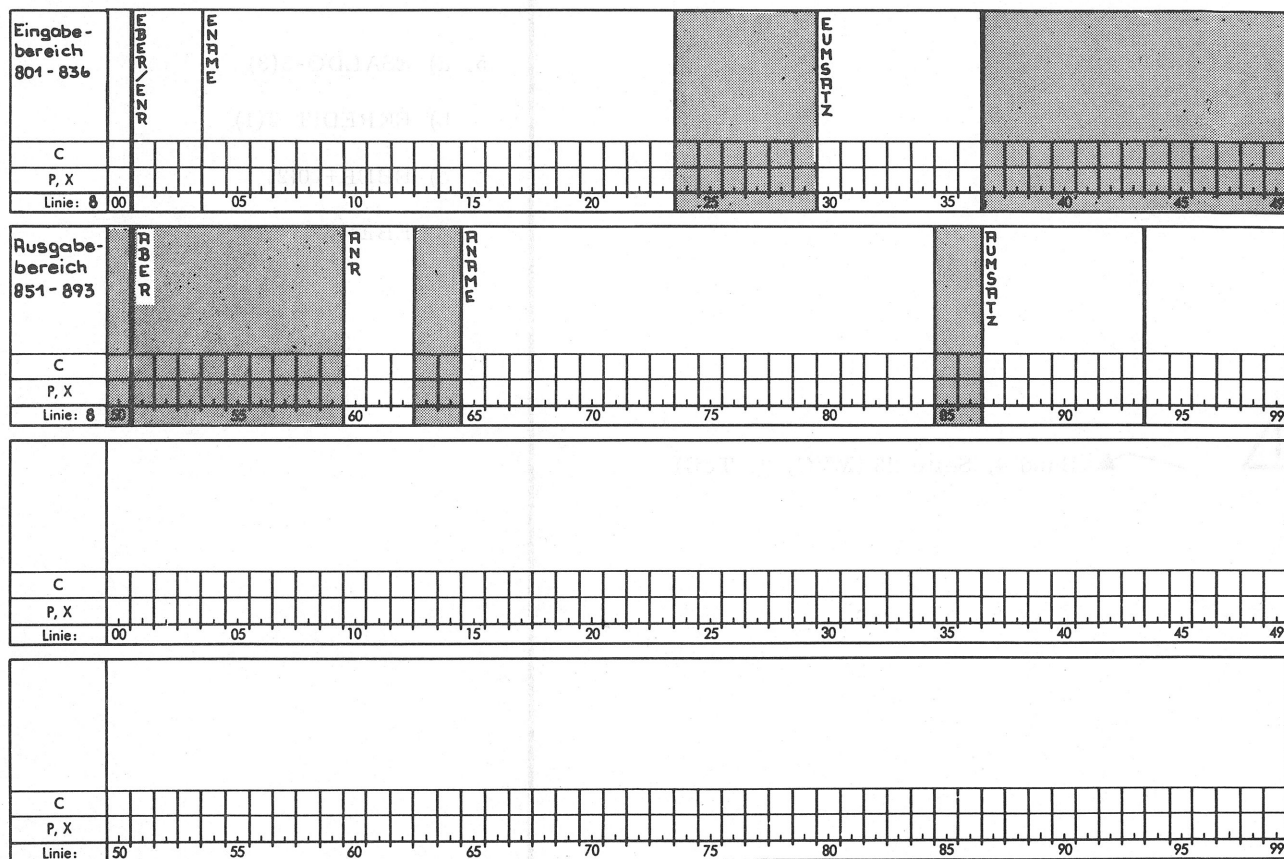


Abb. 1

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.

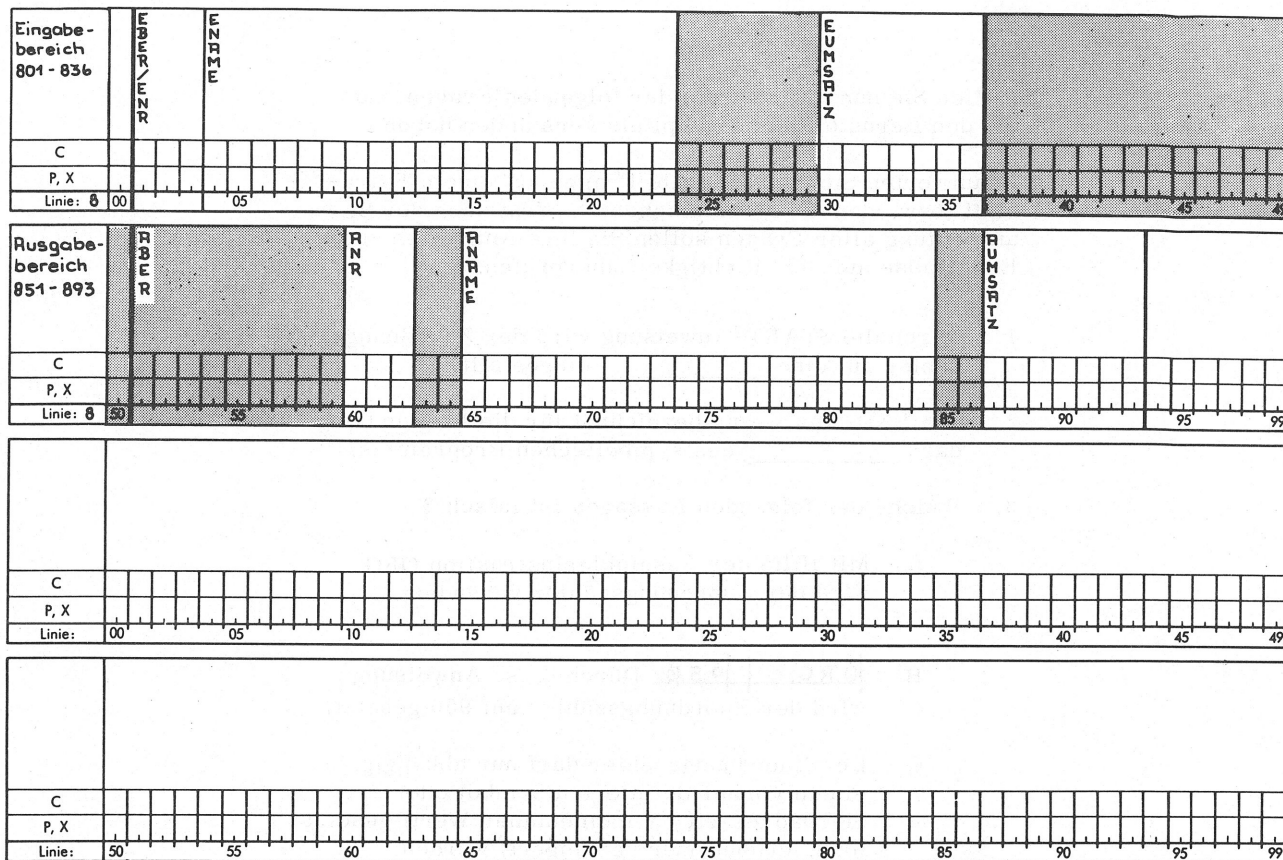


Abb. 1

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 0.R.G. | 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)

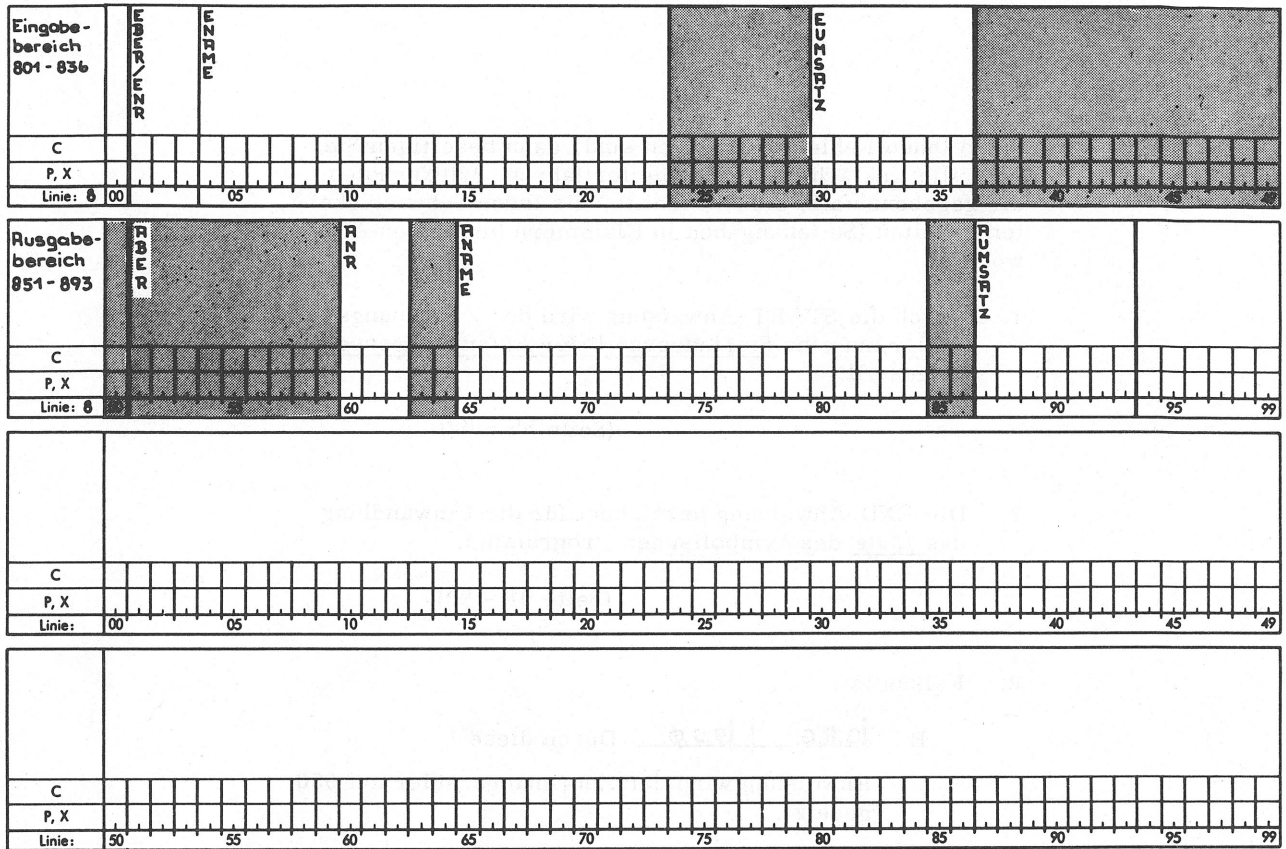


Abb. 1

4.

Name		Operation		Operand	
1	8	10	14	20	
a)	E.B.E.R.	D.S.		Ø.C.L.3.6. . . .	} (Seite 131 - 153)
	E.N.R.	D.S.		C.L.3.	
	E.N.R.M.E. . . .	D.S.		C.L.2.Ø.	
		D.S.		C.L.6.	
	E.U.M.S.R.T.Z. .	D.S.		C.L.7.	
b)		O.R.G.		E.B.E.R.+5.Ø. . .	} (Seite 131 - 153)
c)	E.B.E.R.	D.S.		Ø.C.L.4.3. . . .	
		D.S.		C.L.9.	
	E.N.R.	D.S.		C.L.3.	
		D.S.		C.L.2.	
	E.N.R.M.E. . . .	D.S.		C.L.2.Ø.	
		D.S.		C.L.2.	
	E.U.M.S.R.T.Z. .	D.S.		C.L.7.	

5. a) EUMSATZ(5)

b) EBER(3)

(Seite 155 - 171)

c) ABER+9(5)

d) ANAME-2(2)

KAPITEL IV

DIE DEFINITION DER KONSTANTEN

Beispiel	Zeichenkonstante - C	Gepackte Konstante - P	Hexadezimalkonstante - X	Wortkonstante - F	Halbwortkonstante - H	Adresskonstante - A
	1	2	3	4	5	6

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		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

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 Lern-elementen 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.

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.

- 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.

6

Name		Operation		Operand und Bemerkungen		
1	8	10	14	20	25	30
KONST1		DC		C,'RABATT', = 1,0,%,'		

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

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

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.

C

11

256

KONST1

Zeichenkonstante

12

IBM System / 360
Speicherbelegung



Zeichen- konstan- ten	K C L I S T E											K C 1 5 0											K C 1 0 0 0										
C		L	I	S	T	E		N	R	.	3	1	5	0	1	.	0	0	0	,	0	0											
P, X																																	
Linie: 9	00																																

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 _____.

- 8 Entnehmen Sie bitte Ihren Arbeitsmitteln ein Programmschema.

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

Byte

Name		Operation		Operand und Bemerkung	
1	8	10	14	20	25
K.C.L.I.S.T.E.		D.C. . .		C.'L.I.S.T.E. N.R.-3.'	
K.C.150. . .		D.C. . .		C.'150.'	
K.C.1000. . .		D.C. . .		C.'1.000,00'	

IBM System / 360
Speicherbelegung



Zeichen- konstan- ten	K C L I S T E	K C 1 5 0										K C 1 0 0 0									
C	L	I	S	T	E	N	R	.	3	1	5	0	1	.	0	0	0	,	0	0	
P, X																					
Linie: 9	00																				

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	Längenattribut
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 Längenschlüssels anzugeben.

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

10

Name				Operation				Operand und Be							
18				1014				2025							
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.

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

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

4

4

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 Buchsta-
ben _____ der angegebenen Konstanten ge-
speichert 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.

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

8

2 (A B C D E F)

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		Operation	Operand und Bemerkungen		
1	8	10	14	20	30
KCBLANK		DC		C'	
oder:					
KCBLANK		DC		CL15'	

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		Operation	Operand u		
1	8	10	14	20	
a) KCBLANK		DC		CL30'	
b) KCBLANK		DC		CL59'	
c) KCBLANK		DC		CL24'	

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.'NR.'

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 (NRNRNRNR)

- 16 Wie müssen die Operanden lauten, wenn
- die Konstante 7 zehnmal,
 - die Konstante DM fünfmal,
 - die Konstante ABCD dreimal hintereinander gespeichert werden soll?

- 10C ' 7 '
- 5C ' DM '
- 3C ' ABCD '

- 17 Wir haben bereits bei der Behandlung der DS-Anweisung festgestellt, daß die Angabe eines Duplizierfaktors das Adress- und Längenattribut nicht beeinflußt. - 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.'NR.'

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

DAS FORMAT DES OPERANDEN FÜR DAS DEFINIEREN EINER ZEICHENKONSTANTEN

KONST

DC

5 C L 3 ' A B A '

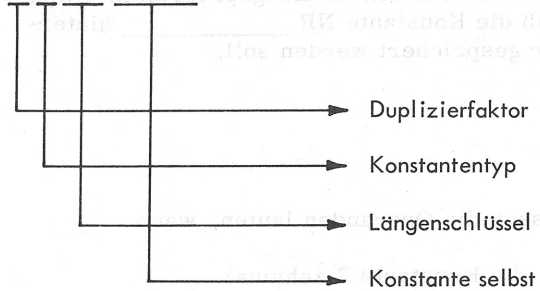


Abb. 3

- 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 Zeichenkonstante 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

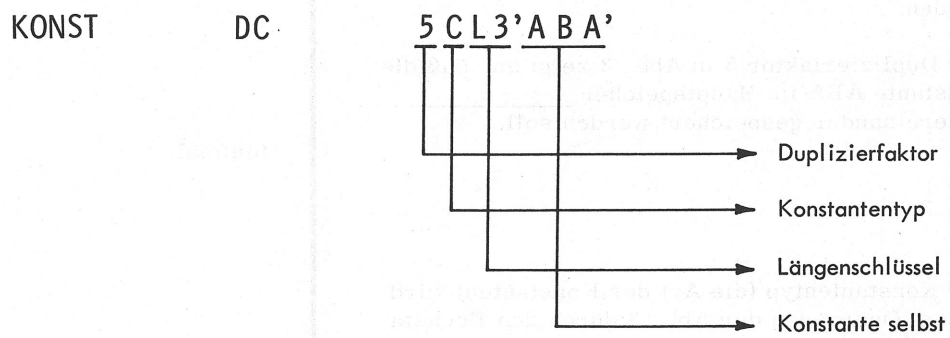


Abb. 3

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 doppelt dargestellt werden.

z. B.:

Name		Operation	Operand und Bemerkungen		
1	8	10	14	20	30
K.O.N.S.T.1.		D.C.		C.'W.I.E. GEHT.' '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.

Abb. 4

Übungen

Entnehmen Sie bitte Ihren Arbeitsmitteln ein Programm-
schema.

- 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		Operation		Operand und Bemerkungen					
	1	8	10	14	20	25	30	35	40	
a)	Z.E.I.K.O.N.1.		D.C.		C'	L.O.H.N.B.B.R.E.C.H.N.U.N.G.	F.E.B.R.	7.7		
	Z.E.I.K.O.N.2.		D.C.		C'	100000				
	Z.E.I.K.O.N.3.		D.C.		C'	A.L.P.H.A.	88	B.E.T.H.		
b)	K.C.B.L.1.		D.C.		C.L.6					
	K.C.B.L.2.		D.C.		C.L.20					
c)	K.O.N.C.1.		D.C.		4.C.	H.B.				
	K.O.N.C.2.		D.C.		5.C.	1000				



Band 4, Seite 79 (CLC)

DIE GEPACKTE KONSTANTE - P

Beispiel:

Name				Operation				Operand 1			
1	8	10	14	20	28	36	44	52	60	68	76
K.ONST.P.				DC.				P, +150,			

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.

- 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.

P

gepackte

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.'+1.5.0.'

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 vor 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		Operation		Operand und Bem	
1	8	10	14	20	25
K.O.N.S.T.P.1.		D.C.		P.'1.000000'	
K.O.N.S.T.P.2.		D.C.		P.'+1.800000'	
K.O.N.S.T.P.3.		D.C.		P.'-500000'	

IBM System / 360 Speicherbelegung

Gepackte Konstanten	K O N S T A N T E N	K O N S T A N T E N	K O N S T A N T E N
C			
P, X	01000000	01800000	+500-
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.

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

- 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

2

1

rechten

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

	K P 1	K P 2	K P 3
	a)	b)	c)
C			
P, X	01255+0500377-010+		
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!

9

Name				Operation				Operanc			
1	8	10	14	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.

Name				Operation				Operand un			
1	8	10	14	20							
a) K.P.1.				D.C.				P.'1255.'			
b) K.P.2.				D.C.				P.'-500377			
c) K.P.3.				D.C.				P.'10.'			

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

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.

K.O.N.S.T.P. | D.C. | P.L5,'10000'

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.

	K O N S T P				
C					
P, X	00	00	10	00	0+
Linie:	00				

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

K.O.N.S.T.P. | D.C. | P.L2,'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.

	K O N S T P	
C		
P, X	55	1+
Linie:	00	

- 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.

	K P B	K P B	K P C	K P D	K P E
	a)	b)	c)	d)	e)
C					
P, X	250000+015-	01550+0120000-	00000000+		
Linie:	00	05	10	15	

2.

	Name	Operation	Operand und Bemerkung
	1	8	10 14 20 25
a)	K.P.B.	D.C.	P' 250000'
b)	K.P.B.	D.C.	P' - 15'
c)	K.P.C.	D.C.	P' 1550'
d)	K.P.D.	D.C.	P' - 1200000'
e)	K.P.E.	D.C.	P.L4' 0'
	oder :		
a)	K.P.B.	D.C.	P' + 250000'
c)	K.P.C.	D.C.	P' + 1550'
e)	K.P.E.	D.C.	P' 00000000'



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.'R.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. ? R.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)

1 0 1 0 0 1 1 0 | 1 1 1 1 0 1 0 0 |
1 Byte

- 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:

1 1 1 1 1 1 1 1

X' FF'

IBM System / 360 Speicherbelegung

Hexadezi- mal kon- stanten	KONST X1	KONST X2	KONST X3
C			
P, X	08R04P3DD44700FF	F5F6	
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.

- 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 KONSTX1 das am weitesten links stehende Halbbyte auf Null gesetzt.

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

KONSTX1

Name		Operation		Operand und Bemerkung	
1	8	10	14	20	25
KONSTX1		D.C.		X.'8A04A'	
KONSTX2		D.C.		X.'3DD44700FF'	
KONSTX3		D.C.		X.'B5F6'	

	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.'

a) 1 5
1 1 1 1 1 0 0 0 1 1 1 1 1 0 1 0 1

b) 1 5 C
0 0 0 0 1 0 0 0 1 0 1 0 1 1 1 1 0 0

c) 1 5
0 0 0 1 1 0 1 0 1

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.

- 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?

2

Halbbyte

Halbbyte

X' 15C'

- 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
KREFE...		DC...		P.'000000.'	
KREFE...		DC...		X.'000000C.'	
oder :					
KREFE...		DC...		P.L3.'0.'	
KREFE...		DC...		X.L3.'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!

Band 1, Seite 171



		KONX1	KONX2	KONX3	KONX4
		a)	b)	c)	d)
C					
P, X		4B57CF0FF0F9B1C550D			
Linie:	00		05		10

Name		Operation	Operand und f
1	8	10	25
K.O.N.X.1.		D.C.	X.'4B57CF.'
K.O.N.X.2.		D.C.	X.'FFF'
K.O.N.X.3.		D.C.	X.'A9B1C55.'
K.O.N.X.4.		D.C.	X.'D'

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.164.

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'64'

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

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

In jedem dieser Felder ist die Dezimalzahl 550 dargestellt.

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 Programm-
schema 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		Operation		Operand
	1	8	10	14	20
a)	K.O.N.F.1.		D.C.		F'4399'
b)	K.O.N.F.2.		D.C.		F'-5000'
c)	K.O.N.F.3.		D.C.		5F'1555'
	oder:				
a)	K.O.N.F.1.		D.C.		F'+4399'
c)	K.O.N.F.3.		D.C.		5F'+1555

DIE HALBWORTKONSTANTE - H

Mit Hilfe einer Halbwortkonstanten können konstante Dualdaten in Halbwortgröße im Hauptspeicher erzeugt werden.

Beispiel:

Name				Operation				Operand			
1				8				10			
K.O.N.S.T.H.				D.C.				H'61'			

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 Programm-
schema und definieren Sie die folgenden Halbwortkonstan-
ten:

- a) KONH1 = 32767
- b) KONH2 = -32768
- c) KONH3 = 100 (100 soll zweimal hintereinander
in je einem Halbwort gespeichert
werden).

	Name		Operation		Operand	
	1	8	10	14	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.		2.H.'100.'	
	oder:					
a)	K.O.N.H.1.		D.C.		H.'+32767.	
c)	K.O.N.H.3.		D.C.		2.H.'+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
1	8	10	14	20
R.K.O.N.S.T.		D.C.		R.(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



	EBER	FELDB	FELDB	FELDC	FELDE
C					
P, X					
Linie: 8	00	05	10	15	20

Name	Operation	Operand und Bemerkung
1	8	10 14 20 25
EBER.	DS.	0CL24.
	DS.	CL4.
FELDB.	DS.	CL5.
FELDB.	DS.	CL5.
FELDC.	DS.	CL5.
FELDE.	DS.	CL5.
a) BKON1.	DC.	R(FELDB).
b) BKON2.	DC.	R(FELDB-2).
c) BKON3.	DC.	R(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 Aus-
druckes ist der Inhalt der Adresskonstanten. In
Adresskonstanten werden also Hauptspeicheradres-
sen als konstante Daten definiert.

Name		Operation		Operand und	
1	8	10	14	20	2
PDRKON.		DC.		R(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
PDR.KON.		DC.	P.(FELD.C.)

00000000000000000000000001100101111

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



E F F E L D															
E B E R															
C															
P, X															
Linie: 9 00 05 10 15															

↑
906

Name								Operation								Operand							
18								1014								20							
E B E R								D S								0 C L 1 5							
								D S								C L 5							
E F E L D								D S								C L 1 0							

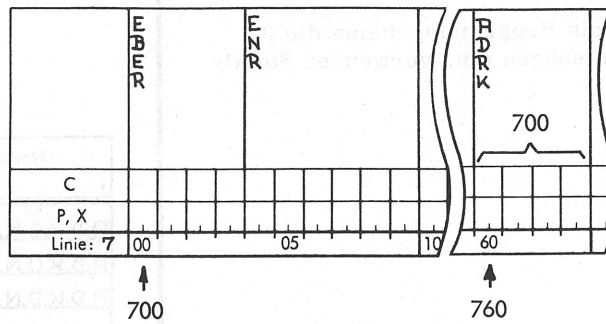
Abb. 3

- 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
ADKON.		DC.		R(EFELD.)
ADKON.		DC.		R(EBER+5.)
ADKON.		DC.		R(*-40.)



Name	Operation	Operant
1	8	10
		14
		20
EBER	D.S.	Ø.C.L.1.5.
	D.S.	C.L.4.
ENR	D.S.	C.L.6.
HDRK	D.C.	R.(EBER.)

Abb. 4

9 Betrachten Sie bitte Abb. 4. -

Es ist zu unterscheiden zwischen der Adresse einer Adresskonstanten im Hauptspeicher und dem Inhalt 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



	EBER	ENR	E.F.E.L.D.1	E.F.E.L.D.2	E.F.E.L.D.3	E.F.E.L.D.4
C						
P, X						
Linie: 10	00	05	10	15	20	25

Name	Operation	Oper
1	8	10 14 20
E.B.E.R. . . .	D.S. . . .	0.C.L.25. .
	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 Programm-
schema und definieren Sie die folgenden Hauptspeicher-
adressen als Adreßkonstanten. Die in Klammern gesetz-
ten Symbole sind im Operanden der DC-Anweisung zu ver-
wenden.

- a) Adresse 1010 (EFELD1)
- b) Adresse 1010 (EBER)
- c) Adresse 1014 (EFELD2)
- d) Adresse 1014 (EFELD4)
- e) Adresse 1010 (augenblicklicher Wert des Zu-
ordnungszählers: 1 100)

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.

	Name		Operation		Operand und Be	
	1	8	10	14	20	25
a)	ADKON1		DC		R(EFELD1)	
b)	ADKON2		DC		R(EBER+9)	
c)	ADKON3		DC		R(EFELD2)	
d)	ADKON4		DC		R(EFELD4-8)	
e)	ADKON5		DC		R(*-90)	

DAS FORMAT DES OPERANDEN BEI DER DEFINITION

DER FELDER UND BEREICHE

NAME

DS

d f L n

Duplizierfaktor

Feldschlüssel

Längenschlüssel

NAME	DS	d	f	L	n
BRKONT	02				
BRKONT	02				
BRKONT	02				
BRKONT	02				
BRKONT	02				
BRKONT	02				

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		Operation	
	1	8	10	14
a)	F.E.L.D.H.	...	D.S.	...
b)	F.E.L.D.F.	...	D.S.	...
c)	F.E.L.D.D.	...	D.S.	...

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		Operation		
1	8	10	14	20
a)	FELDH...	DS...	3H...	
b)	FELDF...	DS...	F...	
c)	FELDD...	DS...	10D...	

- 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 _____.

- 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.

- a) C
- b) H
- c) F
- d) D

C

2, 4 (und) 8

Name				Operation			
1	8	10	14				
H.W.F.E.L.D.				D.S.			
				H.			

Werden Halbwort-, Wort- oder Doppelwortfelder definiert, so wird diesen Feldern im Hauptspeicher automatisch eine Halbwort-, Wort- bzw. Doppelwortadresse zugeordnet.

Name				Operation			
1	8	10	14				2
				D.S.			
				Ø.H.			

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) ?

D.S.	Ø.F.
D.S.	Ø.D.

Schauen Sie sich nun bitte Merkblatt 306 (2 Seiten) an. -



Eingabe- bereich (Russchn)	ENR	E F F E L D 1	
C			
P, X			
Linie: 10	00	05	10
Konstan- ten	K C 1	K C B L E N K	K P 1
	a)	b)	c)
			d)
			e)
			f)
C	A B R E C H N U N G	1 0	
P, X			10000-05000+FFFFF00B4R16R
Linie: 10	50	55	60
C			
P, X			
Linie:	00	05	10
C			
P, X			
Linie:	50	55	60

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	}	Halbwortkon- stanten
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		Operation		Operand und Bemerkungen			
	1	8	10	14	20	25	30	35
a)	K.C.1.		D.C.		C.'ABRECHNUNG.10.'			
b)	K.C.B.L.R.N.K.		D.C.		C.L5.'			
c)	K.P.1.		D.C.		P.'-100000' oder: X.'100000D'			
d)	K.P.2.		D.C.		P.'50000' oder: P.'+50000'			
e)	K.H.E.X.1.		D.C.		X.'FFFFFFF'			
f)	K.H.E.X.2.		D.C.		X.'FB4B16B'			

(Seite 187-235)

3.

R.D.K.O.N.	D.C.	R.(E.F.E.L.D.1.)
oder		
R.D.K.O.N.	D.C.	R.(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		Operation		Operand und Bemerkungen			
	1	8	10	14	20	25	30	35
a)	K.F.1.		D.C.		F' 1000'	oder	F' +1000'	
b)	K.F.2.		D.C.		F' -500'			
c)	K.H.1.		D.C.		H' -850'			
d)	K.H.2.		D.C.		H' 32599'	oder:	H' +32599'	

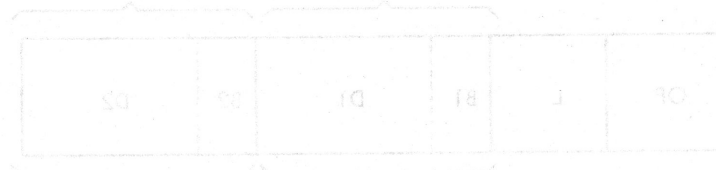
(Seite 237-247)



➤ Band 4, Seite 459 (Operationen für Dualdaten A bis LA)

KAPITEL V

DIE ASSEMBLERANWEISUNGEN USING UND DROP



Lernstoff:

Die Assembleranweisung USING

Die Assembleranweisung DROP

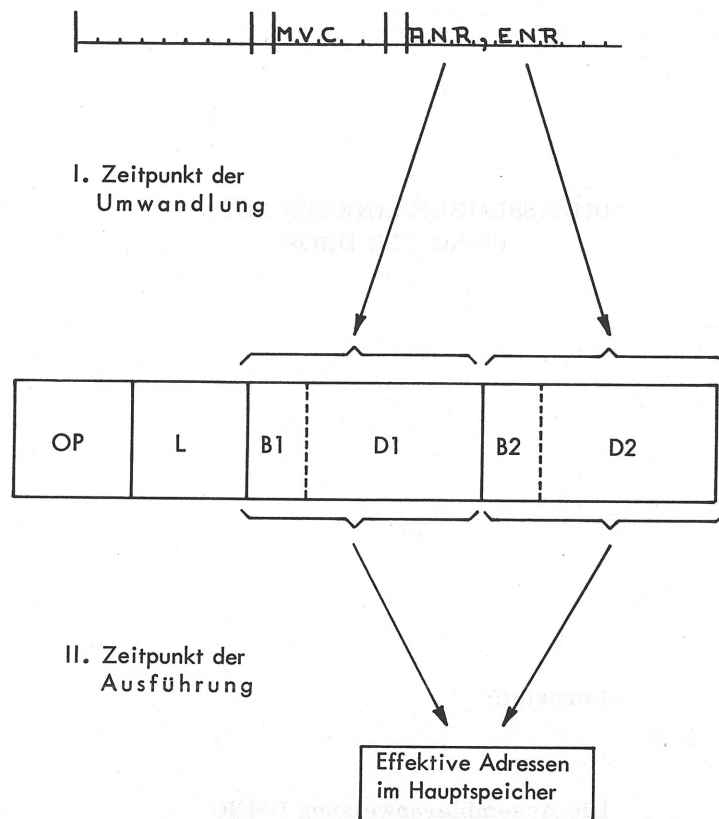


Abb. 1

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)

VERWENDEN BASISREGISTER

DATE RECEIVED 09283104 1974

S.I.N.G * 8

- Das Programm mehrmals benutzt werden.

- 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

| | U.S.I.N.G. | #, 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:

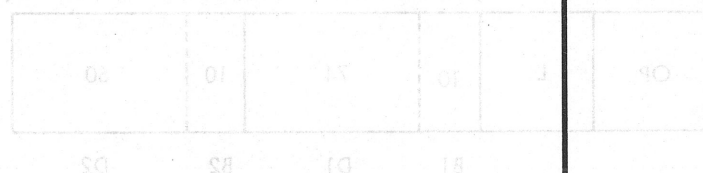
In der USING-Anweisung wird für die Umwandlung angegeben, welches Register als _____ benutzt werden soll und welcher Wert für die Errechnung der relativen Adressen als _____ zu verwenden ist.

Basisregister

Basisadresse

Beachten Sie bitte:

Register 0 darf nicht als Basisregister angegeben werden.



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



1081

D2

284

- 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:

800

802

806

811

1802

Name	Operation	Operand und Beme
1	8	10 14 20 25
	S.T.R.T.	8.0.0.
	↓	
	U.S.I.N.G.	*, 1.0.
	↓	
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

- 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 Zu-
ordnungszählers:

800

802

806

811

1802

Name	Operation	Operand und Beme
1	8	10 14
20	25	
	S.T.R.T.	8.0.0.
	↓	
	U.S.I.N.G.	*, 1.0.
	↓	
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

relative Adresse	Basisregister
1000	STRT
1001	USING
1002	CL
1003	CL
1004	CL
1005	CL
1006	CL
1007	CL
1008	CL
1009	CL
1010	CL
1011	CL
1012	CL
1013	CL
1014	CL
1015	CL
1016	CL
1017	CL
1018	CL
1019	CL
1020	CL
1021	CL
1022	CL
1023	CL
1024	CL
1025	CL
1026	CL
1027	CL
1028	CL
1029	CL
1030	CL
1031	CL
1032	CL
1033	CL
1034	CL
1035	CL
1036	CL
1037	CL
1038	CL
1039	CL
1040	CL
1041	CL
1042	CL
1043	CL
1044	CL
1045	CL
1046	CL
1047	CL
1048	CL
1049	CL
1050	CL
1051	CL
1052	CL
1053	CL
1054	CL
1055	CL
1056	CL
1057	CL
1058	CL
1059	CL
1060	CL
1061	CL
1062	CL
1063	CL
1064	CL
1065	CL
1066	CL
1067	CL
1068	CL
1069	CL
1070	CL
1071	CL
1072	CL
1073	CL
1074	CL
1075	CL
1076	CL
1077	CL
1078	CL
1079	CL
1080	CL
1081	CL
1082	CL
1083	CL
1084	CL
1085	CL
1086	CL
1087	CL
1088	CL
1089	CL
1090	CL
1091	CL
1092	CL
1093	CL
1094	CL
1095	CL
1096	CL
1097	CL
1098	CL
1099	CL
1100	CL
1101	CL
1102	CL
1103	CL
1104	CL
1105	CL
1106	CL
1107	CL
1108	CL
1109	CL
1110	CL
1111	CL
1112	CL
1113	CL
1114	CL
1115	CL
1116	CL
1117	CL
1118	CL
1119	CL
1120	CL
1121	CL
1122	CL
1123	CL
1124	CL
1125	CL
1126	CL
1127	CL
1128	CL
1129	CL
1130	CL
1131	CL
1132	CL
1133	CL
1134	CL
1135	CL
1136	CL
1137	CL
1138	CL
1139	CL
1140	CL
1141	CL
1142	CL
1143	CL
1144	CL
1145	CL
1146	CL
1147	CL
1148	CL
1149	CL
1150	CL
1151	CL
1152	CL
1153	CL
1154	CL
1155	CL
1156	CL
1157	CL
1158	CL
1159	CL
1160	CL
1161	CL
1162	CL
1163	CL
1164	CL
1165	CL
1166	CL
1167	CL
1168	CL
1169	CL
1170	CL
1171	CL
1172	CL
1173	CL
1174	CL
1175	CL
1176	CL
1177	CL
1178	CL
1179	CL
1180	CL
1181	CL
1182	CL
1183	CL
1184	CL
1185	CL
1186	CL
1187	CL
1188	CL
1189	CL
1190	CL
1191	CL
1192	CL
1193	CL
1194	CL
1195	CL
1196	CL
1197	CL
1198	CL
1199	CL
1200	CL

Stand des Zu-
ordnungszählers:

1048

1050

1065

1069

1075

1115

1120

1126

Name	Operation	Operand und Bemerkung
1	8	10 14 20 25
	START	1048
	↓	
	U.S.I.N.G	*.8
	(	
	↓	
E.B.E.R.	D.S.	0.C.L.5.0.
	D.S.	C.L.4.
E.N.R.	D.S.	C.L.6.
E.N.R.M.E.	D.S.	C.L.1.0.
	↓	
R.B.E.R.	D.S.	0.C.L.3.0.
	D.S.	C.L.5.
R.N.R.	D.S.	C.L.6.
R.N.R.M.E.	D.S.	C.L.1.0.
	(	
	↓	
	M.V.C.	R.N.R., E.N.R.
	M.V.C.	R.N.R.M.E., E.N.R.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. |-----| H.N.R., E.N.R. |-----|

OP	L	?	?	?	?
----	---	---	---	---	---

b)

.....	M.V.C.	R.NAME, E.NAME
-------	--------	----------------

OP	L	?	?	?	?
----	---	---	---	---	---

Überprüfen Sie nun bitte Ihre Ergebnisse auf der folgenden Seite !

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 nicht 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 laden.

Das Laden 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)

1.

Name	Operation	Oper
1	8	10
2	10	14
3	12	18
4	14	22
5	16	26
6	18	30
7	20	34
8	22	38
9	24	42
10	26	46
11	28	50
12	30	54
13	32	58
14	34	62
15	36	66
16	38	70
17	40	74
18	42	78
19	44	82
20	46	86
21	48	90
22	50	94
23	52	98
24	54	102
25	56	106
26	58	110
27	60	114
28	62	118
29	64	122
30	66	126
31	68	130
32	70	134
33	72	138
34	74	142
35	76	146
36	78	150
37	80	154
38	82	158
39	84	162
40	86	166
41	88	170
42	90	174
43	92	178
44	94	182
45	96	186
46	98	190
47	100	194
48	102	198
49	104	202
50	106	206
51	108	210
52	110	214
53	112	218
54	114	222
55	116	226
56	118	230
57	120	234
58	122	238
59	124	242
60	126	246
61	128	250
62	130	254
63	132	258
64	134	262
65	136	266
66	138	270
67	140	274
68	142	278
69	144	282
70	146	286
71	148	290
72	150	294
73	152	298
74	154	302
75	156	306
76	158	310
77	160	314
78	162	318
79	164	322
80	166	326
81	168	330
82	170	334
83	172	338
84	174	342
85	176	346
86	178	350
87	180	354
88	182	358
89	184	362
90	186	366
91	188	370
92	190	374
93	192	378
94	194	382
95	196	386
96	198	390
97	200	394
98	202	398
99	204	402
100	206	406
101	208	410
102	210	414
103	212	418
104	214	422
105	216	426
106	218	430
107	220	434
108	222	438
109	224	442
110	226	446
111	228	450
112	230	454
113	232	458
114	234	462
115	236	466
116	238	470
117	240	474
118	242	478
119	244	482
120	246	486
121	248	490
122	250	494
123	252	498
124	254	502
125	256	506
126	258	510
127	260	514
128	262	518
129	264	522
130	266	526
131	268	530
132	270	534
133	272	538
134	274	542
135	276	546
136	278	550
137	280	554
138	282	558
139	284	562
140	286	566
141	288	570
142	290	574
143	292	578
144	294	582
145	296	586
146	298	590
147	300	594
148	302	598
149	304	602
150	306	606
151	308	610
152	310	614
153	312	618
154	314	622
155	316	626
156	318	630
157	320	634
158	322	638
159	324	642
160	326	646
161	328	650
162	330	654
163	332	658
164	334	662
165	336	666
166	338	670
167	340	674
168	342	678
169	344	682
170	346	686
171	348	690
172	350	694
173	352	698
174	354	702
175	356	706
176	358	710
177	360	714
178	362	718
179	364	722
180	366	726
181	368	730
182	370	734
183	372	738
184	374	742
185	376	746
186	378	750
187	380	754
188	382	758
189	384	762
190	386	766
191	388	770
192	390	774
193	392	778
194	394	782
195	396	786
196	398	790
197	400	794
198	402	798
199	404	802
200	406	806
201	408	810
202	410	814
203	412	818
204	414	822
205	416	826
206	418	830
207	420	834
208	422	838
209	424	842
210	426	846
211	428	850
212	430	854
213	432	858
214	434	862
215	436	866
216	438	870
217	440	874
218	442	878
219	444	882
220	446	886
221	448	890
222	450	894
223	452	898
224	454	902
225	456	906
226	458	910
227	460	914
228	462	918
229	464	922
230	466	926
231	468	930
232	470	934
233	472	938
234	474	942
235	476	946
236	478	950
237	480	954
238	482	958
239	484	962
240	486	966
241	488	970
242	490	974
243	492	978
244	494	982
245	496	986
246	498	990
247	500	994
248	502	998
249	504	1002
250	506	1006
251	508	1010
252	510	1014
253	512	1018
254	514	1022
255	516	1026
256	518	1030
257	520	1034
258	522	1038
259	524	1042
260	526	1046
261	528	1050
262	530	1054
263	532	1058
264	534	1062
265	536	1066
266	538	1070
267	540	1074
268	542	1078
269	544	1082
270	546	1086
271	548	1090
272	550	1094
273	552	1098
274	554	1102
275	556	1106
276	558	1110
277	560	1114
278	562	1118
279	564	1122
280	566	1126
281	568	1130
282	570	1134
283	572	1138
284	574	1142
285	576	1146
286	578	1150
287	580	1154
288	582	1158
289	584	1162
290	586	1166
291	588	1170
292	590	1174
293	592	1178
294	594	1182
295	596	1186
296	598	1190
297	600	1194
298	602	1198
299	604	1202
300	606	1206
301	608	1210
302	610	1214
303	612	1218
304	614	1222
305	616	1226
306	618	1230
307	620	1234
308	622	1238
309	624	1242
310	626	1246
311	628	1250
312	630	1254
313	632	1258
314	634	1262
315	636	1266
316	638	1270
317	640	1274
318	642	1278
319	644	1282
320	646	1286
321	648	1290
322	650	1294
323	652	1298
324	654	1302
325	656	1306
326	658	1310
327	660	1314
328	662	1318
329	664	1322
330	666	1326
331	668	1330
332	670	1334
333	672	1338
334	674	1342
335	676	1346
336	678	1350
337	680	1354
338	682	1358
339	684	1362
340	686	1366
341	688	1370
342	690	1374
343	692	1378
344	694	1382
345	696	1386
346	698	1390
347	700	1394
348	702	1398
349	704	1402
350	706	1406
351	708	1410
352	710	1414
353	712	1418
354	714	1422
355	716	1426
356	718	1430
357	720	1434
358	722	1438
359	724	1442
360	726	1446
361	728	1450
362	730	1454
363	732	1458
364	734	1462
365	736	1466
366	738	1470
367	740	1474
368	742	1478
369	744	1482
370	746	1486
371	748	1490
372	750	1494
373	752	1498
374	754	1502
375	756	1506
376	758	1510
377	760	1514
378	762	1518
379	764	1522
380	766	1526
381	768	1530
382	770	1534
383	772	1538
384	774	1542
385	776	1546
386	778	1550
387	780	1554
388	782	1558
389	784	1562
390	786	1566
391	788	1570
392	790	1574
393	792	1578
394	794	1582
395	796	1586
396	798	1590
397	800	1594

DROP

AUFGEBEN BASISREGISTER (Drop Base Address Register)

Beispiel:

Name								Operation				Of							
1								10				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

AUFGABEN BASISREGISTER

(Drop Base Address Register)

Beispiel:

Name		Operation		Order
1	8	10	14	20
		D.R.O.P.		8

1. Der symbolische Operationsschlüssel für die Assembleranweisung "Aufgaben 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.

- 6 Wie muß die Assembler-Anweisung lauten, wenn in einem Programm fortan das Basisregister 10 nicht mehr zur Adreßbildung benutzt werden soll ?

- 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.

unbenutzt

beliebigen

Name				Operation				Op			
1	8	10	14	16	20	24	28	32	36	40	44
				DROP				10			

DROP

14

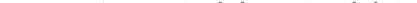
Basisregister

USING

FRAGEN ZUM KAPITEL V

Prüfen Sie nun bitte, ob Sie den Lernstoff dieses Kapitels beherrschen. Beantworten Sie erst alle Fragen auf einem Extrablatt bzw. Programmschema, bevor Sie Ihre Antworten überprüfen.

1. Der symbolische Operationsschlüssel für die Assembleranweisung "Verwenden Basisregister" lautet _____.

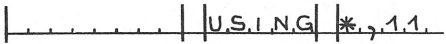
2. 

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. 
(Seite 277 - 293)
5. Ein durch USING eingesetztes Basisregister kann mit Hilfe einer DROP-Anweisung wieder aufgegeben werden.
(Seite 298)



KAPITEL VI

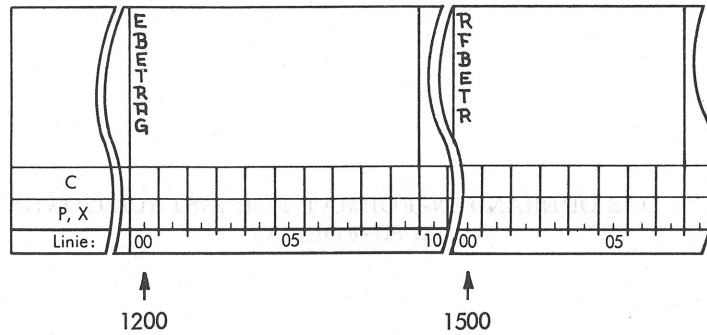
DAS OPERANDEN-FORMAT FÜR INSTRUKTIONEN IM SS-FORMAT

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
		R.P.		RFBTR, EBTRAG			
		R.P.		RFBTR, EBTRAG+4.(6.)			

Basisregister: 11

Basisadresse: 1000

FA	?	?	?	?	?	?
OP	L1	L2	B1	D1	B2	D2

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

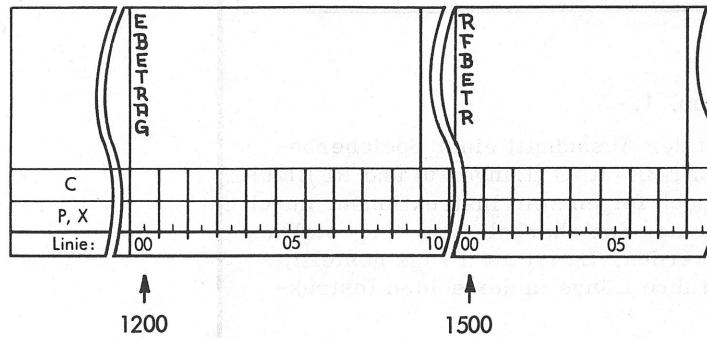
Zusammengesetzte Ausdrücke sind Kombinationen von Symbolen und Direktwerten, die miteinander arithmetisch verknüpft sind (Adressrechnung).

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	
		R.P.		RFBETR, EBETRAG				
		R.P.		RFBETR, EBETRAG + 4. (b.)				

Basisregister: 11

Basisadresse: 1000

FA	?	?	?	?	?	?
OP	L1	L2	B1	D1	B2	D2

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)

FA	7	9	11	500	11	200
----	---	---	----	-----	----	-----

b)

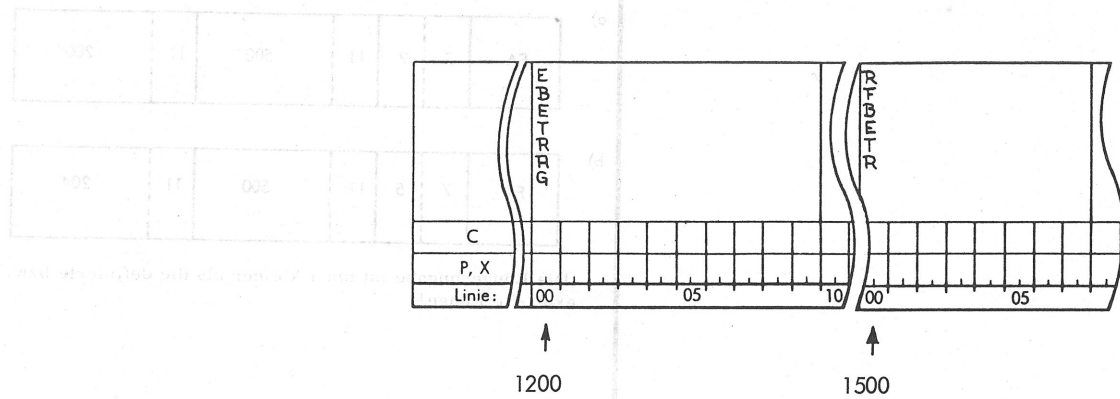
FA	7	5	11	500	11	204
----	---	---	----	-----	----	-----

(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 Adresse eines Operanden explizit 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

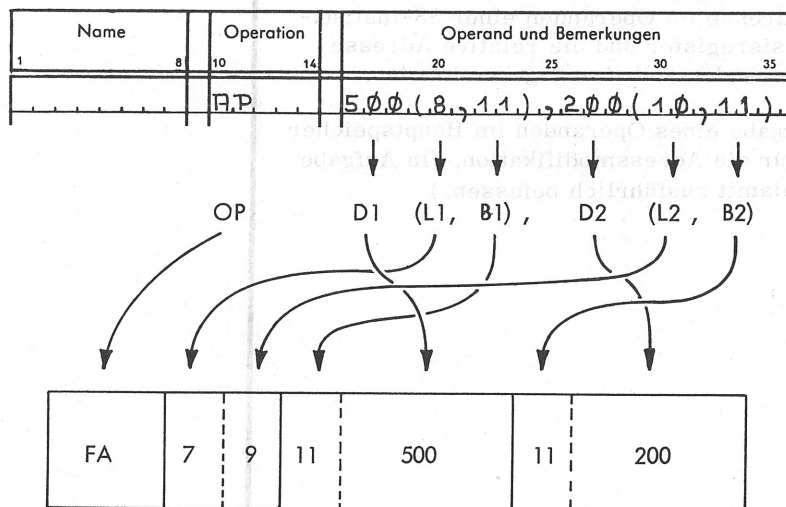


Abb. 2

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

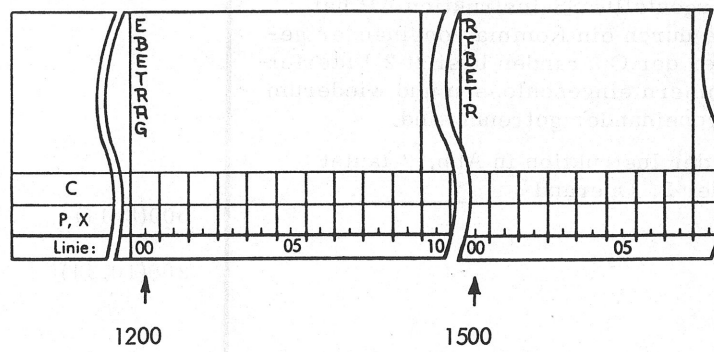
Länge

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

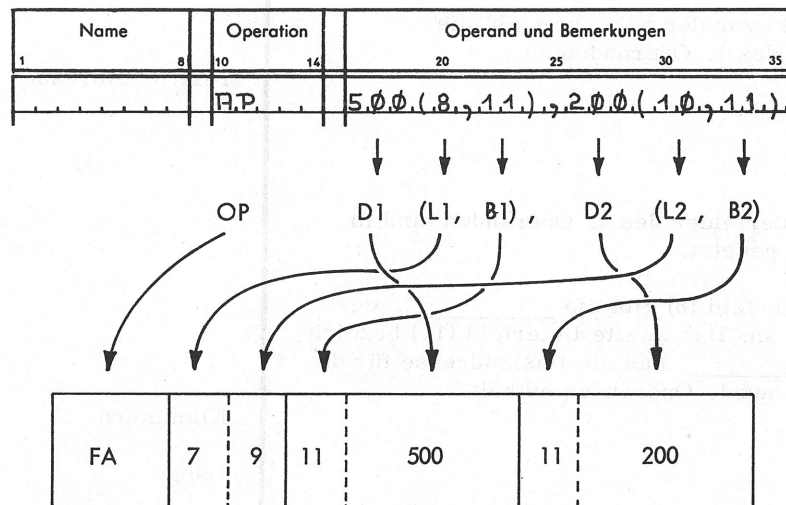


Abb. 2

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 ?

P.A.C.K. | 10.(6.,8.),45.(10.,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:

P.A.C.K. | 10.(6.,8.),45.(10.,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:

S1
S1(L1)
D1(L1,B1)

2. Operand:

S2
S2(L2)
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		Operation		Operand und Bemerkung	
1	8	10	14	20	25
.....		P.R.C.K.		R.E.F.E., 0 (7, 1, 1)	

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:

S1
S1(L1)
D1(L1,B1)

2. Operand:

S2
S2(L2)
D2(L2,B2)

Logische Operationen

1. Operand:

S1
S1(L)
D1(L,B1)

2. Operand:

S2
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

Operation	Operation	Operation
ADD (S1, S2)	REF (S1, B1)	REF (S1, B1)

15

$$O(7, 11)$$

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 eine 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), 15.(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:

S1
S1(L1)
D1(L1,B1)

2. Operand:

S2
S2(L2)
D2(L2,B2)

Logische Operationen

1. Operand:

S1
S1(L)
D1(L,B1)

2. Operand:

S2
L
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

1. Operand (S1)	2. Operand (S2)	Operation (O)	Basisadresse (B)

19

Name		Operation		Operand und Bemerkungen		
1	8	10	14	20	25	30
		M.V.C.		80.(5,8),	45.(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 arithmetischen 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)

$$D1(L1, B1) \text{ (bzw.) } D2(L2, B2)$$

- 21 Bei logischen 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 _____.

Auch bei den Instruktionen aller anderen Formate können die Operanden im Hauptspeicher explizit angegeben werden. Weiteres hierzu erfahren Sie im Folgelehrgang.

S1

S1(L)

D1(L, B1)

S2

D2(B2)

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 Ausdrucks.

(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

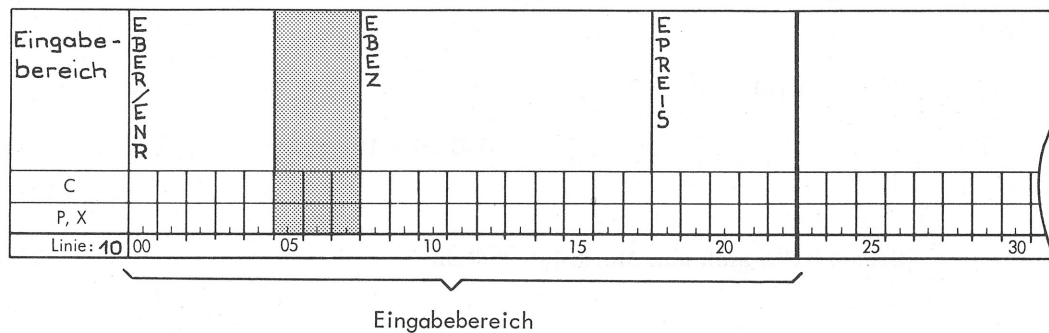


Abb. 1

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
		START		1000

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

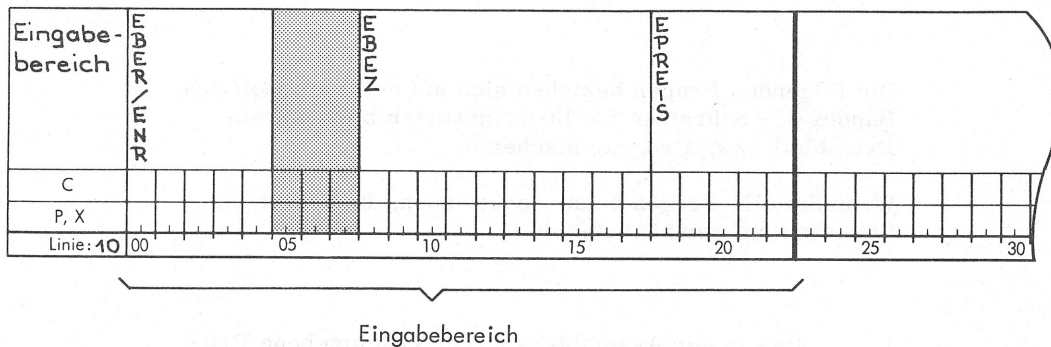


Abb. 1

IBM System / 360 Speicherbelegung

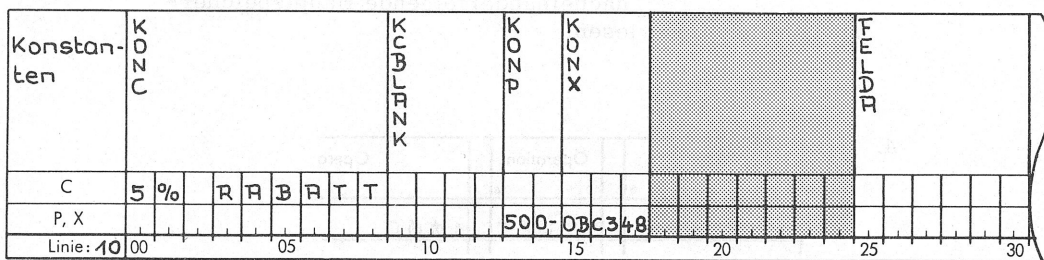


Abb. 2

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) zurück-schlagen und diese durcharbeiten. Kommt es auch dort zu Fehlern, dann wiederholen Sie die betreffenden Lern-elemente. - 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	Operation	Operand
1 8	10 14	20
EBER	DS	0CL23
ENR	DS	CL5
	DS	CL3
EBEZ	DS	CL10
EPREIS	DS	CL5

(Fragen zum Kapitel III, Seite 177-179)

6.

Name	Operation	Operand und Be
1 8	10 14	20 25
:	:	
	ORG	EBER+30
oder :	ORG	ENR+30
oder :	ORG	EBEZ+22
oder :	ORG	EPREIS+12
oder :	ORG	*+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		Operation		Operand und Bemerku	
1	8	10	14	20	25
a)	K.O.N.C.	D.C.		C' 5% RABATT'	
	K.C.B.L.F.N.K.	D.C.		C.L.4' '	
	oder:				
	K.C.B.L.F.N.K.	D.C.		C' ' '	
	K.O.N.P.	D.C.		P' - 500'	
	K.O.N.X.	D.C.		X' BC 348'	
b)	H.D.K.O.N.	D.C.		R.(F.E.L.D.R.)	

(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)