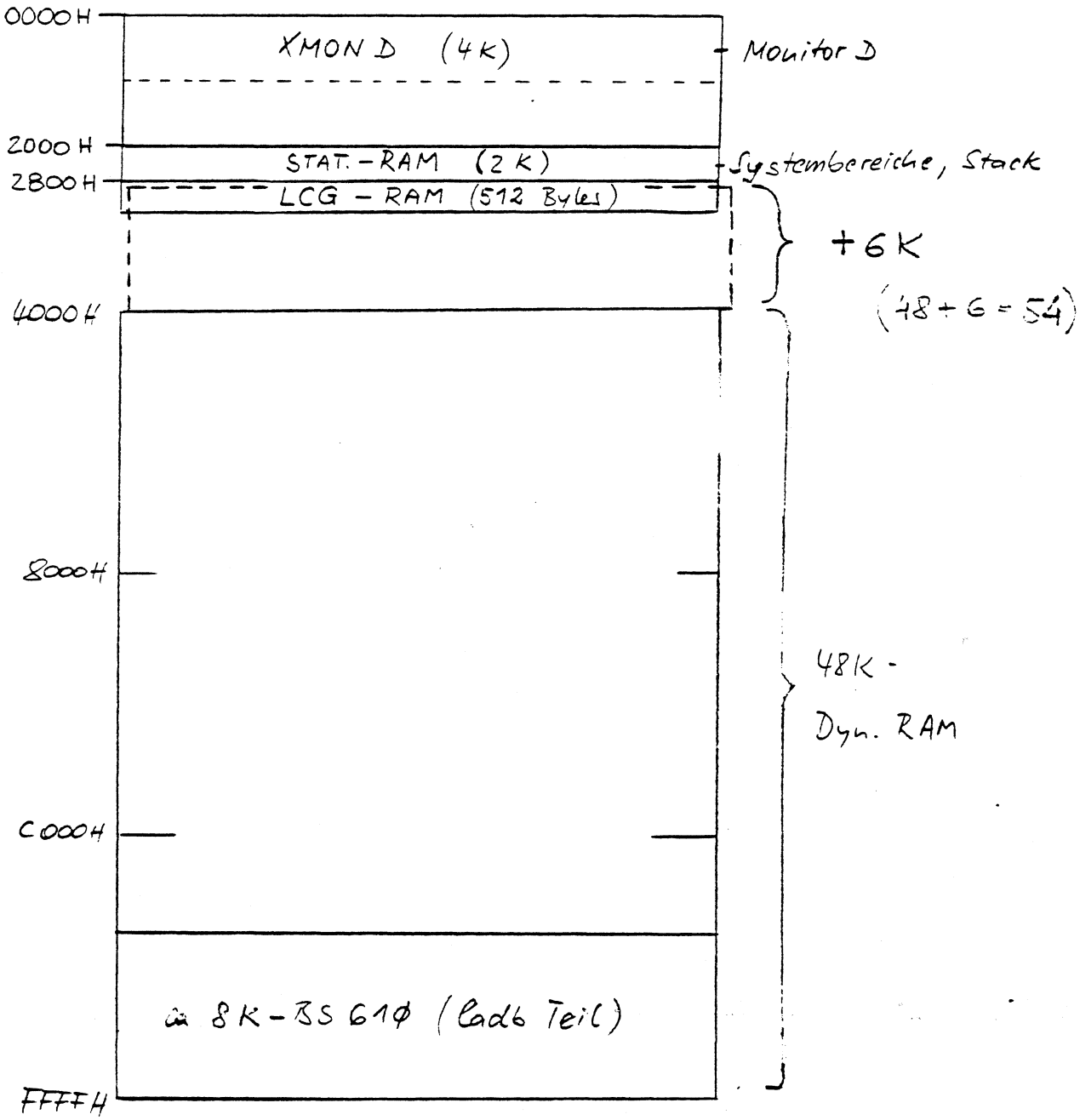
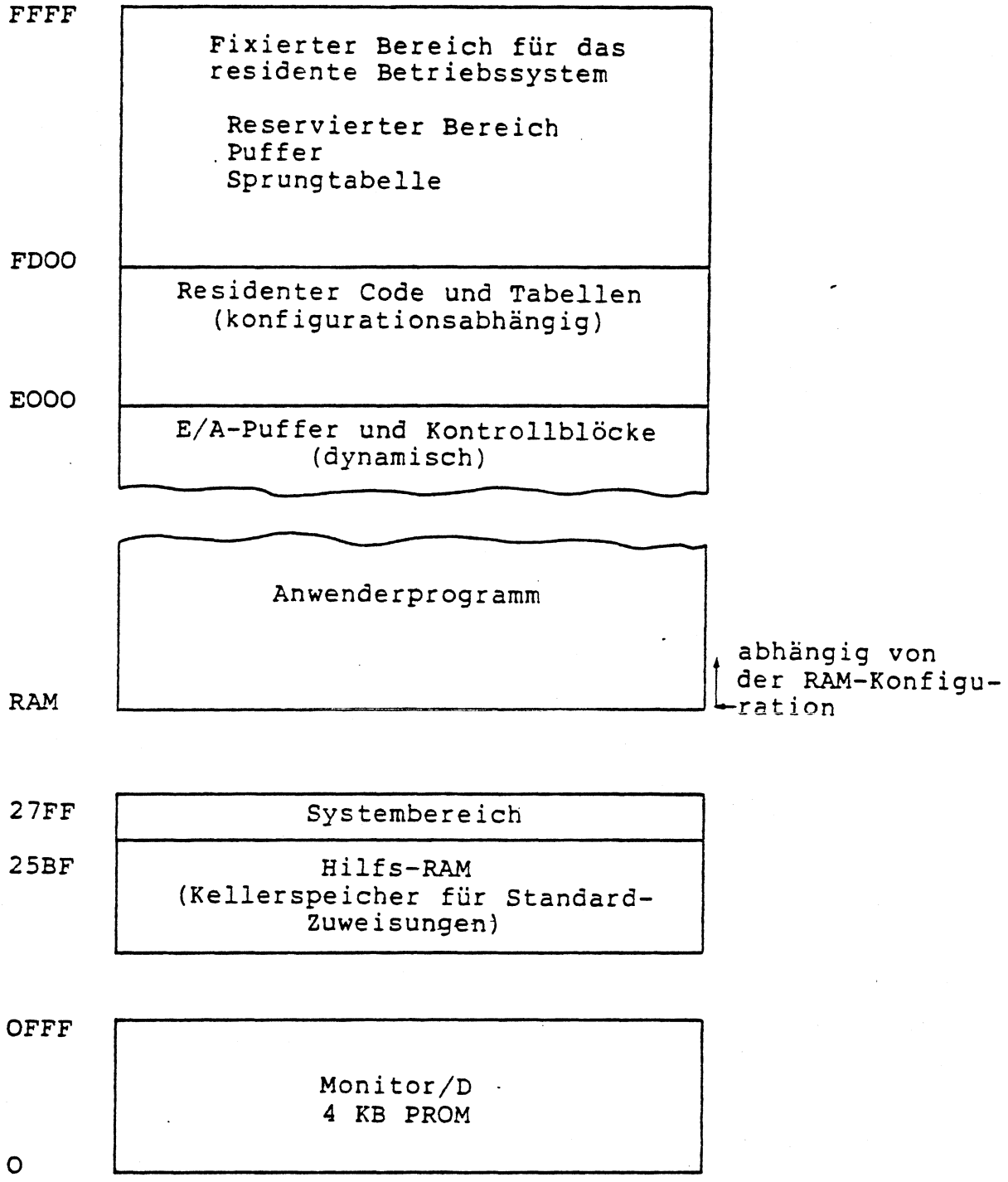
INHALTSVERZEICHNIS

SBB 6.610

Speicherbelegung 6.610 - 48K-RAM

BS 610/V 1.62





Figur 1 Speicherbelegung

Systembereich

Der Bereich von FD00 bis FFFF ist reserviert für Systembenutzung.

Der Bereich von FF00 - FF0F enthält die augenblicklichen Datei- bzw. Gerätezuweisungen für die acht logischen Einheiten:

FF00	CI
FF02	CO
FF04	SI
FF06	SO
FF08	SL
FF0A	AI
FF0C	AO
FF0E	AL

Zwischen FD00 und FDFF liegen eine Sprungtabelle für Systemfunktionen und die Systemzeiger.

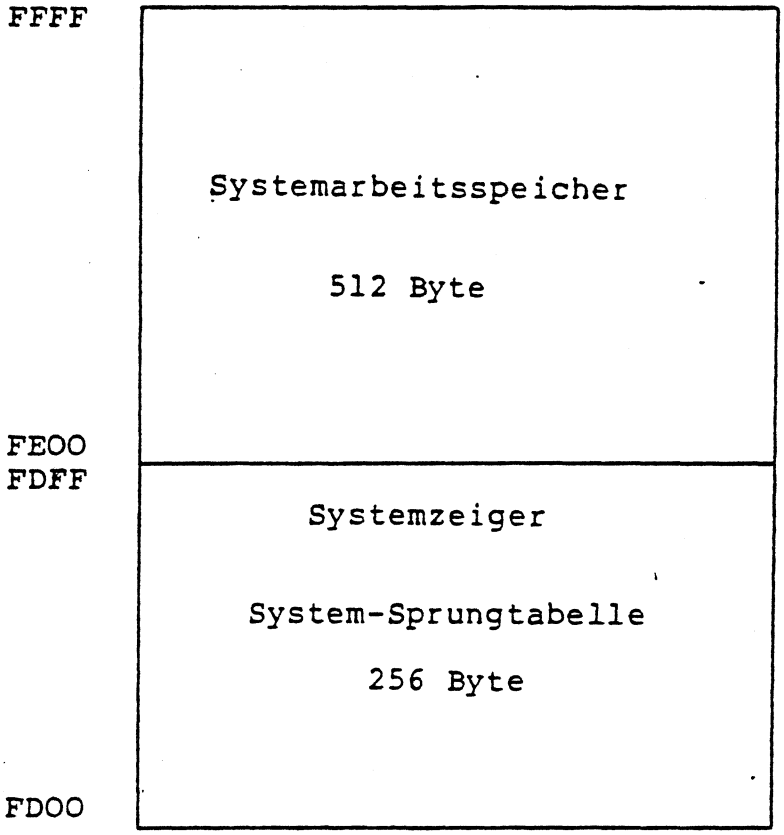
FD00 - FD3F	Vom Anwender benutzbare BS 610-Funktionen
FD40 - FDCC	Sprungtabelle für Grundfunktionen
FDCD - FDEB	Interne Systemfunktionen
FDEC - FDFF	Systemzeiger

Die Adressen der internen Systemfunktionen und der Systemzeiger sind im Anhang C dargestellt. Diese Funktionen dienen ausschließlich der Systemprogrammierung.

Folgende Funktionen des BS 610 sind benutzbar:

FD00	BS 610	FD03	ERROR	FD06	LOAD
FD09	INCHAR	FD0C	OUTCHAR	FD0F	GET
FD12	PUT	FD15	PUTSTR	FD18	GETBIN
FD1B	PUTBIN	FD1E	OPEN	FD21	READ
FD24	WRITE	FD27	CLOSE	FD2A	EDBH
FD2D	EDWH	FD30	EDHB	FD33	EDHW
FD36	BUFALLOC	FD39	BUFDEALLOC	FD3C	STIMER

Die Sprungtabelle der Grundfunktionen hat dieselbe Struktur wie die des Monitors. Bei Systemen, deren Monitor für Diskette und MB-Kassette ausgelegt ist, verweist die Sprungtabelle einfach auf die relevanten Adressen der Monitor-Sprungtabelle. Bei Systemen, deren Monitor nur einen Gerätetyp unterstützt, können die Routinen des anderen Gerätetyps im RAM-Bereich des BS 610 abgelegt werden. Der Aufruf der Grundfunktionen über die BS 610-Sprungtabelle (nicht direkt über den Monitor) stellt sicher, daß Programme auch bei anderen Monitoren ohne Änderung ablauffähig bleiben.



Figur 2 Reservierter Systembereich

Unterbrechungsbehandlung durch den Anwender im BS 610

Beim Auftreten einer Unterbrechung arbeitet der Monitor eine feststehende Sequenz von Befehlen ab. Bei Unterbrechungen der Priorität 2, 4 und 5 kann der Anwender nach dem Ablauf der erwähnten Standardsequenz die Kontrolle übernehmen, wenn er die Adresse seiner Unterbrechungsroutine in bestimmten Speicherstellen ablegt.

Priorität	Unterbrechung durch	Adresse der Unterbrechungsroutine in
2	Sync. Interface	027A4 H
4	Cluster-Interface I	027A8 H
5	Cluster-Interface II	027AA H

Die angegebenen Felder werden beim Einschalten der Netzspannung auf 0 gesetzt. Sind sie = 0, dann wird beim Auftreten einer Unterbrechung in die angegebene Routine verzweigt. Der Anwender ist für das Retten und Wiederherstellen aller Register verantwortlich. Außerdem muß er eine EI-Anweisung (Enable interrupt) geben, ehe er seine Unterbrechungsroutine mit RET verläßt.

Beispiel: Ein (Assembler) Programm soll Unterbrechungen der Priorität 4 abfangen. Die Anwender-Unterbrechungsroutine habe den Namen INTR4.
Folgende Anweisungen sind notwendig:

```
LXI     H,INTR4
SHLD    027A8H
```

Alle Unterbrechungen der Priorität 4 werden nach Ablauf dieser 2 Befehle durch INTR4 behandelt.

Die durch den Taktgeber alle 20 ms mit Prior. 1 erzeugten Unterbrechungen sind dem Anwender nicht direkt zugänglich. Es existiert jedoch eine Systemroutine (STIMER), die es ermöglicht, nach einer bestimmten Anzahl von 20 ms-Takten eine Unterbrechung - mit Hilfe des BS 610 - zu erzeugen.

Falls nur mit dem Monitor gearbeitet wird, muß die Adresse der Unterbrechungsroutine in 27B0H, das Zeitintervall bis zur Unterbrechung als Wort in 27BDH abgespeichert werden. Während der Modifikation der beiden Speicherstellen dürfen keine Unterbrechungen auftreten (DI-Anweisung verwenden!).

Unterbrechungen der Priorität 3 können vom Anwender nur behandelt werden, wenn sie den UART betreffen und die Unterbrechungsursache nicht "data available" (Daten verfügbar) oder "transmitter buffer empty" (Sendepuffer leer) ist. Die Adresse der Unterbrechungsroutine muß in 27A6H gespeichert sein.

Unterbrechungen der Priorität 6 (Diskette) und 7 (MB-Kassette) werden generell durch den Monitor behandelt. Falls für derartige Unterbrechungen eine Behandlung durch den Anwender vorgesehen ist (Adresse gesetzt!), wird die Kontrolle nur nach Ablauf einer vollständigen Operation (z.B. READ oder WRITE) nicht aber nach einem SEEK oder RETRY übergeben.

Unterbrechungen der Priorität 0 können nicht vom Anwender abgefangen werden.

Anhang C

Interne Systemfunktionen

Die folgenden Adressen geben den Einsprung an für bestimmte interne Systemfunktionen und für Tabellen zur Dateiverarbeitung.

Einsprungadressen:

FDCD	DATEINAME
FDDO	IBFIL
FDD3	RELN
FDD6	RELBLK
FDD9	GETN
FDDC	GETBLK

Tabellenadressen:

Die Adressen beziehen sich auf 18 Byte lange Sprungtabellen mit 6 Einträgen zur Verarbeitung von Dateien in verschiedenen Formaten. Aufbau der Sprungtabellen:

0	Eröffnen einer Eingabedatei
3	Eröffnen einer Ausgabedatei
6	Lesen eines Blockes
9	Schreiben eines Blockes
0C	Schließen einer Eingabedatei
0F	Schließen einer Ausgabedatei
FDEA	Tabelle für Intel-formatierte Diskette
FDE8	" " IBM - " "
FDE6	" " BS 610-formatierte MB-Kassette

Die folgenden 4 Routinen setzen voraus, daß die Gerätenummer in UNIT (2730) gespeichert ist, und daß MAP1 und MAP2 (2739,273B) Spur- und Sektorennummer des Belegungsverzeichnisses der im Gerät befindlichen Diskette enthalten.

FDD3	RELN	Freigeben einer Anzahl von Diskettensektoren A - Anzahl HL - Adresse einer Liste von Sektoren
FDD6	RELBLK	Freigabe eines Diskettensektors BC - Spur/Sektor
FDD9	GETN	Reservierung einer Anzahl von Diskettensektoren A - Anzahl

HL - gewünschte Adresse des ersten Sektors

FDDC GETBLK Reservierung eines Disketten-sektors
BC - Spur/Sektor

Lage der Systemzeiger

	+0 +1	+2 +3	+4 +5	+6 +7
FDF8	EBCDIC	ASCII	RAM MEM START	SYSTEM START
FDF0	:NF:	:BB:	:CO:	:CI:
FDE8				:PR:

RAM MEM START: Beginn des RAM-Speichers

Druckerbehandlung im Betriebssystem BS 1

An den Bildschirm-Computer 6.610 können verschiedene Drucker angeschlossen werden. Die Drucker verlangen jedoch eine unterschiedliche Ansteuerung durch das Betriebssystem.

Um zu verhindern, daß sämtliche Gerätetreiber für alle denkbaren Drucker haupspeicherresident sein müssen, wird vom BS 1 nur ein vorgebbarer Treiber nachgeladen.

Dies geschieht dadurch, daß vom Betriebssystem eine Treibervorspann-Routine geladen wird, die unter dem Namen DRUCKR auf der Floppy Disk des Betriebssystems gespeichert ist.

Die Vorspannroutine ergänzt die Systemlademeldung um die Bezeichnung des Druckertreibers und führt einige Plausibilitätskontrollen aus. Anschließend lädt die Routine den eigentlichen Druckertreiber.

Ohne die Anwesenheit des Programmes DRUCKR und der Treiberroutine kann das Betriebssystem also nicht geladen werden.

Zur Zeit stehen die folgenden Treiber zur Verfügung:

Drucker	Treiber-Routine	Treiber-Vorspann-Routine	
80-spaltiger PT 80 Matrixdrucker	DM22.80	MA22.80	A
132-spaltiger PT80	DM22.132	MA22.132	B
PT80 mit Sparschrift	DM22.S	MA22.S	C
BINDER Matrixdrucker	DMBIND	MABIND	D
DIABLO Typenraddrucker mit deutschem Zeichenvorrat	DT24.DEU	TY24.DEU	F
DIABLO Typenraddrucker mit internationalem Zeichenvorrat	DT24.INT	TY24.INT	E
LOGABAX Matrixdrucker	DM25	MA25	G

Bildschirmattribute

Die Bildschirmsteuerung erlaubt 6 verschiedene Darstellungsarten der Zeichen auf dem Bildschirm. Die Darstellung muß über ein Attributbyte gesteuert werden.

Prinzipieller Textaufbau:

Attributbyte T E X T Attributbyte

Aufbau des Attributbytes

7	6	5	4	3	2	1	0
2	2	2	2	2	2	2	2
1	x	x	x	y	y	y	y

1 = Attribut nicht ausgewertet

Kennzeichen für Attributbytes

xxx	Bedeutung
000	nicht verwendet
001	nicht verwendet
010	Halbhelle Darstellung
011	Blinken
100	Invertierte darstellung
101	Unterstrichene Darstellung
110	Dunkel (unsichtbar)
111	Normale Darstellung

Daraus ergibt sich die folgende Hex-darstellung:

A0 = Halbhell
B0 = Blinken
C0 = Invers
D0 = Unterstrichen
E0 = Dunkel
F0 = Normal

Ein Attribut bleibt solange gültig, bis es durch ein neues aufgehoben wird.

Dateiverwaltung

ECMA TC 19	INTEL
Datei muß vorher eingerichtet werden	Datei ~~muß~~ wird automatisch eingerichtet.
Dateigröße muß vorher bekannt sein.	
1 Datensatz pro Sektor	Keine Sektorgrenzen (für Anwender)
max 14 Dateien wählbare	1/6 für den Anwender.
statische Floppyverwaltung	dynamische Floppyverwaltung
schneller in der Anwendung	etwas langsamer

INTEL–DISKETTENFORMAT

Eine Diskette, die 6.61Ø Dateien (Intel-Format) beinhalten soll, muß mit den Programmen

INIT $:In: (Formatieren im ECMA TC19 Format)

und

FORMAT $:Fn:Datenträgername

eingerichtet werden.

Das Programm FORMAT legt 4 Dateien für die Verwaltung der Diskette an:

ISIS.LAB = Datenträger Kennsatz
ISIS.DIR = Inhaltsverzeichnis
ISIS.MAP = Belegungsverzeichnis
ISIS.ERR = Fehlerbeschreibung im Klartext

Die Datei ISIS.LAB enthält den Datenträger-Kennsatz. Dieser dient zur Kennzeichnung der Diskette für den Anwender. Der Name wird bei jedem DIR-Kommando mit ausgegeben.

Aufbau:

Byte Ø 1 2 3 4 5 6 7 8

B	I	B	L	I	O	V	Ø	1

Spur Ø
Sektor 26

Kommando zum Einrichten

FORMAT $:Fn:BIBLIO.VØ1

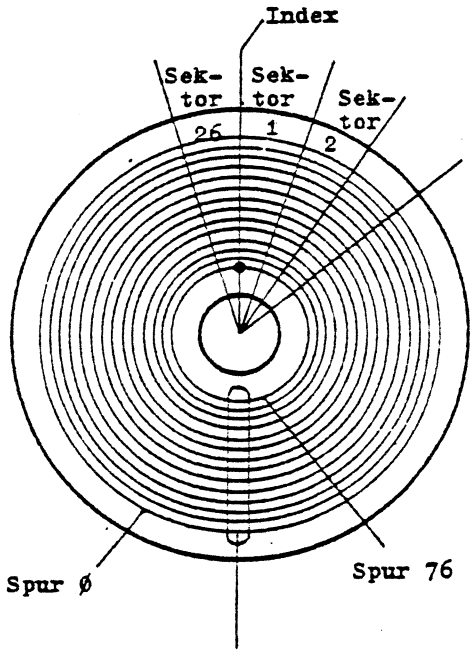
Spur Ø, Sektor 1-24 = ISIS.ERR

```
 DISK-EDIT  REV.:051  FROM:00/01 (HEX) TO:00/19 (HEX) PAGE:0001

  DRIVE: 1 TRACK: 00 (HEX) SECTOR: 01 (HEX) CODE:HX/AS
       * * * * * * * *  * * * E N D   O F   F I L E * *  * * * * * * * *
000:852A2A2A2A2A2A2A 2A2A2A454E44204F 462046494C452A2A 2A2A2A2A2A2A2A2A
       * * * * * * * *  * S E C T O R     M I S S I N G *  * * * * * * * *
020:852A2A2A2A2A2A2A 2A534543544F5220 4D495353494E472A 2A2A2A2A2A2A2A2A
       * * * * * * * *  * * * * C R C     E R R O R * * *  * * * * * * * *
040:852A2A2A2A2A2A2A 2A2A2A2A43524320 4552524F522A2A2A 2A2A2A2A2A2A2A2A
     T I M I N G     E R R O R     R E   A D / W R I T E     X F E R * * *
060:8554494D494E4720 4552524F52205245 41442F5752495445 20584645522A2A2A

  DRIVE: 1 TRACK: 00 (HEX) SECTOR: 02 (HEX) CODE:HX/AS
       * * * * * * * *  * N O   A D D R   E S S   M A R K  * * * * * * * *
000:852A2A2A2A2A2A2A 2A4E4F2041444452 455353204D41524B 2A2A2A2A2A2A2A2A
       * * * D I S K   E T T E   O P E   R A T I O N   T   I M E O U T * *
020:852A2A2A44495534B 45545445204F5045 524154494F4E2054 494D454F55542A2A
       * T I M I N G     E R R O R   I N   R E A D   O P E R A T I O N
040:852A54494D494E47 204552524F522049 4E2052454144204F 5045524154494F4E
       * * * * * * * *  * B U S Y - B I   T   T I M E O U   T * * * * * * *
060:852A2A2A2A2A2A2A 2A42555359D4249 542054494D454F55 542A2A2A2A2A2A2A

  DRIVE: 1 TRACK: 00 (HEX) SECTOR: 03 (HEX) CODE:HX/AS
       * * * * * * * *  * D R I V E   N O T   R E A D Y  * * * * * * * *
000:852A2A2A2A2A2A2A 2A44524956452045 4F54205245414459 2A2A2A2A2A2A2A2A
       * * N A M E     N O T   I N   D I R   A N D   D I R   F U L L *
020:852A2A4E414D4520 4E4F5420494E2044 495220414E442044 49522046554C4C2A
       * * * * * N A   M E   N O T   I N   D I R E C T   O R Y * * * * *
040:852A2A2A2A2A4E41 4D45204E4F542049 4E20444952454354 4F52592A2A2A2A2A
       * * * * * P R O G R A M   N A M E   N O T   F O U N D * * * *
060:852A2A2A2A2A5052 4F4752414D204E41 4D45204E4F542046 4F554E442A2A2A2A

  DRIVE: 1 TRACK: 00 (HEX) SECTOR: 04 (HEX) CODE:HX/AS
       * * * * * I L   L E G A L   D I   R E C T O R Y     E N T R Y * * *
000:852A2A2A2A2A494C 4C4547414C204449 524543544F525920 454E5452592A2A2A
       * * D I R E C   T O R Y   B L O C K   E R R O R     -   L O A D *
020:852A2A44495524543 544F525920424C4F 434B204552524F52 202D204C4F41442A.
       * D A T A   L E N G T H   E R R O R   D U R I N G   L O A D *
040:852A44415441204C 454E4754482045 52524F522044555249 4E47204C4F41442A
       * * * * * * D I S K E T T E   M A P   M I S S I N G * * * * * *
060:852A2A2A2A2A2A44 49534B4554544520 4D4150204D495353 494E472A2A2A2A2A
```

SPUR Ø, SEKTOR 26 = ISIS.LAB

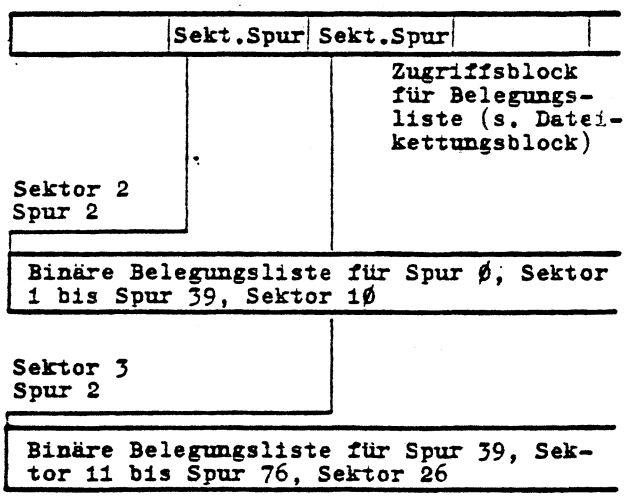
```
DISK-EDIT  REV.:051  FROM:00/1A (HEX) TO:00/1A (HEX) PAGE:0001

DRIVE: 1 TRACK: 00 (HEX) SECTOR: 1A (HEX) CODE:HX/AS
    K R S S Y S V 1  0
000:4B525353595356 31 0000000000000000 0000000000000000 0000000000000000
020:0000000000000000 0000000000000000 0000000000000000 0000000000000000
040:0000000000000000 0000000000000000 0000000000000000 0000000000000000
060:0000000000000000 0000000000000000 0000000000000000 0000000000000000
```

Eine Diskette (derzeitige Kapazität 256256 Byte) ist aufgeteilt in 77 Spuren á 26 Sektoren



Somit beinhaltet die Diskette insgesamt 2ØØ2 Sektoren (entspricht 2ØØ2 Blöcken). Je Block können max. 128 Byte auf die Diskette geschrieben werden.
Über eine separate Datei wird jeder einzelne Sektor (Block) vom BS61Ø verwaltet. Diese Datei heißt

ISIS.MAP

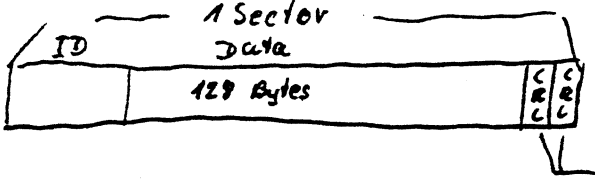


In der Binären Belegungsliste entspricht ein Bit = einem Sektor

Anmerkung:
Das BS61Ø arbeitet intern grundsätzlich Blockweise.

Single Side Density

77 Spuren
26 Sectoren
26 x 77 = 2002 sekt.



ISIS.MAP

```
Disk-Edit  Rev.:001  From:02/02 (HEX) To:02/03 (HEX) Page:0001
Drive: 1 Track: 02 (HEX) Sector: 02 (HEX) Code:HX/AS
000:FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF
020:FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF
040:FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF
060:FFFFFFFFFFFFFFFF FFFC000000000000 0000000000000000 0000000000000000
Drive: 1 Track: 02 (HEX) Sector: 03 (HEX) Code:HX/AS
000:0000000000000000 0000000000000000 0000000000000000 0000000000000000
020:0000000000000000 0000000000000000 0000000000000000 0000000000000000
040:0000000000000000 0000000000000000 0000000000000000 0000000000000000
060:0000000000000000 0000000000000000 0000000000000000 0000000000000000
```

F = 4 Sectoren

Die Datei ISIS.DIR enthält das Inhaltsverzeichnis (Buchhalter) für alle auf der Diskette vorhandenen Dateien. Der Buchhalter umfaßt alle 26 Sektoren der Spur 1. Für jede Datei existiert ein Eintrag von 16 Byte Länge; damit eines pro Diskette.

128	Byte/Sektor
: 16	Byte/Datei
8	Dateieinträge pro Sektor
x 25	Sektoren
200	Dateien
- 4	Dateien ISIS.
196	Dateien für den Benutzer möglich

Aufbau eines Eintrages:

Zustand	NAME	ERWEITERUNG	FLAG	A B	ANZAHL BLÖCKE-1	SEKTOR-Nr.	SPUR Nr.

Zustand: 00 = Datei vorhanden
FF = Datei gelöscht (DELETE-Kommando)
7F = Datei nicht vorhanden

Flag: 80 = F (ISIS.xxx von FORMAT)
08 = ALIAS (Programm ALIAS)
04 = W (Write-Protect)
02 = S (System-file) Programm ATTRIB
01 = I (Inhibit-DIR)
40 = X ISAM/DA - DATEI

AB : Anzahl Bytes im letzten Datenblock

Sektor Nr. / Spur Nr. : Sektor und Spurnummer des 1. Dateikettungsblockes

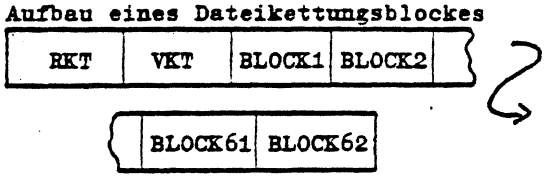
Anmerkung:
Der Dateikettungsblock enthält die Diskettenadressen aller Blöcke einer Datei. Aufbau siehe Seite .

```
Disk-Edit  Rev.:001  From:01/02 (HEX) To:01/05 (HEX) Page:0001

Drive: 1 Track: 01 (HEX) Sector: 02 (HEX) Code:HX/AS
      I S I S     D  I R                I S I S     M  A P
000:004953495300004 4 4952818019000101 004953495300004D 4150818002000102
      I S I S     L  A B                I S I S     E  R R
020:004953495300004C 4142818001001900 0049534953000045 5252818017001800
      S Y S T E M       . Ö <           I N I T
040:005359535445 4D00 0000065C3C000402 00494E4954000000 0000060A12000D04
      F O R M A T       X *             C O P Y                   p
060:00464F524D415400 0000065B2A000605 00434F5059000000 000006700E001706

Drive: 1 Track: 01 (HEX) Sector: 03 (HEX) Code:HX/AS
      D C O P Y               >         D G E N                   w
000:0044434F50590000 0000063E07000C07 0044474E45000000 0000067705001407
      D I R                             D I R P A C               u
020:0044495200000000 0000060F10001A07 0044495250414300 0000067503001108
      R E M A P               $         D E L E T E
040:0052454D41500000 0000062405001508 0044454C45544500 0000065F08000109
      R E S C U E             B         A L L O C
060:0052455343554500 0000067E06000A09 00414C4C4F430000 000006 1A0F001109

Drive: 1 Track: 01 (HEX) Sector: 04 (HEX) Code:HX/AS
      A L I A S               >         R E N A M E               6
000:00414C4941530000 0000063E0500070A 0052454E414D4500 0000063608000D0A
      A T T R I B             &         E X E C                   ?
020:0041545452494200 000007260600160A 0045584543000000 0000063F01000308
      E D I T                 c         T X T                     !
040:0045444954000000 000006638800050B 0054585400000000 0000072111000E10
      A S M                   ? .       R A S M                   X 4
060:0041534D00000000 0000063F2E000611 005241534D000000 0000065834000113

Drive: 1 Track: 01 (HEX) Sector: 05 (HEX) Code:HX/AS
      X R E F                           L I N K
000:0058524546000000 0000061011000215 004C494E4B000000 0000060A1C001415
      D E B U G                         M E M D M P               8
020:0044454255470000 0000060E01001716 004D454D444D5000 0000063805001916
      M A 2 2     1  3 2                D M 2 2     1  3 2  V
040:004D413232000031 3332071402000517 00444D3232000031 3332075602000817
      M A 2 5                           D M 2 5                   $
060:004D413235000000 0000071402000B17 00444D3235000000 0000072401000E17
```

```
DISK-EDIT  REV.:051  FROM:01/01 (HEX) TO:01/1A (HEX) PAGE:0002

DRIVE: 1 TRACK: 01 (HEX) SECTOR: 05 (HEX) CODE:HX/AS
      D E B U G                          M E M D M P           8
000:0044454255470000 0000060E01001115 004D454D444D5000 0000073805001315
      S T A C K                          A U F G O 5           &
020:00535441434B0000 0000001701001915 0041554647303500 0000002601000116
    % A S M                  ? .      % R A S M             X 4
040:7F41534D00000000 0000063F2E001A0F 7F5241534D000000 0000065834001511
    % L I N K                         % X R E F
060:7F4C494E4B000000 0000060A1C001613 7F58524546000000 0000061011001914

DRIVE: 1 TRACK: 01 (HEX) SECTOR: 06 (HEX) CODE:HX/AS
    % D E B U G                       % M E M D M P           8
000:7F44454255470000 0000060E01001115 7F4D454D444D5000 0000073805001315
    % S T A C K                       % E D I T               C
020:7F535441434B0000 0000001701001915 7F45444954000000 000006638800110A
    % A S M                  ? .      % R A S M             X 4
040:7F41534D00000000 0000063F2E001A0F 7F5241534D000000 0000065834001511
    % L I N K                         % X R E F
060:7F4C494E4B000000 0000060A1C001613 7F58524546000000 0000061011001914

DRIVE: 1 TRACK: 01 (HEX) SECTOR: 07 (HEX) CODE:HX/AS
    % D E B U G                       % M E M D M P           8
000:7F44454255470000 0000060E01001115 7F4D454D444D5000 0000073805001315
    % S T A C K                       % E D I T               C
020:7F535441434B0000 0000001701001915 7F45444954000000 000006638800110A
    % A S M                  ? .      % R A S M             X 4
040:7F41534D00000000 0000063F2E001A0F 7F5241534D000000 0000065834001511
    % L I N K                         % X R E F
060:7F4C494E4B000000 0000060A1C001613 7F58524546000000 0000061011001914

DRIVE: 1 TRACK: 01 (HEX) SECTOR: 08 (HEX) CODE:HX/AS
    % D E B U G                       % M E M D M P           8
000:7F44454255470000 0000060E01001115 7F4D454D444D5000 0000073805001315
    % S T A C K                       % E D I T               C
020:7F535441434B0000 0000001701001915 7F45444954000000 000006638800110A
    % A S M                  ? .      % R A S M             X 4
040:7F41534D00000000 0000063F2E001A0F 7F5241534D000000 0000065834001511
    % L I N K                         % X R E F
060:7F4C494E4B000000 0000060A1C001613 7F58524546000000 0000061011001914
```

Über die Datei ISIS.DIR kann für jede Datei die Diskettenadresse des ersten Dateikettungsblockes ermittelt werden. Dieser enthält für jeden Block einer Datei einen Eintrag über die Sektor- und Spurnummer. Werden mehr als ein Kettungsblock erforderlich, so werden die Dateikettungsblöcke sowohl vorwärts, als auch rückwärts gekettet.

Aufbau eines Dateikettungsblockes



RKT = Sektor- und Spurnummer des vorherigen Dateikettungsblockes (Rückwärts-Kettung)

VKT = Sektor- und Spurneummer des nächsten Dateikettungsblockes (Vorwärtskettung)

BLOCK1 = Sektor- und Spurnummer des 1. bis
⋮
BLOCK62 = 62. Datenblockes

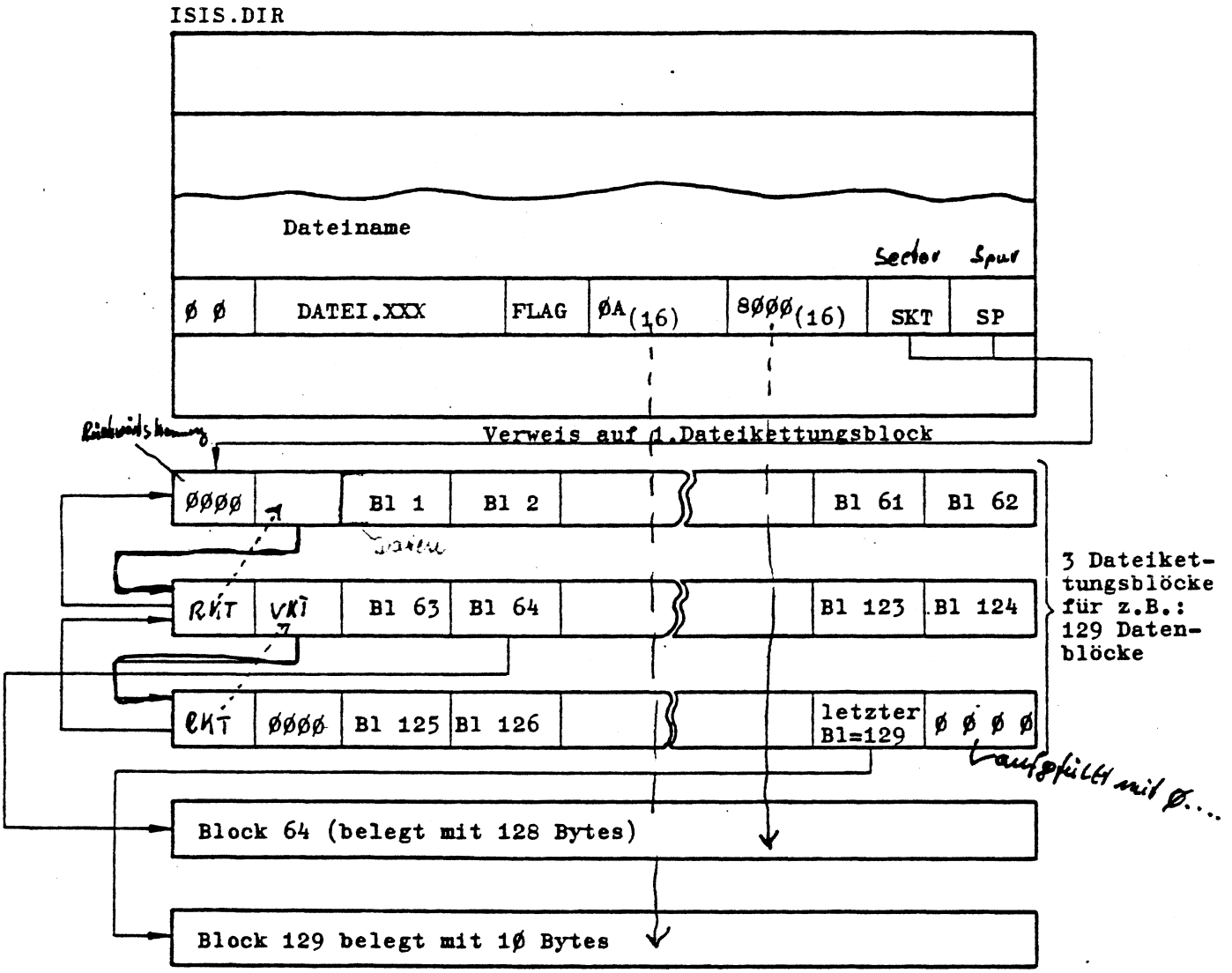
Anmerkung:
Sind weniger als 62 Blöcke vorhanden, sind die restlichen Diskettenadressen ØØØØ

```
 Disk-Edit  Rev.:001  From:18/02 (HEX) To:18/02 (HEX) Page:0001
 Drive: 1 Track: 18 (HEX) Sector: 02 (HEX) Code:HX/AS
000:00000D1A03180418 0518061807180818 09180A180B180C18 0D180E180F181018
020:1118121813181418 1518161817181818 19181A1801190219 0319041905190619
040:0719081909190A19 0B190C190D190E19 0F19101911191219 1319141915191619
060:1719181919191A19 011A021A031A041A 051A061A071A081A 091A0A1A0B1A0C1A
```

```
 Disk-Edit  Rev.:001  From:1A/0D (HEX) To:1A/0D (HEX) Page:0001
 Drive: 1 Track: 1A (HEX) Sector: 0D (HEX) Code:HX/AS
000:0218181C0E1A0F1A 101A111A121A131A 141A151A161A171A 181A191A1A1A011B
020:021B031B041B051B 061B071B081B091B 0A1B0B1B0C1B0D1B 0E1B0F1B101B111B
040:121B131B141B151B 161B171B181B191B 1A1B011C021C031C 041C051C061C071C
060:081C091C0A1C0B1C 0C1C0D1C0E1C0F1C 101C111C121C131C 141C151C161C171C
```

```
 Disk-Edit  Rev.:001  From:1C/18 (HEX) To:1C/18 (HEX) Page:0001
 Drive: 1 Track: 1C (HEX) Sector: 18 (HEX) Code:HX/AS
000:0D1A0000191C1A1C 011D021D031D041D 051D061D071D081D 091D0A1D0B1D0C1D
020:0D1D0E1D0F1D101D 111D121D131D141D 151D161D171D181D 191D1A1D011E021E
040:031E041E051E061E 071E081E091E0A1E 0B1E0C1E0D1E0E1E 0F1E101E111E121E
060:131E141E151E161E 171E181E191E1A1E 0000000000000000 0000000000000000
```

Dateiverwaltung: Übersicht



Dateiverwaltung: Beispiel

```
Disk-Edit  Rev.:001  From:01/07 (HEX) To:01/07 (HEX) Page:0001

Drive: 1 Track: 01 (HEX) Sector: 07 (HEX) Code:HX/AS
      B I N A R 3          +          B C D A R 1          -
000:0042494E41523300 0000002B0100051F 0042434441523100 000000400100071F
      B C D A R 2          A          B C D A R 2 S  R C
020:0042434441523200 000000410100091F 0042434441523253 524300130F000B1F
      S T A C K                       A U F G 0 7          &
040:00535441434B0000 0000001701000120 0041554647303700 0000002601000320
      S S T B A S A  S S                S S T B A S B  A S  d
060:0053535442415341 5353001002000520 0053535442415342 4153006406000320
                                                        Sec  Sp.
```

Auszug aus ISIS.DIR

```
Disk-Edit  Rev.:001  From:20/08 (HEX) To:20/08 (HEX) Page:0001

Drive: 1 Track: 20 (HEX) Sector: 08 (HEX) Code:HX/AS
                  ← Datenblock
000:0000000009200A20 0B200C200D200E20 0000000000000000 0000000000000000
020:0000000000000000 0000000000000000 0000000000000000 0000000000000000
040:0000000000000000 0000000000000000 0000000000000000 0000000000000000
060:0000000000000000 0000000000000000 0000000000000000 0000000000000000
```

Dateikettungs-block

```
Disk-Edit  Rev.:001  From:20/09 (HEX) To:20/09 (HEX) Page:0001

Drive: 1 Track: 20 (HEX) Sector: 09 (HEX) Code:HX/AS
    0 0 1 0  D I S  P L A Y  a $ , 1 , 2 7 , 6 , 1      0 0 2 0  D
000:3030313020444953 504C415920612420 312C32372C362C31 0D0A303032302044
    I S P L A Y  i $ , 2 5 , 2 4 , 1 , 4    0 0 3 0  L E T  b $
020:4953504C41592069 242C32352C32342C 312C340D0A303033 30204C4554206224
    = "       "    0 0 4 0  L E T  c $ = "
040:3D2220202020220D 0A30303430204C45 542063243D222020 2020202020202020
                "    0 0 5 0  P R I N T  , C L R    0 0 6 0
060:2020202020202022 0D0A303035302050 52494E54202C434C 520D0A3030363020
```

1. Datenblock

DISKETTE NACH "FORMAT"

```
Disk-Edit  Rev.:001  From:00/01 (DEC) To:76/26 (DEC) Page:0001

Drive: 1 Track: 00 (DEC) Sector: 01 (DEC) Code:HX/AS
     E N D   O F   F I L E                                    S E C T O R   M I S S I N G
000:854554E4204F462046494C4520202020202020202020202020202020202020202020202020202020202020202020202020
              C R C   E R R O R                                          T I M I N G   E R R O R   R E A D / W R I T E
060:202020208543524320455252414F52202020202020202020202020202020202020202020202020202020202020202020202020
     X F E R
120:2058464552202020

Drive: 1 Track: 00 (DEC) Sector: 02 (DEC) Code:HX/AS
     N O   A D D R E S S   M A R K                            D I S K E T T E   O P E R A T I O N   T I M E O U T
000:854E4F204144444524553532044152420202020202020202020202020202020202020202020202020202020202020202020
              T I M I N G   E R R O R   I N   R E A D   O P E R A T I O N   B U S Y - B I T   T I M E   O U T
060:202020208554494D494E4720455252414F52204948452020202020202020202020202020202020202020202020202020

120:2020202020202020

Drive: 1 Track: 00 (DEC) Sector: 03 (DEC) Code:HX/AS
     D R I V E   N O T   R E A D Y                            N A M E   N O T   I N   D I R   A N D   D I R   F U L
000:85445249564520454E4F5420524541445920202020202020202020202020202020202020202020202020202020202020202020
     L        N A M E   N O T   I N   D I R E C T O R Y                  P R O G R A M   N A M E   N O T   F O U N D
060:4C20202085414144520202020202020202020202020202020202020202020202020202020202020202020202020202020202020

120:2020202020202020

Drive: 1 Track: 00 (DEC) Sector: 04 (DEC) Code:HX/AS
     I L L E G A L   D I R E C T O R Y   E N T R Y            D I R E C T O R Y   B L O C K   E R R O R   -   L O A
000:8549434C4547414C2044495245435449524F52592045204552592020202020202020202020202020202020202020202020
     D        D A T A   L E N G T H   E R R O R   D U R I N G   L O A D      D I S K E T T E   M A P   M I S S I N G
060:4420202085444154420202020202020202020202020202020202020202020202020202020202020202020202020202020202

120:2020202020202020

Drive: 1 Track: 00 (DEC) Sector: 05 (DEC) Code:HX/AS
     N O N - V A L I D   D I S K E T T E   C O M M A N D      I M P R O P E R   U N I T   S P E C I F I C A T I O N
000:854E4F4E2D56414349442044495354455220434F4D4D414E4420202020202020202020202020202020202020202020202020
              C O N V E R S I O N   T A B L E   M I S S I N G                D R I V E   N U M B E R   N O T   0 - 3
060:202020208543494E564552534F4E20544124C4520414D495353494E4720202020202020854454520202020202020202020202

120:2020202020202020

Drive: 1 Track: 00 (DEC) Sector: 06 (DEC) Code:HX/AS
     - - - - -U N U S E D - - - - - - - - - - - - - - - - - -  D E - E D I T   N O N H E X   D I G I T   ( 0 - 9 , A
000:852D2D2D2D2D554E5553454432D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D
     - F )    I N P U T   F R O M   N O N - I N P U T   F I L E                R E A D   A F T E R   E N D - O F - F I L E
060:2D46292085494E505554204652414D204E4F4E2D49414550555420464941544552204E4F4E2D464F4C4520202020202020

120:2020202020202020

Drive: 1 Track: 00 (DEC) Sector: 07 (DEC) Code:HX/AS
     O U T P U T   T O   N O N - O U T P U T   F I L E        U N A S S I G N E D   D E V I C E
000:854F5554505554205446F204E4F4E2D4F555450554E415353494744454420444556494345202020202020202020202020
              U N K N O W N   D E V I C E   T Y P E                                C O M / F N A M   N O T   L E T T E R / D I G
060:202020208555345B4E4F574E204441556949345205950452020202020202020202020202020854F4D2F464E414D204E4F5420
     I T   1 S T
120:4954203153542020

Drive: 1 Track: 00 (DEC) Sector: 08 (DEC) Code:HX/AS
     C O M / F N A M   >  6   O R   E X T   >  3   C H A R S  C O M M A N D   U N I T   N O T   :F N :   O R   :H
000:8543414D2F4E414E414D203E203620464F522045585420333E203320434841525320204854414E444155454E20554E495420
     N :      T O S   N O T   L O A D E D                                  C O M M A N D   N O T   T E R M I N A T E D
060:4E3A2020855444F5320454F5420434F4144454420202020202020202020202020202020202020854144454420544552
     P R O P E R L Y
120:5052414F504552494359
```

```
Disk-Edit  Rev.:001  From:00/01 (DEC) To:76/26 (DEC) Page:0002

Drive: 1 Track: 00 (DEC) Sector: 09 (DEC) Code:HX/AS
     D E L E T E D   D A T A   R E C O R D                    L O G I C A L   U N I T   N A M E   U N K N O W N
000:854444544C4554454442044 4154411205245434F5244 2020202020202020 2020854CAF474943414C 2055AE495A204E414D45 2055AE4BAE4F57AE2020
          W R I T E   O N   N O N - B L  O C K  F I L E                    W R I  T E / R E A D  M I  S S I N G  I N  C
060:20202020855752495445 204F4E204EAF4E2D424C 4F434B2046494C452020 2020202020208557524 9 5445 2F5245414420AD49 5353494EA7204 9AE20A3
    T L - B L K
120:544C2D424C4B2020

Drive: 1 Track: 00 (DEC) Sector: 10 (DEC) Code:HX/AS
     R E A D   O N   F  I L E T Y P E   O T H E R   T H A N   4  , 5   L I N E   O V  E R F L O W
000:8552454144204F4E2046 494C4554595045204F54 4845522054484A4E2034 2C3585AC494E4520AF56 4552464C4F5720202020 2020202020202020
          - - - - -  U N U S E D - - - -  - - - - - - - - - -  - - - - - -  F I L  E  A L R E A D Y   O P E N
060:20202020852D2D2D2D2D 554E5553454A2D2D2D2D 2D2D2D2D2D2D2D2D 2D2D2D2D2D2D2D2D 2D2D2D2D2D2D854649AC 4520414C52454A4A5920 AF50454E202020202020

120:2020202020202020

Drive: 1 Track: 00 (DEC) Sector: 11 (DEC) Code:HX/AS
     N O   O P E N - R  O U T I N E   I N    C T L - B L K          C L O S E   O N  C L O S E D  F  I L E
000:85AE4F204F50A5AE2D52 4F5554494E4520494E20 4354AC2D424CAB202020 2020854341C4F53A520F AE2043AC4F53A54A2046 49AC4520202020202020
          N O   C L  O S E - R O U T I N E   I N   C T L - B  L K          B U F  F E R   C H A I N    D E S T R O Y E D
060:2020202085AE4F2043AC 4F534252D52AF5554494E 4520494E20435AAC2DA2 ACAB20202020854255A6 46A55220434841494E20 4A4A5535452AF594544120

120:2020202020202020

Drive: 1 Track: 00 (DEC) Sector: 12 (DEC) Code:HX/AS
     B U F F E R   H E  A D E R   I M P R O  P E R                    I M P R O P E  R  F I L E  D E S  I G N A T O R
000:854255464655220A845 4144455220494D5052AF 50455220202020202020 2020854941D5052AF5045 5220464941C4520444553 4947AE4154AF52202020
          I M P R O  P E R  A S S I G N  M E N T  F O R M A  T                    N O   R O O M   O N   D I  S K E T T E
060:20202020854941D5052AF 5045522041535349474E 4D454E5420464F524D41 5420202020208 5AE4F20 524F4F4D204F4E2044A9 534B4554544520202020

120:2020202020202020

Drive: 1 Track: 00 (DEC) Sector: 13 (DEC) Code:HX/AS
     - - - - -U N U S  E D - - - - - - - - -  - - - 3 0 H - - - -  - -  N O N - E X I  S T E N T  F I L E   N A M E  -  I N
000:852D2D2D2D2D554E5553 4544A2D2D2D2D2D2D2D2D 2D2D2D3330482D2D2D2D 2D2D854E4F4E2D558449 535445452046494C45 204E4144D5202D2049AE
    P U T   O U T P U T  F I L E  A L R E A D Y  O N  D I S K E T T E  N O   R O O M  O N  D I  S K E T T E
060:505554208854F55545055 5420464941C4520414C52 4541445920F4E20A449 534B4554544585AE4F20 524F4F4D204F4E2044A9 534B4554544520202020

120:2020202020202020

Drive: 1 Track: 00 (DEC) Sector: 14 (DEC) Code:HX/AS
     F I L E N A M E   >  8  C H A R A C  T E R S                    T R A C K  N  U M B E R  >  7 6
000:8546494C454E414D4520 3E203820434841524143 5445525320202020202020 2020855452414348204E 554D42552203E203736 2020202020202020202020
          S E C T O R  N U M B E R  =   0  O R  >  2 6                    B U F  F E R   A D D R E S S   Z E R O
060:2020202085534543544F 5220AE554D424552203D 203020AF52203E203236 2020202020208545544 6A55220A1444A4524553 53205A4552AF20202020

120:2020202020202020

Drive: 1 Track: 00 (DEC) Sector: 15 (DEC) Code:HX/AS
     R O U T I N E   N O T  A V A I L A B  L E                    : M N :   D I  R  B L O C K  N O  T  3 2  B Y T E S
000:8552AF555449AE4520AE 4F5420415641494C4142 4C4520202020202020 2020853A4D4E3A204A49 5220424C4F434B20AE4F 5420333220425954A553
          V O L U M  E  H E A D E R  M I S S I N G                    D I R  E C T O R Y  B L O C K  E R R O R
060:2020202085564F4C554D 4520484541444A55220AD 495353494E4720202020 2020202020208544A952 45435447F525920424C4F 434A20455252AF522020

120:2020202020202020

Drive: 1 Track: 00 (DEC) Sector: 16 (DEC) Code:HX/AS
     O U T P U T   F I  L E   A L R E A D Y   O N   C A R T R .          N O   M O R E   S P A C E  O N   C A R T R I D G E
000:854F5554505554204649 4C4520414C5245414459 204F4E2043415254522E 2020854EAF204D4F5245 2053504143A520AF4E20 4341525A5249444A4520
          I N P U T   F I L E  O V E R  R U N                    E X T  R A  O R  M I S P  L A C E D  T A P E
060:2020202085494E505554 204649AC4520AF564552 5255AE2020202020202020 2020202020208545054 524120AF5220AD495350 4C4143A54A2054415045
    M A R K
120:4D4152AB20202020
```

```
Disk-Edit  Rev.:001  From:00/01 (DEC) To:76/26 (DEC) Page:0003

 Drive: 1 Track: 00 (DEC) Sector: 17 (DEC) Code:HX/AS
      E N D   O F   T A P E   E R R O R                              C A R T R I D G E   C R C   E R R O R
000:8545H4H4204F4620544 1 504520455252 4F522020 2020202020202020202020 2020854341452545249444 4745204352432045525 2 4F52202020202020202020
            F A T A L     C A R T R I D G E     E R R O R                              C A R   T R I D G E   C R C     +   F A T A L   E
060:2020202085464154414C 2043415254529444474 5 20455252 4F5220202020 2020202020208543 4152 5452494447452043523 202B204641544 14C2045
     R R O R
120:52524F5220202020

 Drive: 1 Track: 00 (DEC) Sector: 18 (DEC) Code:HX/AS
      C A R T R .   W R I T E   P R O T E C T / I N V A L I D     C M   C R C   +   W R I T . P R O T / I   N V . C O M
000:8543415254522E205752 4954452050524F5445 43 5 42F494E5641 4C494420 434D8543524320B2057 52495 42E50524F542F49 4E562E434F4D20202020
            R E A D I N G     E R A S E D     T A P E                              C R C     +   F A T A L   +     W R . P R T / I N
060:2020202085524541 4449 4E4720455241534544 20 5441504520202020202020 2020202020208543523 202B204641544 14C202B 2057522E50525 42F49 4E
     V . C M
120:562E434D20202020

 Drive: 1 Track: 00 (DEC) Sector: 19 (DEC) Code:HX/AS
      T A P E   M A R K     D E T E C T E D                                C R C   +   T A P E   M A R K
000:8554415041504D41524B 2044455445354544420 2020202020202020202020 2020854352432022B2054 4150452041D41524B2020 2020202020202020202020
            F A T A L     E R R O R   +   T A P E   M A R K                              C R C     +   F A T A L   +     T A P E   M A R K
060:2020202085464154414C 2055252 4F52202B2054 4150452041D41524B2020 2020202020208543523 202B204641544 14C202B 2054415041504D41524B

120:2020202020202020

 Drive: 1 Track: 00 (DEC) Sector: 20 (DEC) Code:HX/AS
      W R T . P T / I N   V . C M   +   T A P E   M A R K                  C R C   +   W R T . P T / I N V . C M   +   T A P E
000:8557525 42E505 42F494E 562E434D202B205441150 4520 4D41524B20202020 2020854352432022B2057 52542E505 42F494E562E 434D202B205441504520
      M A R K   F A T A L   + W R T . P T / I N   V . C M + T A P E     M A R K       C R C   + F A T A L + W P T   / I V C + T A P E
060:4D41524B8546154414C 2B57525 42E505 42F49 4E 562E434D2B5441504520 4D41524B2020854352432 2B464154414C2B57505 4 2F4956432B5441504520
      M A R K
120:4D41524B20202020

 Drive: 1 Track: 00 (DEC) Sector: 21 (DEC) Code:HX/AS
      C A R T R   I L L E G A L   U N I T   N U M B E R                  C A R T R I D G E   I L L E G A L     T R A C K   N U M
000:8543415254522049434C 4547414C20554E495420 4E554D42455220202020 2020854341452545249444 4745204949434C4547414C 2054524143 4B204E554D
      B E R     C A R T R I D G E   B L O C K     S I Z E   I L L E G A L           C A R   T R I D G E   T R A   C K   C H A N G E
060:42455220854341525452 4944474520424C4F434B 2053495A4520494C4C45 4741 4C20202085434152 5452494447452054541 434B204348414E474520
      E R R O R
120:455252 4F52202020

 Drive: 1 Track: 00 (DEC) Sector: 22 (DEC) Code:HX/AS
      C A R T R I D G E     U N I T   N O T   R E A D Y                    C A R T R I D G E   I L L E G A L     B U F F E R   A D
000:8543415254529444474 5 2055 4E495420 4E4F5420 52454144592020202020 2020854341452545249444 4745204949434C4547414C 2041255464655220414 4
      D R .     C A R T R   I D G E   N O T   S U P P O R T E D                              C A R   T R I D G E   E X T   R A   B L O C K S
060:4452 2E2085434152525452 4944474520 4E4F542053 555050 4F525445 4420 2020232020208543 4152 5452494447452055854 524120424C4F434B5320
      I N   F I L E
120:494E204649 4C4520

 Drive: 1 Track: 00 (DEC) Sector: 23 (DEC) Code:HX/AS
      C A R T R I D G E     C O N T .   H D R     M I S S I N G                I N C O R R E C T   B L O C K   T Y P E   I N   H E A
000:8543415254529444474 5 2043 4F4E542E204844452 2041D4953539 4E472020 2020854 94E434F5252 45 435420424C4F434B2054 5950452049 4E204844541
      D E R     I N C O R   R E C T   N A M E     I N   H E A D E R     B L O C K     I N A   P P R O P R I A T E     B L K   S I Z E
060:4445522085494E434F52 5245435420 4E414D4520 494E204845414444 5220 424C4F434B2085494E41 505052 4F5052494154455 2042 4C4B2053495A4520
      L O A D
120:4C4F414420202020

 Drive: 1 Track: 00 (DEC) Sector: 24 (DEC) Code:HX/AS

000:0000000001000200030 0 040005000600070008 00 09000A000B000C000D00 0E000F0010001100 1200 1300140015001600 1700 00000000000000000000

060:0000000000000000000 0 000000000000000000 00 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000

120:00000000000000
```

← Fabrikkennsatzblock bis ISIS. ERR

```
Disk-Edit  Rev.:001  From:00/01 (DEC) To:76/26 (DEC) Page:0004

Drive: 1 Track: 00 (DEC) Sector: 25 (DEC) Code:HX/AS

000:000000001A0000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000
060:00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000
120:0000000000000000

Drive: 1 Track: 00 (DEC) Sector: 26 (DEC) Code:HX/AS
    S B B 6     6 1 0
000:53424236000036313000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000
060:00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000
120:0000000000000000

Drive: 1 Track: 01 (DEC) Sector: 01 (DEC) Code:HX/AS

000:00000000020103010401 05010601070108010901 0A010B010C010D010E01 0F011001110112011301 14011501160117011801 19011A010000000000000
060:00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000
120:0000000000000000

Drive: 1 Track: 01 (DEC) Sector: 02 (DEC) Code:HX/AS
      I S I S     D I R              I S I S     M A P              I S I S     L A B              I S I S     E R R
000:004953495300004449521818019000101004953491 53000040415081800200 0102004953495300004C 4142818001001900004949 534953000045525281800
060:17001800 7F0000000000 00000000000000000000 7F0000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000000
120:0000000000000000

Drive: 1 Track: 01 (DEC) Sector: 03 (DEC) Code:HX/AS

000:7F0000000000000000000 00000000000007F000000 00000000000000000000 00007F0000000000000 00000000000000007F00 00000000000000000000
060:000000007F0000000000 00000000000000000000 7F0000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000000
120:0000000000000000

Drive: 1 Track: 01 (DEC) Sector: 04 (DEC) Code:HX/AS

000:7F0000000000000000000 00000000000007F000000 00000000000000000000 00007F0000000000000 00000000000000007F00 00000000000000000000
060:000000007F0000000000 00000000000000000000 7F0000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000000
120:0000000000000000

Drive: 1 Track: 01 (DEC) Sector: 05 (DEC) Code:HX/AS

000:7F0000000000000000000 00000000000007F000000 00000000000000000000 00007F0000000000000 00000000000000007F00 00000000000000000000
060:000000007F0000000000 00000000000000000000 7F0000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000000
120:0000000000000000

Drive: 1 Track: 01 (DEC) Sector: 06 (DEC) Code:HX/AS

000:7F0000000000000000000 00000000000007F000000 00000000000000000000 00007F0000000000000 00000000000000007F00 00000000000000000000
060:000000007F0000000000 00000000000000000000 7F0000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000000
120:0000000000000000
```

```
Disk-Edit  Rev.:001  From:00/01 (DEC) To:76/26 (DEC) Page:0005

 Drive: 1 Track: 01 (DEC) Sector: 07 (DEC) Code:HX/AS

000:7F00000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000 00000000000000007F00 00000000000000000000

060:000000007F0000000000 00000000000000000000 7F00000000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000

120:00000000000000000

 Drive: 1 Track: 01 (DEC) Sector: 08 (DEC) Code:HX/AS

000:7F00000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000 00000000000000007F00 00000000000000000000

060:000000007F0000000000 00000000000000000000 7F00000000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000

120:00000000000000000

 Drive: 1 Track: 01 (DEC) Sector: 09 (DEC) Code:HX/AS

000:7F00000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000 00000000000000007F00 00000000000000000000

060:000000007F0000000000 00000000000000000000 7F00000000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000

120:00000000000000000

 Drive: 1 Track: 01 (DEC) Sector: 10 (DEC) Code:HX/AS

000:7F00000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000 00000000000000007F00 00000000000000000000

060:000000007F0000000000 00000000000000000000 7F00000000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000

120:00000000000000000

 Drive: 1 Track: 01 (DEC) Sector: 11 (DEC) Code:HX/AS

000:7F00000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000 00000000000000007F00 00000000000000000000

060:000000007F0000000000 00000000000000000000 7F00000000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000

120:00000000000000000

 Drive: 1 Track: 01 (DEC) Sector: 12 (DEC) Code:HX/AS

000:7F00000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000 00000000000000007F00 00000000000000000000

060:000000007F0000000000 00000000000000000000 7F00000000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000

120:00000000000000000

 Drive: 1 Track: 01 (DEC) Sector: 13 (DEC) Code:HX/AS

000:7F00000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000 00000000000000007F00 00000000000000000000

060:000000007F0000000000 00000000000000000000 7F00000000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000

120:00000000000000000

 Drive: 1 Track: 01 (DEC) Sector: 14 (DEC) Code:HX/AS

000:7F00000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000 00000000000000007F00 00000000000000000000

060:000000007F0000000000 00000000000000000000 7F00000000000000000000 00000000000007F000000 00000000000000000000 00007F00000000000000

120:00000000000000000
```

```
Disk-Edit  Rev.:001  From:00/01 (DEC) To:76/26 (DEC) Page:0006
 Drive: 1 Track: 01 (DEC) Sector: 15 (DEC) Code:HX/AS
000:7F00000000000000000 00000000007F000000 00000000000000000000 00007F00000000000000 0000000000000007F00 00000000000000000000
060:000000007F0000000000 00000000000000000000 7F00000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000
120:00000000000000000
 Drive: 1 Track: 01 (DEC) Sector: 16 (DEC) Code:HX/AS
000:7F00000000000000000 00000000007F000000 00000000000000000000 00007F00000000000000 0000000000000007F00 00000000000000000000
060:000000007F0000000000 00000000000000000000 7F00000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000
120:00000000000000000
 Drive: 1 Track: 01 (DEC) Sector: 17 (DEC) Code:HX/AS
000:7F00000000000000000 00000000007F000000 00000000000000000000 00007F00000000000000 0000000000000007F00 00000000000000000000
060:000000007F0000000000 00000000000000000000 7F00000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000
120:00000000000000000
 Drive: 1 Track: 01 (DEC) Sector: 18 (DEC) Code:HX/AS
000:7F00000000000000000 00000000007F000000 00000000000000000000 00007F00000000000000 0000000000000007F00 00000000000000000000
060:000000007F0000000000 00000000000000000000 7F00000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000
120:00000000000000000
 Drive: 1 Track: 01 (DEC) Sector: 19 (DEC) Code:HX/AS
000:7F00000000000000000 00000000007F000000 00000000000000000000 00007F00000000000000 0000000000000007F00 00000000000000000000
060:000000007F0000000000 00000000000000000000 7F00000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000
120:00000000000000000
 Drive: 1 Track: 01 (DEC) Sector: 20 (DEC) Code:HX/AS
000:7F00000000000000000 00000000007F000000 00000000000000000000 00007F00000000000000 0000000000000007F00 00000000000000000000
060:000000007F0000000000 00000000000000000000 7F00000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000
120:00000000000000000
 Drive: 1 Track: 01 (DEC) Sector: 21 (DEC) Code:HX/AS
000:7F00000000000000000 00000000007F000000 00000000000000000000 00007F00000000000000 0000000000000007F00 00000000000000000000
060:000000007F0000000000 00000000000000000000 7F00000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000
120:00000000000000000
 Drive: 1 Track: 01 (DEC) Sector: 22 (DEC) Code:HX/AS
000:7F00000000000000000 00000000007F000000 00000000000000000000 00007F00000000000000 0000000000000007F00 00000000000000000000
060:000000007F0000000000 00000000000000000000 7F00000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000
120:00000000000000000
```

Disk-Edit Rev.:001 From:00/01 (DEC) To:76/26 (DEC) Page:0007

Drive: 1 Track: 01 (DEC) Sector: 23 (DEC) Code:HX/AS

000:7F0000000000000000000 0000000000007F000000 0000000000000000000 0007F00000000000000 0000000000000007F00 0000000000000000000

060:000000007F0000000000 0000000000000000000 7F0000000000000000000 0000000000007F000000 0000000000000000000 00007F000000000000

120:00000000000000000

Drive: 1 Track: 01 (DEC) Sector: 24 (DEC) Code:HX/AS

000:7F0000000000000000000 0000000000007F000000 0000000000000000000 00007F00000000000000 0000000000000007F00 0000000000000000000

060:000000007F0000000000 0000000000000000000 7F0000000000000000000 0000000000007F000000 0000000000000000000 00007F000000000000

120:00000000000000000

Drive: 1 Track: 01 (DEC) Sector: 25 (DEC) Code:HX/AS

000:7F0000000000000000000 0000000000007F000000 0000000000000000000 00007F00000000000000 0000000000000007F00 0000000000000000000

060:000000007F0000000000 0000000000000000000 7F0000000000000000000 0000000000007F000000 0000000000000000000 00007F000000000000

120:00000000000000000

Drive: 1 Track: 01 (DEC) Sector: 26 (DEC) Code:HX/AS

000:7F0000000000000000000 0000000000007F000000 0000000000000000000 00007F00000000000000 0000000000000007F00 0000000000000000000

060:000000007F0000000000 0000000000000000000 7F0000000000000000000 0000000000007F000000 0000000000000000000 00007F000000000000

120:00000000000000000

Drive: 1 Track: 02 (DEC) Sector: 01 (DEC) Code:HX/AS

000:00000000020203020000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000

060:0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000

120:00000000000000000

Drive: 1 Track: 02 (DEC) Sector: 02 (DEC) Code:HX/AS

000:FFFFFFFFFFFFFE000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000

060:0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000

120:00000000000000000

Drive: 1 Track: 02 (DEC) Sector: 03 (DEC) Code:HX/AS

000:0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000

060:0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000

120:00000000000000000

Drive: 1 Track: 02 (DEC) Sector: 04 (DEC) Code:HX/AS

000:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5

060:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5

120:E5E5E5E5E5E5E5E5E5

bis Disketten-Ende identisch mit Spur 02 / Sektor 04

SIEMENS

Schule für
Daten- und Informationssysteme
Systemservice-Ausbildung

8-14

```
SIEMENS BS 610 VER 1.62
*copy si=SYSTEM,so=:f1:SYSTEM
SIEMENS BS 610 VER 1.62
*copy si=BS610,so=:f1:BS610
SIEMENS BS 610 VER 1.62
*copy si=DIRPAC,so=:f1:DIRPAC
SIEMENS BS 610 VER 1.62
*delete$:f1:BS610
:F1:BS610     DELETED
SIEMENS BS 610 VER 1.62
*copy si=DEBUG,so=:f1:DEBUG
SIEMENS BS 610 VER 1.62
*
```

```
Disk-Edit  Rev.:001  From:00/26 (DEC) To:01/04 (DEC) Page:0001

Drive: 1 Track: 00 (DEC) Sector: 26 (DEC) Code:HX/AS
    S B B 6     6 1 0
000:53424236000036313000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000

060:00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000

120:0000000000000000

Drive: 1 Track: 01 (DEC) Sector: 01 (DEC) Code:HX/AS

000:00000000020103010401 05010601070108010901 0A010B010C010D010E01 0F011001110112011301 14011501160117011801 19011A01000000000000

060:00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000

120:0000000000000000

Drive: 1 Track: 01 (DEC) Sector: 02 (DEC) Code:HX/AS
      I S I S     D I R               I S I  S     M A P              I S I S     L  A B              I  S I S     E R R
000:004953495300004449 52818019000101004953 49530000 4D41508180020 0 010200495349530000 4C414281800100190000 49534953000045525281 80
            S Y S T E  M                  B S 6 1 0          V ?        D I R  P A C                    D E B U G
060:1700180000535953544 45 4D0000000020020004 02FF4253363130000000 00 00563F0007020049 44952504143000000000 30400140400444542554 70000

120:0000001E01000702

Drive: 1 Track: 01 (DEC) Sector: 03 (DEC) Code:HX/AS

000:7F000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000 0000000000000007F00 00000000000000000000

060:000000007F0000000000 00000000000000000000 7F0000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000

120:0000000000000000

Drive: 1 Track: 01 (DEC) Sector: 04 (DEC) Code:HX/AS

000:7F000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000 0000000000000007F00 00000000000000000000

060:000000007F0000000000 00000000000000000000 7F0000000000000000000 0000000000007F000000 00000000000000000000 00007F000000000000

120:0000000000000000
```

SIEMENS

Schule für
Daten- und Informationssysteme
Systemservice-Ausbildung

S-15

IBM Diskettenformat

ECMA DC 19 FORMAT

Genauere Informationen können z. B. aus der Beschreibung des SIEMENS-Datenerfassungssystems TRANSDATA 920 entnommen werden.

Aufbau der Diskette

Spur	Sektor	
0	1)	Reserviert
0	4)	
0	5	ERMAP
0	6	Reserviert
0	7	Datenträger-Kennsatz
0	8)	Dateikennsätze
0	26)	
1	1)	Datenbereich
74	26)	
75	1)	Ersatzspur 1
75	26)	
76	1)	Ersatzspur 2
76	26)	

Im folgenden sind nur Felder beschrieben, welche bei der Benutzung IBM-formatierter Disketten im BS 610 relevant sind. Nicht beschriebene Felder sind als reserviert zu betrachten.

ERMAP

Spur 0 Sektor 5

Stelle	
1-5	ERMAP
7-8	Leer oder Nummer der ersten defekten Spur. Dezimal!
11-12	Leer oder Nummer der zweiten defekten Spur.

Datenträger-Kennsatz

Spur 0 Sektor 7

Stelle	
1 - 4	VOL 1
5 - 10	Datenträger-Kennsatzinformation
11	Zugriffsschutzfeld)*
38 - 51	Eigentümerinformation)*
76	Phy. Sektorlänge der Spuren 1 - 7)* leer - 128 Bytes 1 - 256 Bytes 2 - 512 Bytes
77 - 78	Phys. Sektorfolge

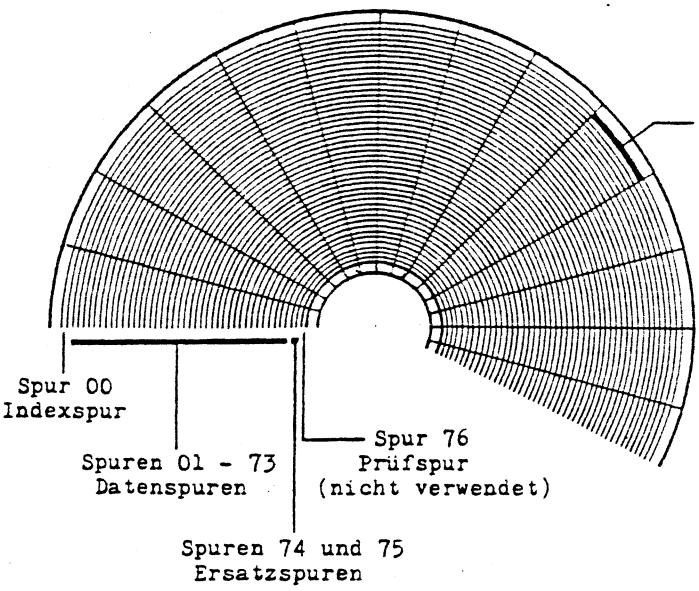
Dateikennsatz

Spur 0 Sektoren 8 - 26

Stelle	
1 - 4	HDR1 (Benutzer-Anfangsetikett)
6 - 13	Dateiname
29 - 33	Bereichsbeginn (BOE), erster zu einer Datei gehörender Sektor 29 - 30 Spurnummer, 32 - 33 Sektornummer
34	Phys. Satzlänge)* Vergl. Stelle 76 des Datenträger-Kennsatzes
35 - 39	Bereichsende (EOE), letzter für diese Datei reservierte Sektor Format wie BOE
42	Zugriffsschutzfeld)*
43	Schreibschutz Leerfeld - ungeschützt P - geschützt
48 - 53	Schreibdatum JJMMDD
54 - 57	Satzlänge)*
67 - 72	Freigabedatum JJMMDD
75 - 79	Datenende (EOD), Adresse des nächsten unbelegten Sektors Format wie BOE

)* vorläufig nicht benutzt

AUFBAU EINES SEKTORS



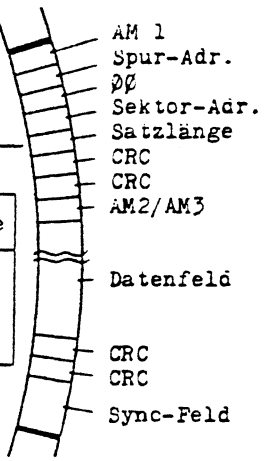
1 Sektor

Die Indexspur hat immer 26 Sektoren.

Die Daten- und Ersatzspuren können haben:

Zahl der Sektoren	Bytes im Datenfeld	Code
26	128	ØØ
15	256	Ø1
8	512	Ø2
4	1o24	Ø3

Keine Verwend. bei 6.610 !



AUFBAU DER INDEXSPUR

nur 655,6 erforderlich

Sektor 1
Sektor 2
Sektor 3
Sektor 4 — Reserviert für Urladeprogramm (siehe Befehl READ IPL)

Sektor 5 — Hex

1	2	3	4	5		7	8		11	12
E	R	M	A	P		X	X		Y	Y

1. ------ 2. Fehlerspur 74 – 75

Sektor 6 — frei

Sektor 7

1	2	3	4	5	6	7	8	9	1o		77	78		80
V	O	L	1								4	Ø		W

Eigentümer — Sektorfolge-Code

INTF: Ja: - sequentiell. Seqeg
INTF: Fa: - 1-14-2-15 ... -13

Sektor 8 bis 26 (Dateikennsätze)

1	2	3	4		6	7	8	9	1o	11	12	13		23	24	25	26	27
H	D	R	1															

Datei-Name — Satzlänge

29	3o	31	32	33		35	36	37	38	39		41	42	43		45	46	47	48	49	5o	51	52	53
T	T	O	S	S		T	T	O	S	S									Y	Y	M	M	D	D

BOE (Datei-Anfang) — EOE (Datei-Ende) — Erstellungsdatum

Fortl.Nummer (o1 - 99)

B = Datei wird nicht zur ZE übertragen

≠ Blank = Datei lesegeschützt

P = Datei schreibgeschützt

Mehrplattendatei-Anzeige
C = Datei wird fortgesetzt
L = Datei wird auf dieser FD beendet
- = Datei beginnt und endet auf dieser FD

67	68	69	7o	71	72	73		75	76	77	78	79
Y	Y	M	M	D	D			T	T	O	S	S

Freigabedatum — EOD (Schreib-Ende)

V = Datei wurde geprüft

DISKETTE NACH 'INIT'

```
Disk-Edit  Rev.:001  From:00/01 (DEC) To:76/26 (DEC) Page:0001

 Drive: 1 Track: 00 (DEC) Sector: 01 (DEC) Code:HX/EB
000:4040404040404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040
060:4040404040404040 4040404040404040 0000000000000000 0000000000000000 0000000000000000 0000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 02 (DEC) Code:HX/EB
000:4040404040404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040
060:4040404040404040 4040404040404040 0000000000000000 0000000000000000 0000000000000000 0000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 03 (DEC) Code:HX/EB
000:4040404040404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040
060:4040404040404040 4040404040404040 0000000000000000 0000000000000000 0000000000000000 0000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 04 (DEC) Code:HX/EB
000:4040404040404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040
060:4040404040404040 4040404040404040 0000000000000000 0000000000000000 0000000000000000 0000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 05 (DEC) Code:HX/EB
    E R M A P
000:C5D9D4C1D7404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040
060:4040404040404040 4040404040404040 0000000000000000 0000000000000000 0000000000000000 0000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 06 (DEC) Code:HX/EB
000:4040404040404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040
060:4040404040404040 4040404040404040 0000000000000000 0000000000000000 0000000000000000 0000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 07 (DEC) Code:HX/EB
    V O L 1
000:E5D6D3F140404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040 4040404040404040
                                    W
060:4040404040404040 4040404040404040 0000000000000000 0000000000000000 0000000000000000 0000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 08 (DEC) Code:HX/EB
    H D R 1   D A T A                                 0 8 0   0 1   0 0 1   7 3 0 2 6
000:C8C4D9F140C4C1E3C140 4040404040404040 4040404F0F0F0F0F1 F0F0F140F7F3F0F2F640 4040404040404040 4040404040404040
                         0 1 0 0 1
060:4040404040404040 4040404F0F1F0F0F140 0000000000000000 0000000000000000 0000000000000000 0000000000000000
120:0000000000000000
```

```
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 10 (DEC) Code:HX/EB

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 11 (DEC) Code:HX/EB

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 12 (DEC) Code:HX/EB

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 13 (DEC) Code:HX/EB

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 14 (DEC) Code:HX/EB

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 15 (DEC) Code:HX/EB

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 16 (DEC) Code:HX/EB

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 17 (DEC) Code:HX/EB

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 18 (DEC) Code:HX/EB

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 19 (DEC) Code:HX/EB

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 20 (DEC) Code:HX/EB

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 21 (DEC) Code:HX/EB

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 22 (DEC) Code:HX/EB
```

```
Disk-Edit  Rev.:001  From:00/01 (DEC) To:76/26 (DEC) Page:0003
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 23 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 24 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 25 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 26 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 01 (DEC) Sector: 01 (DEC) Code:HX/EB
    V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
000:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
    V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
060:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
    V V V V V V V V
120:E5E5E5E5E5E5E5E5

Drive: 1 Track: 01 (DEC) Sector: 02 (DEC) Code:HX/EB
    V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
000:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
    V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
060:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
    V V V V V V V V
120:E5E5E5E5E5E5E5E5

Drive: 1 Track: 01 (DEC) Sector: 03 (DEC) Code:HX/EB
    V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
000:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
    V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
060:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
    V V V V V V V V
120:E5E5E5E5E5E5E5E5

Drive: 1 Track: 01 (DEC) Sector: 04 (DEC) Code:HX/EB
    V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
000:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
    V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
060:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
    V V V V V V V V
120:E5E5E5E5E5E5E5E5

Drive: 1 Track: 01 (DEC) Sector: 05 (DEC) Code:HX/EB
    V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
000:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
    V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
060:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
    V V V V V V V V
120:E5E5E5E5E5E5E5E5

Drive: 1 Track: 01 (DEC) Sector: 06 (DEC) Code:HX/EB
    V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
000:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
```

DISKETTE NACH AUFNAHME DER
DATEI "TEXT" MIT COPY (VORHER ALLOC)

```
Disk-Edit  Rev.:001  From:00/01 (DEC) To:76/26 (DEC) Page:0001

 Drive: 1 Track: 00 (DEC) Sector: 01 (DEC) Code:HX/EB
000:40404040404040404040 40404040404040404040 40404040404040404040 40404040404040404040 40404040404040404040
060:40404040404040404040 40404040404040404040 00000000000000000000 00000000000000000000 00000000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 02 (DEC) Code:HX/EB
000:40404040404040404040 40404040404040404040 40404040404040404040 40404040404040404040 40404040404040404040
060:40404040404040404040 40404040404040404040 00000000000000000000 00000000000000000000 00000000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 03 (DEC) Code:HX/EB
000:40404040404040404040 40404040404040404040 40404040404040404040 40404040404040404040 40404040404040404040
060:40404040404040404040 40404040404040404040 00000000000000000000 00000000000000000000 00000000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 04 (DEC) Code:HX/EB
000:40404040404040404040 40404040404040404040 40404040404040404040 40404040404040404040 40404040404040404040
060:40404040404040404040 40404040404040404040 00000000000000000000 00000000000000000000 00000000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 05 (DEC) Code:HX/EB
    E R M A P
000:C5D9D4C1D7404040404040 40404040404040404040 40404040404040404040 40404040404040404040 40404040404040404040
060:40404040404040404040 40404040404040404040 00000000000000000000 00000000000000000000 00000000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 06 (DEC) Code:HX/EB
000:40404040404040404040 40404040404040404040 40404040404040404040 40404040404040404040 40404040404040404040
060:40404040404040404040 40404040404040404040 00000000000000000000 00000000000000000000 00000000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 07 (DEC) Code:HX/EB
    V O L 1
000:E5D6D3F140404040404040 40404040404040404040 40404040404040404040 40404040404040404040 40404040404040404040
                                              W
060:40404040404040404040 40404040404040404040E6 00000000000000000000 00000000000000000000 00000000000000000000
120:0000000000000000

 Drive: 1 Track: 00 (DEC) Sector: 08 (DEC) Code:HX/EB
    H D R 1   D A T A                                0 8 0   0 1   0 0 1   7 3 0 2 6
000:C8C4D9F140C4C1E3C140 40404040404040404040 40404040F0F8F0404F0F1 F0F0F140F7F3F0F2F640 40404040404040404040
                    .  .         0 1 0 0 1
060:40404040404040404040 40404040F0F1F0F0F140 00000000000000000000 00000000000000000000 00000000000000000000
120:0000000000000000
```

```
Disk-Edit  Rev.:001  From:00/01 (DEC) To:76/26 (DEC) Page:0002

Drive: 1 Track: 00 (DEC) Sector: 09 (DEC) Code:HX/EB
    H D R 1   T E X T                                1 2 8   0 5   0 0 1   0 5 0 1 0     B          7 9 0  9 1 0
000:C8C4D9F1404E3C5E7E340 4040404040404040404040 4040404040F1F2F8404F0F5 F0F0F1404F0F5F0F1F040 40C2404040404040F7F9F0 F9F1F0404040404040404040
                  7 9 0 9   1 1     0 5 0 0 4
060:4040404040404F7F9F0F9 F1F1404F0F0F5F0F0F440 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000

120:0000000000000000

Drive: 1 Track: 00 (DEC) Sector: 10 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 11 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 12 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 13 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 14 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 15 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 16 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 17 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 18 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 19 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 20 (DEC) Code:HX/EB
 ************     DELETED  DATA  RECORD     ************
```

SIEMENS

Schule für
Daten- und Informationssysteme
Systemservice-Ausbildung

B-22

```
Disk-Edit  Rev.:001  From:00/01 (DEC) To:76/26 (DEC) Page:0003

Drive: 1 Track: 00 (DEC) Sector: 21 (DEC) Code:HX/E8

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 22 (DEC) Code:HX/E8

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 23 (DEC) Code:HX/E8

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 24 (DEC) Code:HX/E8

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 25 (DEC) Code:HX/E8

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 00 (DEC) Sector: 26 (DEC) Code:HX/E8

 ************     DELETED  DATA  RECORD     ************

Drive: 1 Track: 01 (DEC) Sector: 01 (DEC) Code:HX/E8
    V V V V V V V V V V V V  V V V V V V V V V V V V  V V V V V V V V V V V V  V V V V V V V V V V V V  V V V V V V V V V V V V
000:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
    V V V V V V V V V V V V  V V V V V V V V V V V V  V V V V V V V V V V V V  V V V V V V V V V V V V  V V V V V V V V V V V V
060:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
    V V V V V V V V
120:E5E5E5E5E5E5E5E5      usw. bis Spur 5 / Sekt. 1
```

```
Disk-Edit  Rev.:001  From:05/01 (DEC) To:05/06 (DEC) Page:0001

 Drive: 1 Track: 05 (DEC) Sector: 01 (DEC) Code:HX/EB
     A N F A N G - 8 9 0  1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0
000:C1D5C6C1D5C760F8F9F0 F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0
     1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0
060:F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 4040404040404040404040 4040404040404040404040 4040404040404040404040 4040404040404040404040

120:40404040404040

 Drive: 1 Track: 05 (DEC) Sector: 02 (DEC) Code:HX/EB
     1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0
000:F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0
     1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0
060:F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 4040404040404040404040 4040404040404040404040 4040404040404040404040 4040404040404040404040

120:40404040404040

 Drive: 1 Track: 05 (DEC) Sector: 03 (DEC) Code:HX/EB
     1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 9 0  1 2 3 4 5 6 7 8 - D
000:F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F8F9F0 F1F2F3F4F5F6F7F860C4
     A T E I L A E N G E  = 2 4 0 - - E N D E
060:C1E3C5C9D3C1C5D5C7C5 7EF2F4F06060C5D5C4C5 4040404040404040404040 4040404040404040404040 4040404040404040404040 4040404040404040404040

120:40404040404040

 Drive: 1 Track: 05 (DEC) Sector: 04 (DEC) Code:HX/EB
     V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
000:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
     V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
060:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
     V V V V V V V V
120:E5E5E5E5E5E5E5E5

 Drive: 1 Track: 05 (DEC) Sector: 05 (DEC) Code:HX/EB
     V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
000:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
     V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V  V V V V V V V V V V
060:E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5 E5E5E5E5E5E5E5E5E5E5
     V V V V V V V V
120:E5E5E5E5E5E5E5E5     usw. bis Sp. 76 / Sekt. 26
```

SIEMENS

Schule für
Daten- und Informationssysteme
Systemservice-Ausbildung

R - 24

SECTOR	FILENAME	REC	BOE	EOE	P	CRDATE	EXDATE	EOD
8	DATA	080	01/01	73/26				01/01

DIR - AUSDRUCK NACH "INIT"

SECTOR	FILENAME	REC	BOE	EOE	P	CRDATE	EXDATE	EOD
8	DATA	080	01/01	73/26				01/01
9	TEXT	128	05/01	05/10		790910	790911	05/10

DIR - AUSDRUCK NACH "ALLOC"

SECTOR	FILENAME	REC	BOE	EOE	P	CRDATE	EXDATE	EOD
8	DATA	080	01/01	73/26				01/01
9	TEXT	128	05/01	05/10		790910	790911	05/04

DIR - AUSDRUCK NACH "COPY"

TRANSDATA 920-Kompatibilität

Auf der 6.610 können Floppy Disks erstellt und verarbeitet werden, die - mit gewissen Einschränkungen - kompatibel sind zu denen, die auf dem Erfassungsplatz TRANSDATA 920 bzw. IBM 3740 benutzt werden.

Diese Floppy Disks werden auf der 6.610 mit

INIT$:Jn: (n = Laufwerknummer)

initialisiert.

Bei der Initialisierung können bei Spurdefekten keine Ersatzspuren angelegt werden. Mit der Auslieferung der Version 1.6 des BS 1 und Version 5 des Monitors werden Floppy Disks mit aktiven Ersatzspuren korrekt behandelt.

Beim Initialisieren wird ein Dateibereich DATA eingerichtet. Weitere Dateien können mit dem Programm ALLOC erstellt werden.

Dateien auf 920/3740-kompatiblen Floppy Disks werden mit :Jn: Dateiname angesprochen.

Die physikalische Einheit :In: gewährleistet nicht volle 920/3740-Kompatibilität.

Bei Eingabe der physikalischen Einheit :Jn: findet automatische Code-Umwandlung von ASCII in EBCDIC statt, da TRANSDATA 920 bzw. IBM 3740 im EBCDI- Code arbeiten.

Folgende Dienstprogramme erkennen die physikalische Einheit :Jn:

INIT, ALLOC, COPY, DIR, EDIT

Mit dem Editor EDIT kann man direkt auf 920/3740-kompatible Floppy Disks schreiben, hat aber zu berücksichtigen, daß die 920 bzw. 3740 nur Großbuchstaben kennt, während der Editor auch Kleinbuchstaben zuläßt.

	I	II	III
Datenträger-kennsatz			
Datenträgerkenn-satzinformation			x
Zugriffsschutzfeld			x
physik. Sektorlänge			x
Sektorfolge			x
Dateikennsatz			
Log.Satzlänge		x	
Bereichsbeginn	x		
Bereichsende	x		
Übergehen der Datei			x
Zugriffsmöglich-keit			x
Schreibschutz		x	
Mehrplattendatei-anzeiger			x
Fortlaufende Nummer			x
Schreibdatum	x		
Freigabedatum		x	
Prüfzeichen			x
Datenende	x		

I = Bedeutung wie bei 920/3740

II = Auf der 6.610 veränderbar aber ohne Bedeutung

III = Auf der 6.610 nicht beeinflußbar und ohne Bedeutung

Commercial Basic wird 920/3740-Dateien erst ab Version 1.5 voll unterstützen, in Version 1.4 wird die Einheit :Jn: noch nicht erkannt. Trotzdem können 920/3740-kompatible Dateien erstellt und gelesen werden, indem man

- grundsätzlich 80 Byte lange Sätze ohne CRLF wegschreibt
- 920/3740-Dateien mit INCHAR liest.

Die 6.610 verarbeitet alle abdruckbaren Zeichen der 920/3740 mit folgenden Ausnahmen:

#	wird auf der 6.610 als £ dargestellt
¢	(Cent-Zeichen) bewirkt auf der 6.610 Invertierung des Bildschirms

Der Editor EDIT verarbeitet Deleted Data Records (DDR, gelöschte Datensätze) im zweiten Leseversuch. Beim Dienstprogramm COPY führt das Auftreten von DDR auf einen Fehler.

In der folgenden Übersicht wird dargestellt, welche Eintragungen im Daten- bzw. Dateiträgerkennsatz auf der 6.610 möglich sind und welche ignoriert werden.

Magnetband-Kassettenformat im BS 610

Die Spuren haben die Nummern 0 - 3. Spur 0 wird als Inhaltsverzeichnis benutzt, die Spuren 1 - 3 für Daten. Bei einer Blocklänge von 128 Bytes ergibt sich pro Kassette eine Datenkapazität von 750 KBytes.

Spur 0 enthält:

- einen Datenträger-Kennsatz (1 Block)
- einen Inhaltsverzeichnisblock für jede auf der Kassette vorhandene Datei (gleichgültig ob existent oder gelöscht)
- einen Kassetten-Belegungsblock
- Abschnittsmarke

Jede der Spuren 1 - 3 enthält eine Folge von Dateisegmenten. Diese bestehen aus:

- einem Dateikennsatz (1 Block)
- einer Anzahl von Datenblöcken
- Abschnittsmarke
- einem Dateiendeblock
- Abschnittsmarke

Nach dem letzten Dateisegment einer Spur folgen

- ein Spurendeblock
- Abschnittsmarke

Normalerweise besteht eine Datei aus nur einem Segment, es sei denn, daß während des Schreibens der Datei das Spurende erreicht wird. In diesem Fall wird auf der nächsten Spur weitergeschrieben, auf der Platz frei ist.
Um zu gewährleisten, daß die Datei später einwandfrei gelesen werden kann, wird an der Abbruchstelle ein besonderer Dateiende(=Fortsetzungs-)block und vor dem Rest der Datei ein Fortsetzungs-Kennsatz geschrieben. In jedem Fall ist eine derartige Fortsetzung sowohl beim Lesen wie beim Schreiben zeitaufwendig und sollte tunlichst vermieden werden.

Die Blocklänge kann vom Anwender bestimmt werden, Minimum ist 32, Maximum 2048 Bytes. Die verschiedenen Kontrollblöcke (Datenträger-Kennsatz, Datei-Inhaltsverzeichnis, Belegungsverzeichnis, Dateikennsatz, Dateiendeblock, Spurendeblock) haben eine feste Länge von 32 Bytes.

Formate der Kontrollblöcke

Datenträger-Kennsatz

Byte		
	0	88H
	1 - 9	Name des Datenträgers
	10 - 31	unbenutzt

Datei-Inhaltsverzeichnis

Byte		
	0	0: Datei existiert FFH: Datei gelöscht
	1 - 9	Dateiname
	10	Spurnummer des 1.Segments
	11	Dateinummer des 1.Segments
	12 - 13	Blocklänge (min. 32, max. 2048 Bytes) (die niedrigst-signifikanten 8 Bits im Byte 12)
	14 - 15	Anzahl Bytes im letzten Block (die niedrigstsign. 8 Bits im Byte 14)
	16 - 17	Anzahl Blöcke in der Datei (die niedrigstsign. 8 Bits im Byte 16)
	18	Attributbyte
	19 - 31	unbenutzt

Belegungsverzeichnis

Byte		
	0	7FH
	1 - 5	Belegung der Spur 1 1 - Anzahl Dateisegmente 2/3 - Anzahl Blöcke auf der Spur (die niedrigstsign. 8 Bits im Byte 2) 4 - Spurfüllungsanzeiger: = 0: voll 5 - unbenutzt
	6 - 10	Belegung der Spur 2 Format wie Spur 1
	11 - 15	Belegung der Spur 3 Format wie Spur 1
	16 - 31	unbenutzt

Dateikennsatz

Byte		
	0	'(' beim ersten Dateisegment
	1 - 9	Dateiname
	10	Segmentnummer (erstes = 0)
	11 - 12	Blocklänge
	31 - 31	unbenutzt

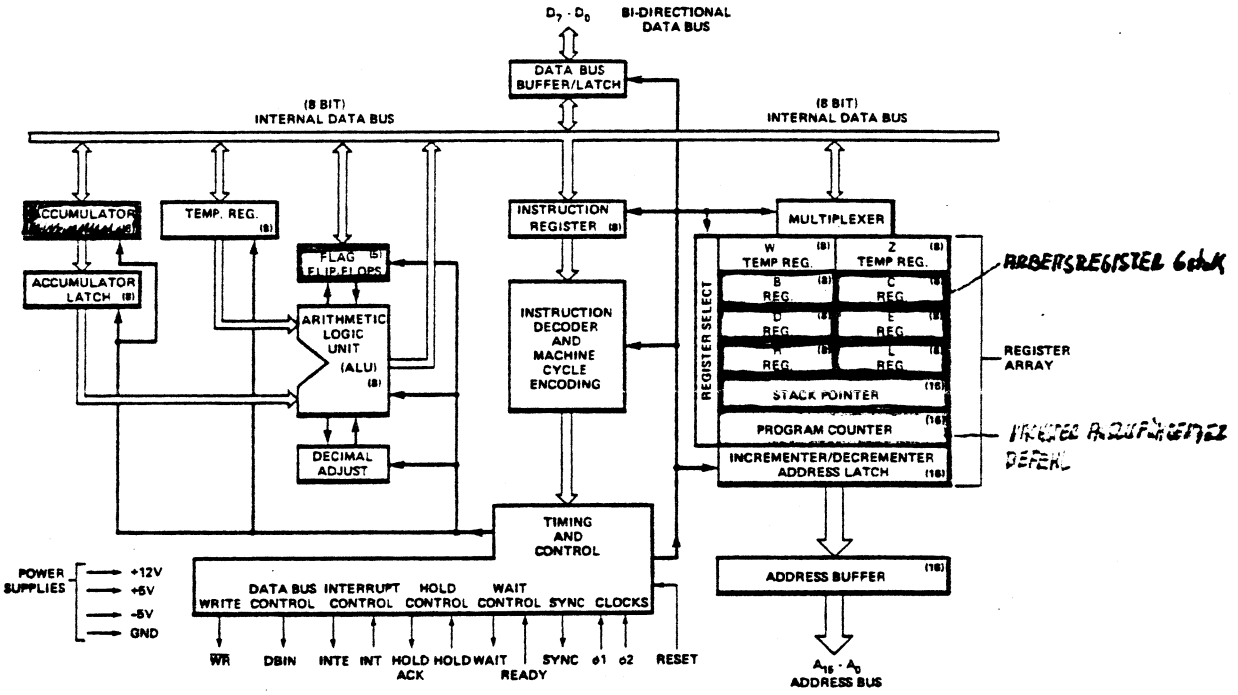
Dateiendeblock

Byte	0	`)" bei letztem Segment `+" wenn die Datei fortgesetzt wird
	1 - 9	Dateiname
	10	Nummer des nächsten Segments (0, wenn es das letzte ist)
	11	Fortsetztungsspur (0, wenn Dateiende)
	12 - 31	unbenutzt

Spurendeblock

Byte	0	`/`
	1 - 31	unbenutzt

A S S E M B L E R 6 . 6 1 0



BLOCKSCHALTBILD: MIKROPROZESSOR 8080 A

CPU 8080

Arbeitsregister

Der Mikroprozessor 8080 beinhaltet 6 Arbeitsregister, die für den Benutzer zugänglich sind.

Register B,C,D,E,H,L

Diese Register können sowohl einzeln (für 8 - Bit - Operationen), als auch paarweise (für 16 - Bit - Operationen) verwendet werden.
Letzteres ist insbesonders für die Arbeitsspeicher - Adressierung erforderlich.
Die Register B/C, D/E und H/L bilden jeweils ein Paar, sie werden dann mit B,D,H angesprochen.

ACCUMULATOR

8 - Bit - Register
Der Akku (Register A) wird als Universalregister verwendet.
Viele Operationen können nur über den Akku durchgeführt werden.

FLAG

Für die bei Befehlsausführung entstehenden Anzeigen (z.B. Ergebnis eines Vergleiches) wird das Anzeigeregister verwendet.

STACK POINTER

16 - Bit - Register
Der Stapelanzeiger weist auf einen Arbeitsspeicherbereich, der als Kellerspeicher benutzt wird.

PROGRAM COUNTER

16 - Bit - Register
Der Befehlszähler enthält immer die Arbeitsspeicheradresse, auf der der nächste von der CPU auszuführende Befehl steht.

ARITHMETIC - LOGIC UNIT

Die ALU führt alle arithmetischen und logischen Operationen durch (z.B. Addieren,Exclusiv oder).Sie besitzt einen Zusatz zur Dezimalkorrektur.

TIMING AND CONTROL

Die Ablaufsteuerung übernimmt die Ausführung von Befehlen, Steuerung des Daten - und Adressbus, sowie die Verbindung zur Außenwelt der CPU.(E/A - Bausteine etc.)

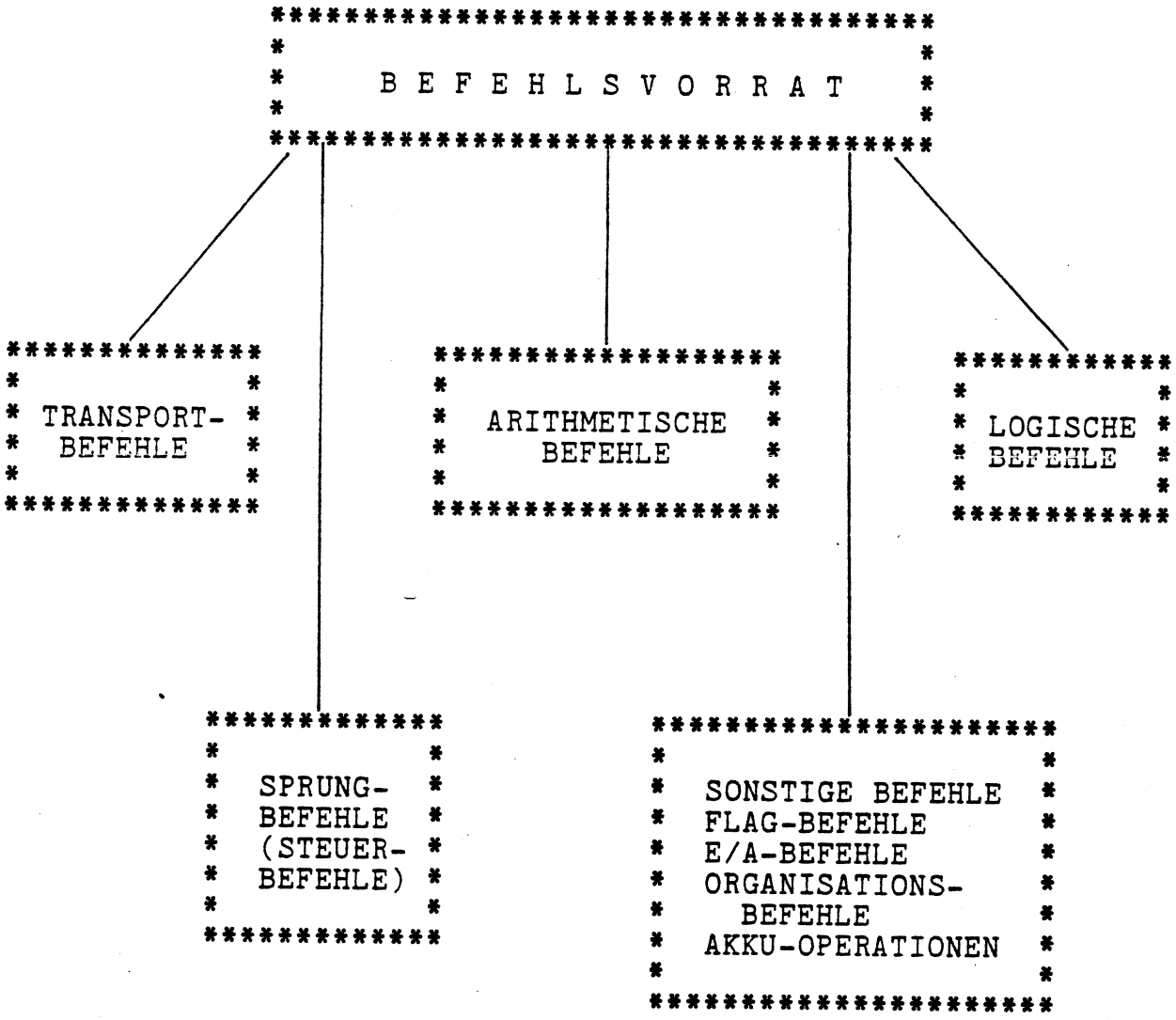
BI - DIRECTIONAL DATA BUS

Der Datenbus ist 8 Bit breit, d.h. 1 Byte für Daten-Ein - und Ausgabe.

ADRESS BUS

Der Adressbus ist 16 Bit breit. Durch ihn werden die E/A - Bausteine, sowie die diversen Speicher adressiert.

Anmerkung: die CPU 8080 arbeitet völlig codefrei.


B E F E H L S V O R R A T
TRANSPORT-BEFEHLE
ARITHMETISCHE BEFEHLE
LOGISCHE BEFEHLE
SPRUNG-BEFEHLE (STEUER-BEFEHLE)
SONSTIGE BEFEHLE
FLAG-BEFEHLE
E/A-BEFEHLE
ORGANISATIONS-BEFEHLE
AKKU-OPERATIONEN

Der ASSEMBLER ist ein Programm, das ein in maschinenorientierter, mnemotechnischer Sprache erstelltes Programm in maschinenlesbaren Code (binär verschlüsselte Befehle) umsetzt.
Die Umschlüsselung erfolgt 1:1, d.h. ein Befehl in mnemotechnischer Sprache entspricht einem Befehl in Maschinensprache.

z.B. mnemotechnische Anweisung	Maschinen - Befehl	Beschreibung
MVI A,41H	3E41	die Konstante "41"(hex) in den Akku transportieren

Um dem Programmierer eine Arbeitsspeicher - Adressrechnung zu ersparen, wird beim Assembler eine symbolische Adressierung verwendet. So kann ein Arbeitsspeicherplatz aufgrund dieser symbolischen Adresse von beliebigen Stellen des Programms einfach angesprungen werden.

Für das fehlerfreie Arbeiten mit dem ASSEMBLER muß der Sprachsyntax unbedingt eingehalten werden.

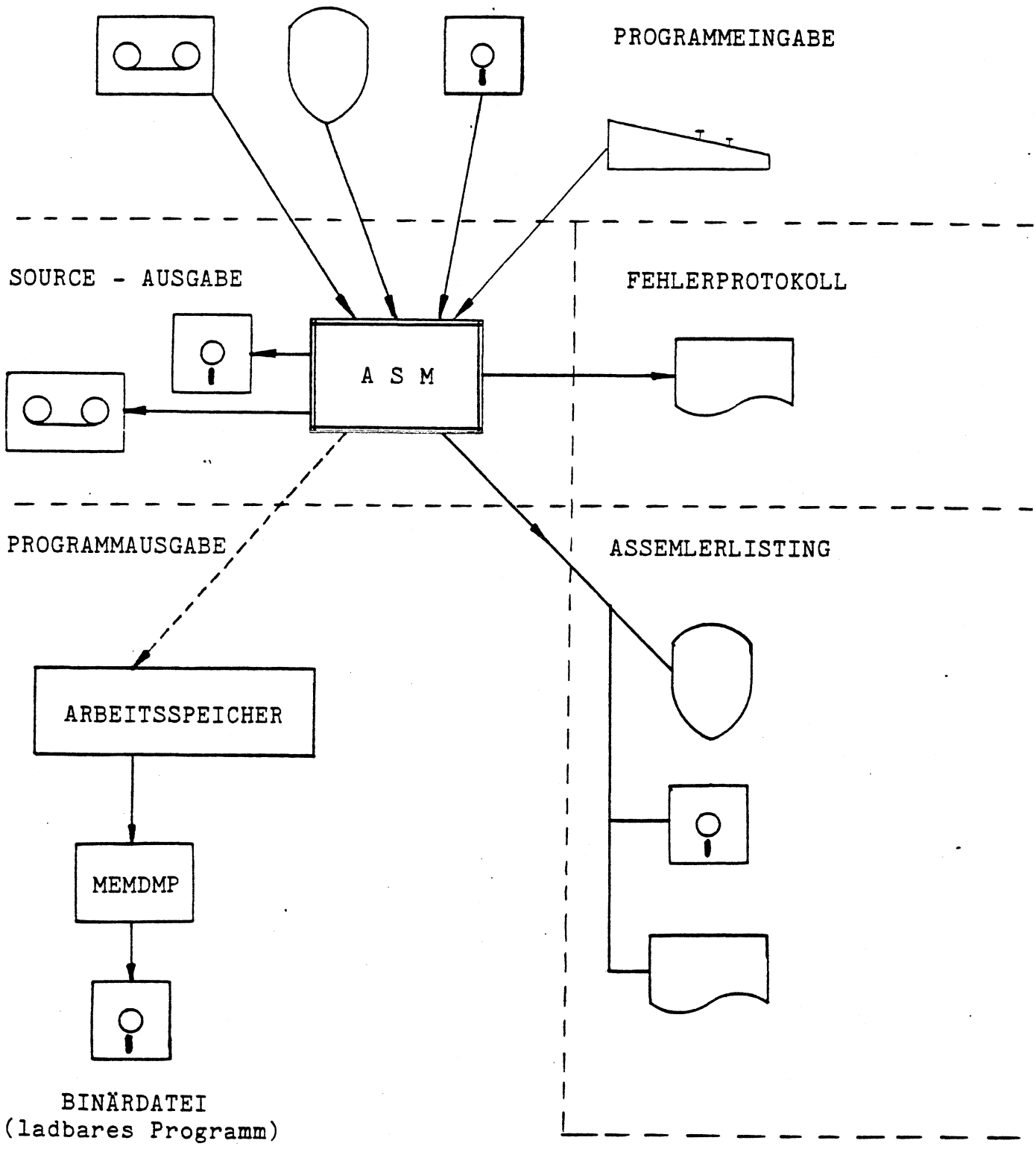
Für die Übersetzung eines in mnemotechnischer Sprache (Assemlersprache 8080) erstellten Programmes stehen im Betriebssystem AMBOSS 1 zwei ASSEMBLER zur Verfügung.

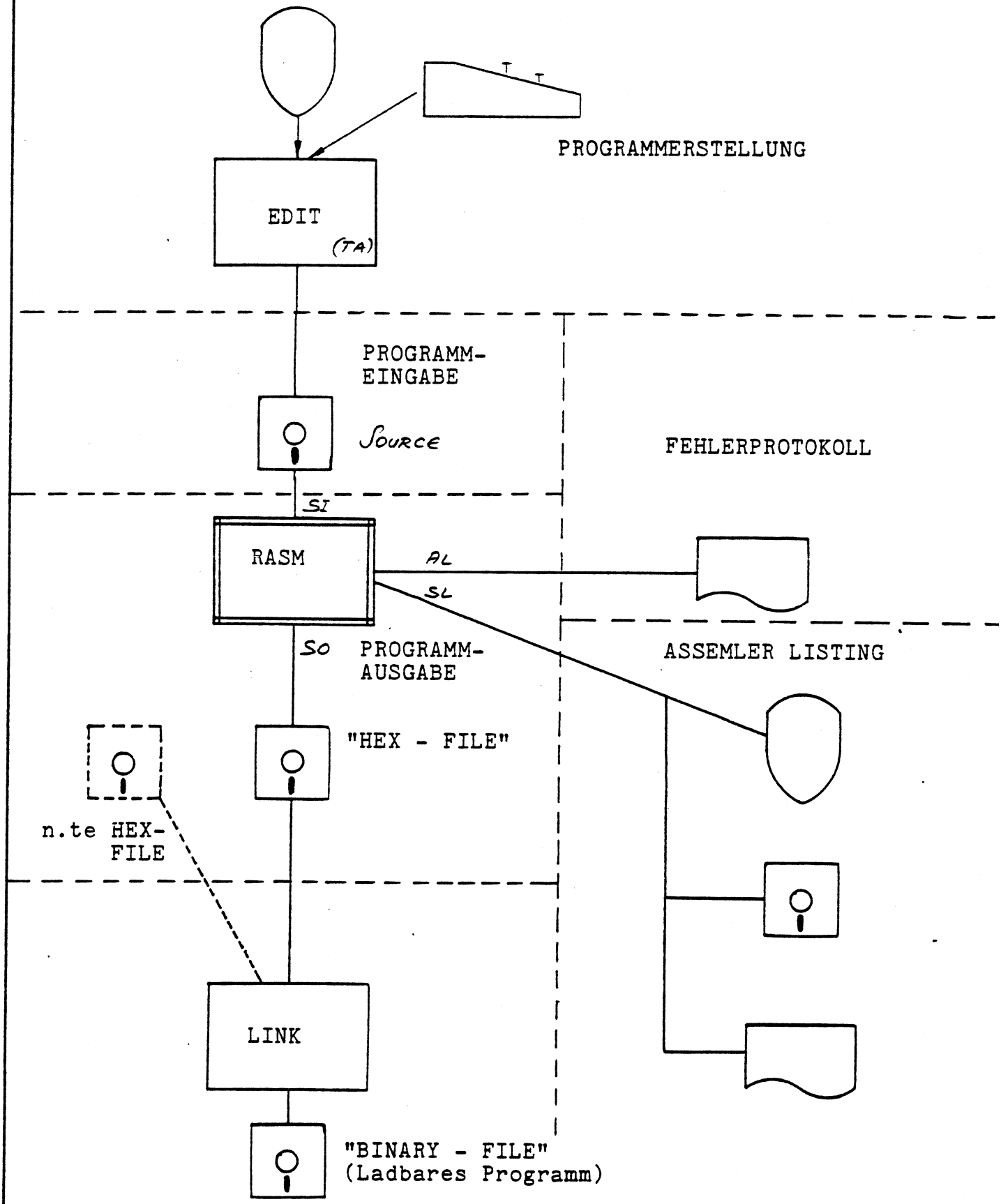
1. A S M

Der Übersetzer ASM ist ein sog. "Festassembler", d.h. das Übersetzungsergebnis (Programm) ist an die einmal festgelegten Arbeitsspeicheradressen gebunden.
Der ASM kann als Dialogassemler verwendet werden. Er ist sehr gut geeignet, die Assemblersprache zu erlernen.

2. R A S M

Der Übersetzer RASM ist ein "RELOCATABLE - Assembler", d.h. ein mit ihm übersetztes Programm ist später im Arbeitsspeicher verschiebbar.
Er ermöglicht zusätzlich externe Adressierung, um mehrere Programmteile oder Programme zu einem ablauffähigen Programm zusammenzufügen. Dieses Zusammenfügen nennt man "Binden".
Das Programm dafür ist der Binder "LINK".


PROGRAMMEINGABE
SOURCE - AUSGABE
FEHLERPROTOKOLL
A S M
PROGRAMMAUSGABE
ASSEMLERLISTING
ARBEITSSPEICHER
MEMDMP
BINÄRDATEI
(ladbares Programm)


PROGRAMMERSTELLUNG
EDIT
(TA)
PROGRAMM-
EINGABE
Source
FEHLERPROTOKOLL
SI
RASM
AL
SL
SO
PROGRAMM-
AUSGABE
ASSEMLER LISTING
"HEX - FILE"
n.te HEX-
FILE
LINK
"BINARY - FILE"
(Ladbares Programm)

Dialog-ASSEMBLER (ASM)

```
ASM SI=Input-file  SO=Hex-Output-file
    SL=List-file  AO=Source-Output-file
    AL=Error-list-file
```

Beispiel1: Bildschirmeingabe, Source-Ausgabe (Dialog-Assembler)

```
ASM AO=PROG.SRC
```

Beispiel2: Übersetzung aus Source-file

```
ASM SI=PROG.SRC,
    SO=PROG.HEX,
    SL=:F1:PROG.LST
    AL=:CO:
```

Relocatable ASSEMBLER (RASM)

```
RASM SI=Input-file ,SO=hex-output-file
     ,SL=listfile ,AL=error-list-file
```

Beispiel1:

```
RASM  SI=PROC.SRC,SO=PROG.HEX,SL=:LP:
```

Beispiel2: nur Übersetzung um ein Listing zu erhalten und auf einem anderen Gerät (mit Drucker) auszugeben

```
RASM SI=PROG.SRC,SL=PROG.LST
MOVE SI=PROG.LST,SO=:LP:
```

Binder LINK

```
LINK [CI=Binder-Kommando-Datei]
     SO=binär-file
     [SL=list-file]
```

Binderkommandos:

=C = XXXX	Ladeadresse eines Programm-Moduls
=D = XXXX	Ladeadresse eines Datenmoduls
=S = XXXX	Startadresse des Programms
=Q	Abbruch
/	Ende der Eingaben

Die einzelnen Module werden in der Form :Fn:Hex-file angegeben.

```
Beispiel:  LINK SO=PROG,SL=:LP:
         > =C = 4000
         > PROG.HEX
         > =S = 4000
         > /
```

Die 61ø Assemblersprache umfaßt drei verschiedene Gruppen von Assemblerinstruktionen:

1) Assembler Anweisungen
2) Daten-Definitionen
3) Befehle

Anweisungen steuern den Übersetzungsvorgang
Definitionen hinterlegen Konstanten im Arbeitsspeicher
Befehle werden in maschinenlesbare Binärinformationen übersetzt.

Zeichenvorrat:

Buchstaben: A-Z
a-z
Ziffern: 0-9
Sonderzeichen: Blank,!,",# ,$,%,',&,(,),
*,+,,,-,.,/,:,;,<,=,>,?,
@,^,`,|

Achtung: Einige Sonderzeichen haben spezielle Bedeutung! STEUERBEFEHLE

Wertevorrat:

Dezimalzahlen: 1 Byte 0 - 255
+0 - +127
-1 - -128

2 Byte 0 - 65535
+0 - +32767
-1 - -32768

Oktalzahlen: 0-7 Q

Sedezimalzahlen: 0-9 und A-F H

MVI A, Konst

10 → Dez.
10H → Sed.
7Q → OKT
1M000B → BIN

} ASS — Maschinenspr. Hex.

Eine Assemblersprachenzeile (Statement) besteht aus maximal 4 verschiedenen Feldern:

1) Namensfeld (symbolische Adressen)
2) Operationsfeld
3) Operandenfeld
4) Bemerkungsfeld

Die einzelnen Felder werden, wenn vorhanden, durch sog. 'Fluchtsymbole' voneinander getrennt.

Fluchtsymbol :	trennt das Feld Symbolische Adresse vom Operationsfeld
Fluchtsymbol ␣	trennt Operationsfeld vom Operandenfeld
Fluchtsymbol ;	trennt Operandenfeld vom Bemerkungsfeld
Fluchtsymbol ,	trennt die einzelnen Operanden
Fluchtsymbol $	ist ein spezielles Fluchtsymbol.

Anmerkung:
Nach einem Fluchtsymbol (mit Ausnahme von $) dürfen beliebig viele Blanks folgen.

1) Namensfeld

Das Namensfeld wird benutzt, um einem Speicherplatz (Befehl, Konstante) einen symbolischen Namen zuzuordnen, auf den man sich beziehen möchte. Zulässige Zeichen sind alle Buchstaben und Ziffern, sowie die Sonderzeichen @ und ?. Das 1. Zeichen muß auf jeden Fall ein Buchstabe, @ oder ? sein.
Das Namensfeld ist wahlfrei verwendbar. Namen dürfen nicht Operationscode sein.

2) Operationsfeld

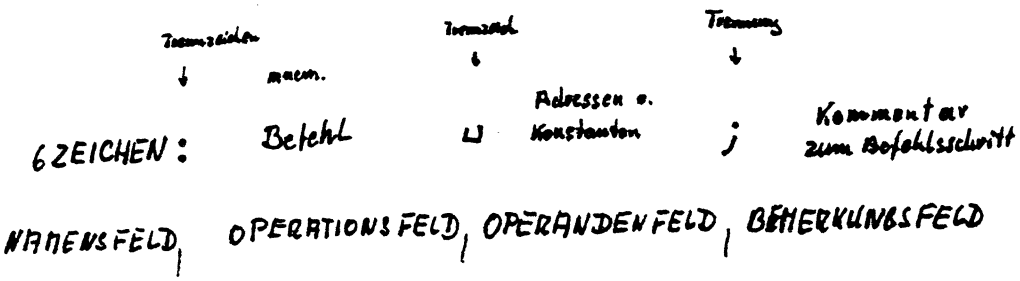
In das Operationsfeld werden die Assemblerinstruktionen eingetragen. Als Instruktionen gelten:
1) Anweisungen
2) Daten-Definitionen
3) Befehle

3) Operandenfeld

Werden bei einer Operation noch Operanden, wie Registerangaben, Speicheradressen usw., benötigt, so werden diese in das Operandenfeld eingetragen. Mehrere Operanden werden durch Kommas getrennt.

4) Bemerkungsfeld

Das Bemerkungsfeld dient nur zu Dateninformetionszwecken. Alle Zeichen des USASCII Zeichenvorrates sind zulässig.



Assembleranweisungen

$ TITLE ␣ Überschrift

1) Seitenüberschrift
   $TITLE␣Text
   Dieser Text wird in die Kopfzeile jeder Seite projeziert.

2) Papiersteuerung
   a) $PW dezimal Zahl
      Mit der $PW-Anweisung kann die Zeilenlänge festfelegt werden. Standardmäßig wird 80 angenommen.
   b) $PL dezimal Zahl
      Mit der $PL-Anweisung kann die Papierlänge in Zeilen festgelegt werden. Standardmäßig wird 72 angenommen.
   c) $E
      Mit der $E Anweisung wird sofortiger Papiervorschub durchgeführt

3) Ausdrucken der kompletten DB-Definition
   $DB

Datendefinitionen

1) Definieren von Bytes

NAME: DB byte1,Byte2,.....,Byten

Beispiel:

START: DB 074H,12H,20,17Q,11011000B

TEXT : DB "TEXTAUSGABE"

2) Wortdefinitionen

Die Anweisung DW definiert immer 2 Byte

NAME: DW Wort1,Wort2,....Wortn

Beispiele:

3) Bereiche Definieren

NAME: DS Wert

Beispiele:

4) Gleichsetzen von Namen, Adressen und Worten

NAME EQU Ausdruck

Beispiele:

TTI EQU 0040H

CALL TTI

5) Verändern des Adresspegels

NAME: ORG Ausdruck

Beispiele:

Achtung: Wird ein Programm mit einer festen Zuweisung z.B.: ORG 4000H begonnen, so ist es nicht mehr verschiebbar.
ORG ist nur auf fest Adresspegel möglich.

Akkumulator (Register A)

Alle arithmetischen und boolschen Operationen lassen sich nur mit Hilfe des Akkus durchführen. Der Akku ist deshalb das wichtigste Byte-Register für den Benutzer. Der Befehlsvorrat der 8080-CPU bringt es mit sich, daß man den Akkuinhalt sehr häufig zwischenspeichern muß, um ihn für eine weitere Operation wieder frei zu haben. Es existieren deshalb eine Reihe Lade- und Speicherbefehle direkt für den Akku.

Beispiel: STA SAVE Speichert Akkuinhalt in die Zelle, die mit SAVE adressiert ist

LDA SAVE Lädt den Akku mit dem Inhalt der Zelle, die mit SAVE adressiert ist.

Das Flag-(Anzeigen-) Register

Für das Verzweigen eines Programmes nach einem Vergleichsbefehl oder für bedingte Sprünge nach arithmetischen bzw. boolschen Operationen, muß die CPU für den Anwender Anzeigen (z.B. 1.Operand>2.Operand) hinterlassen. Diese Informationen werden im FLAG-Register hinterlegt.

Bis auf das Übertragsbit werden alle Bits ausschließlich durch den Befehlsablauf verändert. Das Übertragungsbit kann per Befehl verändert werden. Über die Befehle PUSH und POP können jedoch alle veränderbaren Bits verändert werden.

Aufbau des Flag-Registers

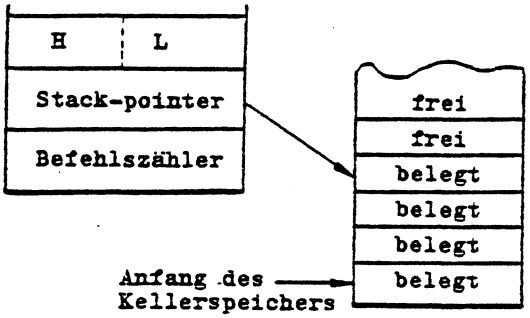
$2^0 = 0$	Kein Übertrag aus Bitstelle 2^7	CY
$2^0 = 1$	Übertrag aus Bitstelle $2^7 \rightarrow 2^8$	CY
$2^1 = 1$	frei (immer 1)	
$2^2 = 0$	Paritätsbit (Anzahl Bits) ungerade	P
$2^2 = 1$	Paritätsbit (Anzahl Bits) gerade	P
$2^3 = 0$	frei (immer 0)	
$2^4 = 0$	Kein Übertrag aus Bitstelle 2^3	AC
$2^4 = 1$	Übertrag aus Bitstelle $2^3 \rightarrow 2^4$	AC
$2^5 = 0$	frei (immer 0)	
$2^6 = 0$	Ergebnis ist ungleich 0	Z
$2^6 = 1$	Ergebnis ist gleich 0	Z
$2^7 = 0$	Ergebnis Bit 2^7=0 (positiv)	S
$2^7 = 1$	Ergebnis Bit 2^7=1 (negativ)	S

Registerpaar H/L

Das Registerpaar H/L ist neben dem Akkumulator das wichtigste Doppel-Register, da es in den meisten Speicherbefehlen die Arbeitsspeicheradresse enthält. Es wird in diesen Befehlen meist nicht direkt angegeben, sondern mit M (Memory) gekennzeichnet.

Beispiel: MOV a,M; Laden des Akkus mit dem Inhalt der Speicherzelle, die durch H/L adressiert ist.

Der Kellerspeicher ist ein Teil des Arbeitsspeichers. Er wird von der CPU durch den Stapelanzeiger (Stack-Pointer) verwaltet. Der Kellerspeicher wird abwärts belegt und aufwärts freigegeben.



Der Stack-pointer weist immer auf die ASP-Zelle mit der letzten eingeschriebenen Information.

Mit dem Befehl

SPLH		Kann der Stack-Pointer geladen werden

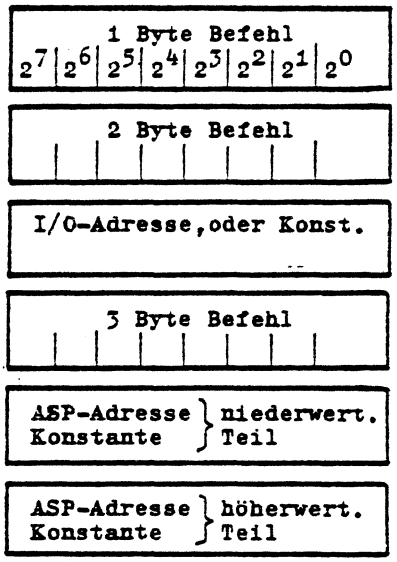
Mit der Befehlsfolge

LXI	H,0	
DAD	SP	kann der Inhalt des Stack-Pointers nach H/L gebracht werden

Hinweis: Im BS61Ø beginnt der Kellerspeicher auf der ASP-Adresse $25BE_{(16)}$

Befehlsaufbau

Im Befehlsvorrat der 8ØSØ CPU sind unterschiedlich lange Befehle enthalten. Die Länge liegt zwischen 1 und 3 Byte. Das 1. Byte enthält in jedem Fall eine binäre Verschlüsselung des Operationscodes. Zusätzlich kann in diesem Byte noch die Verschlüsselung von einem oder zwei Registern, sowie die Verschlüsselung eines Registerpaares enthalten sein. Aus diesem Grund kann aus dem 1.Byte eines Befehls nicht ohne weiteres auf den Befehl und/oder dessen Länge, sowie die Verwendung von Registern geschlossen werden.



Registerverschlüsselung:

Einzelregister

ØØØ = B	1ØØ = H
ØØ1 = C	1Ø1 = L
Ø1Ø = D	11Ø = M (H/L)
Ø11 = E	111 = A (Akku)

Registerpaar

ØØ = B/C
Ø1 = D/E
1Ø = H/L
11 = Stack-Pointer

Arbeitsspeicheradressierung

Der Arbeitsspeicher ist Byte orientiert, d.h. es gibt keine Ausrichtung auf Wortgrenzen. Im Arbeitsspeicher werden Befehle (jeweils 1 Byte = Operationscode), Adressen (jeweils 2 Byte bilden eine Adresse) und Daten (entweder 1 oder 2 Byte-Daten) abgelegt.

Beispiel: Laden des Registerpaares D/E mit der Anfabgsadresse einer Dezimaltabelle

LAD:	LXI D,	DEZTAB	
	11	8B |	17
	Ø1ØØ	Ø1Ø1	Ø1Ø2

Der symbolischen Adresse LAD entspricht die sedezimale Adresse Ø1ØØ

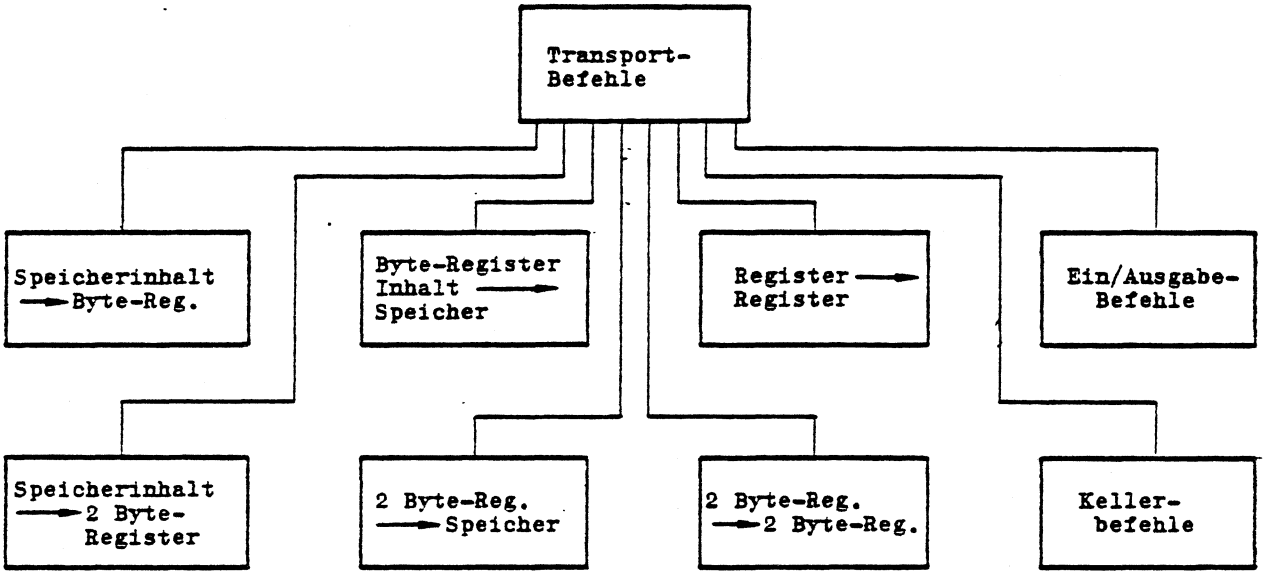
Nach Ausführung des Befehls enthält:

Register D = 17
Register E = 8B

DEZTAB:			
	DW 1	≙ ØØØ1	Der symbolischen Adresse DEZTAB entspricht die sedezimale Adresse 178B
	Ø1 | ØØ		
	178B 178C		
	DW 1Ø	≙ ØØØA	
	ØA | ØØ		
	178D 178E		
	DW 1ØØ	≙ ØØ64	
	64 | ØØ		
	178F 179Ø		
	DW 1ØØØ	≙ Ø3E8	
	E8 | Ø3		
	1791 1792		
	DW 1ØØØØ	≙ 271Ø	
	1Ø | 27		
	1793 1794		

Achtung: Bei Worten sind die beiden Bytes, die ein Wort bilden im ASP vertauscht.

Transportbefehle



Allgemeines:

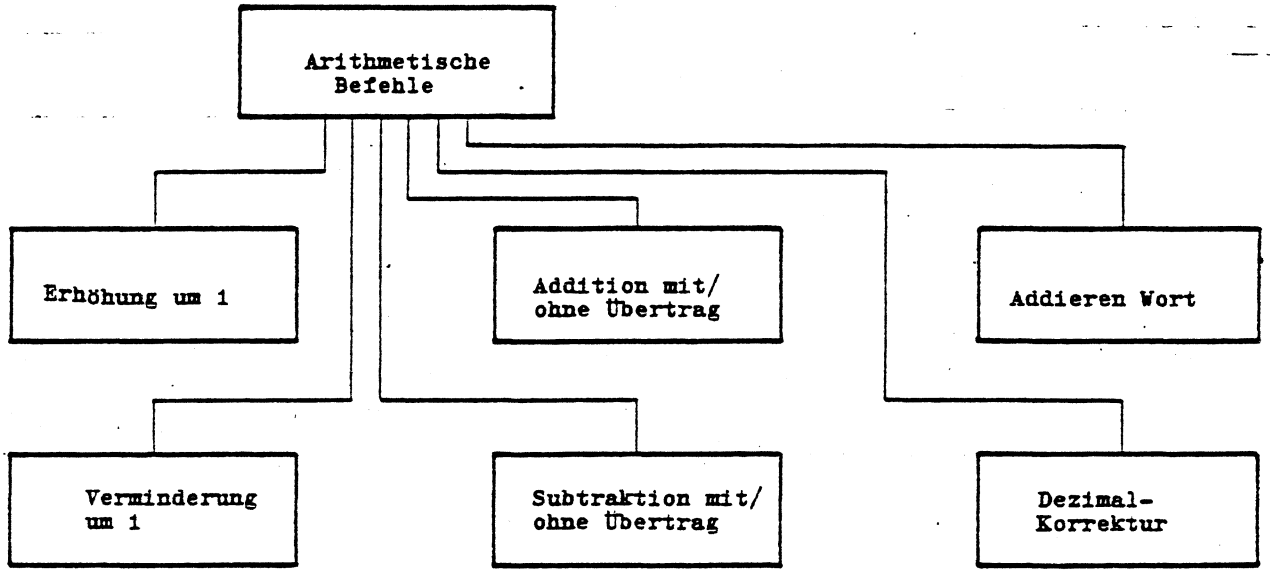
Für die Adressierung des Arbeitsspeichers werden 16 Bit benötigt. Generell bestehen zwei Möglichkeiten, eine ASP-Adresse anzugeben:

1.) Direkt hinter dem Befehl werden 2 Byte mit der ASP-Adresse belegt
(z.B.: LXI B,TAB)

2.) Ein Registerpaar enthält bereits die ASP-Adresse
(z.B.: MOV B,M)

Bei keinem der Transportbefehle (Ausnahme: POP) werden die Anzeigen (Flag-Register) verändert.

Arithmetische Befehle



Allgemeines:

Mit wenigen Ausnahmen wird bei den arithmetischen Befehlen der Akkumulator als 1. Operand und Ergebnisregister benutzt.
Nur ein Befehl (DAD) kann 2-Byteweise addieren.
Ein Befehl (DAA) beschäftigt sich mit einer Dezimalkorrektur.
Bei Addition und Subtraktion wird grundsätzlich binär gearbeitet.
Die Subtraktion wird auf die Addition mit dem 2er-Komplement zurückgeführt.
Die arithmetischen Befehle hinterlassen Anzeigen im Flag-Register, die der Anwender auswerten und interpretieren muß (z.B.: Überlauf, Vorzeichenwechsel usw.)

Rechenregeln der Binärarithmetik
(für SAB 3080A)

1.) Arithmetik mit vorzeichenlosen Binärzahlen

Wertebereich : Ø - 255

a) Auswertung bei Addition und Subtraktion

CY = 1 Bereichsüberlauf
CY = Ø Ergebnis im Wertebereich

2.) Arithmetik mit vorzeichenbehafteten Binärzahlen

Wertebereich : +Ø - +127
-1 - -128

Für die Auswertung des Ergebnisses wird der Übertrag aus Bitstelle 2^6 in Bitstelle 2^7 benötigt. Dieser sog. C2-Übertrag wird jedoch von der 8080-CPU nicht im Flagregister abgelegt. Er muß daher vor der eigentlichen arithmetischen Operation erzeugt werden.

Die Auswertung ist bei Addition und Subtraktion identisch:

C1 = C2 Ergebnis im Wertebereich
S = Ø Ergebnis positiv
S = 1 Ergebnis negativ

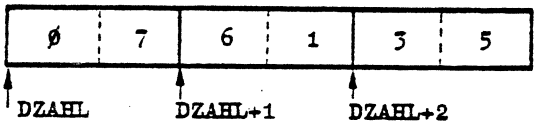
C1 ≠ C2 Bereichsüberlauf
S = Ø negativer Bereichsüberlauf
S = 1 positiver Bereichsüberlauf

Die Auswertung ist bei Addition und Subtraktion identisch:

Befehl DEZIMAL-KORREKTUR

Der Befehl Dezimalkorrektur ist für die Bearbeitung von Dezimalzahlen vorgesehen. Dabei wird davon ausgegangen, daß die Zahlen im BCD-Code vorliegen, wobei jeweils 2 Stellen im Byte belegt werden.

Beispiel: $76135_{(10)}$



Werden zwei im BCD-Code vorliegende Dezimal Zahlen z.B.: addiert, so muß das Ergebnis korrigiert werden, da die Addition von der CPU nur binär durchgeführt wird.

CY = Ø, AC = Ø

1 Ø Ø 1	1 Ø Ø 1	9 9
+ 1 Ø Ø 1	1 Ø Ø 1	9 9
Ø Ø 1 1	Ø Ø 1 Ø	32
CY = 1	AC = 1	

Das durch diese Addition gefundene Ergebnis ist falsch. Es muß mittels des Befehls DAA korrigiert werden.
Der Befehl DAA korrigiert ein Byte (2 Dezimalstellen) im Akku in zwei Schritten, wobei bei jedem Schritt ein Halbbyte bearbeitet wird.

Schritt 1: Eine Bearbeitung im Schritt 1 erfolgt nur dann, wenn entweder das rechte Halbbyte größer als 9 ist, oder aber das Hilfscarry-Bit gesetzt ist. In beiden Fällen wird ..Ø6 zum Akku hinzuaddiert. Das Hilfscarry-Bit wird nur bei Übertrag aus der Stelle 2^3 gesetzt. Entsteht kein Übertrag, wird es rückgesetzt. Das Carry-Bit bleibt unverändert.

CY = 1, AC = 1

ØØ11	ØØ1Ø	
+ ØØØØ	Ø11Ø	da AC = 1
ØØ11	1ØØØ	

CY = 1, AC = Ø

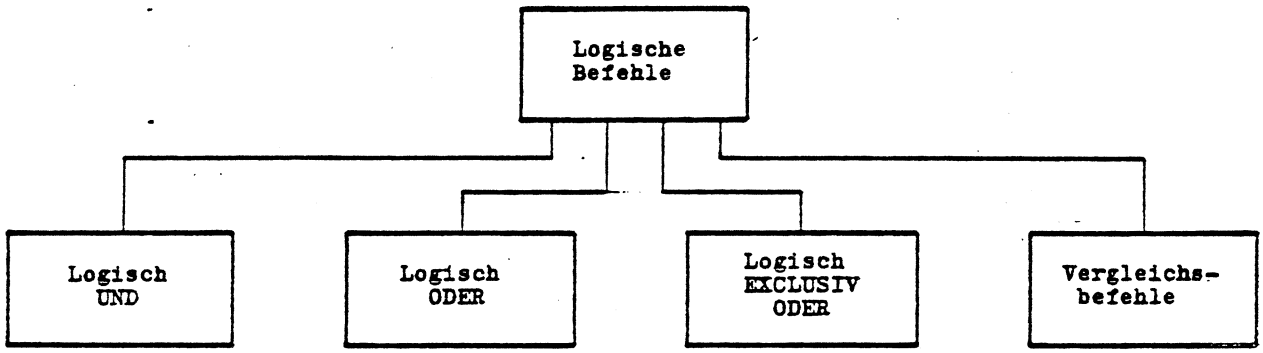
Schritt 2: Eine Bearbeitung im Schritt 2 erfolgt nur dann, wenn entweder das linke Halbbyte größer als 9 ist, oder aber das Carry-Bit gesetzt ist. In beiden Fällen wird 6Ø zum Akku hinzuaddiert. Das Hilfcarry-Bit wird nicht mehr beeinflußt. Das Carry-Bit kann nicht rückgesetzt werden. Es wird jedoch bei Überlauf aus 2^7 gesetzt.

CY = 1, AC = Ø

ØØ11	1ØØØ	
+ Ø11Ø	ØØØØ	da CY = 1
1ØØ1	1ØØØ	CY =1, AC = Ø

Das Ergebnis ist 98, da Carry =1 wurde Überlauf festgestellt. Damit ist das korrekte Ergebnis 1 98

Logische Befehle



Die logischen Befehle lassen sich sehr vielseitig verwenden:
Logisch UND zum Ausblenden von Bits
Logisch ODER zum Einblenden von Bits
Logisch EXCLUSIV ODER zum Schalten von Software-Weichen
Mittels der Vergleichbefehle können Register- bzw. Arbeitsspeicherinhalte mit dem Inhalt des Akkus verglichen werden. Die aus den logischen Befehlen resultierenden Anzeigen können durch bedingte Sprungbefehle zu Programmverzweigungen verwendet werden.

Hinweis: Der Vergleich Akku = Ø sollte mit den Befehlen logisch UND oder logisch ODER durchgeführt werden. (Laufzeit und Arbeitsspeicherplatz)

Wertetabellen für die logischen Befehle

1.) Logisch UND

Operand1	Operand2	Ergebnis
Ø	Ø	Ø
Ø	1	Ø
1	Ø	Ø
1	1	1

2.) Logisch ODER

Operand1	Operand2	Ergebnis
Ø	Ø	Ø
Ø	1	1
1	Ø	1
1	1	1

3.) EXCLUSIV ODER

Operand1	Operand2	Ergebnis
Ø	Ø	Ø
Ø	1	1
1	1	1
1	1	Ø

Anwendung der logischen Befehle

1.) Umsetzung einer Ziffer, die im ASCII-Code vorliegt in eine Sedezimalzahl

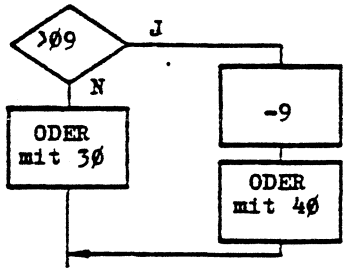
Tastaturtaste 7
ASCII-Code $37_{(16)}$
Ergebnis soll sein $07_{(16)}$

Binär:	ØØ11	Ø111
logisch UND mit	ØØØØ	1111
Ergebnis	ØØØØ	Ø111 = $07_{(16)}$

2.) Umsetzung eines Sedezimal-Halbbytes in ein abdruckbares Zeichen

Aus dem Hex-Werten ØØ-Ø9 soll werden 3Ø-39
Aus den Hex-Werten ØA-ØF soll werden 41-46

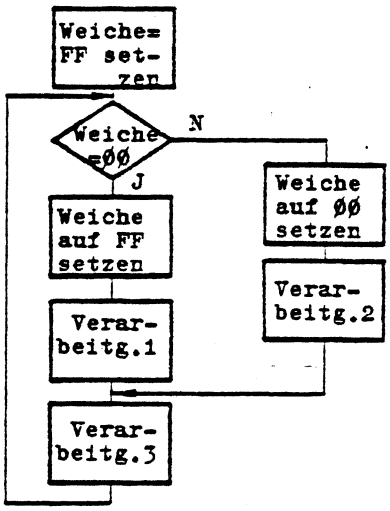
Ablauf:



Beispiel: $ØB_{(16)}$ soll werden $42_{(16)}$
≙ Buchstabe B im ASCII-Code)

	ØØØØ	1Ø11
	- ØØØØ	1ØØ1
	ØØØØ	ØØ1Ø
logisch ODER	Ø1ØØ	ØØØØ
mit	Ø1ØØ	ØØ1Ø = $42_{(16)}$

3.) Verwendung der logischen Befehle als Weichensteuerung



Lösung mit EXCLUSIV ODER

Ausgangspunkt	1111	1111
EXCLUSIV ODER	1111	1111
	ØØØØ	ØØØØ Weiche für Verarbeitung 1
EXCLUSIV ODER	1111	1111
	1111	1111 Weiche für Verarbeitung 2
EXCLUSIV ODER	1111	1111
	ØØØØ	ØØØØ Weiche für Verarbeitung 1

usw.

Bei allen Vergleichsbefehlen ist der 1. Vergleichsoperand immer der Akkumulator.
Ein Vergleich wird zurückgeführt auf die binäre Subtraktion, dabei wird jedoch das Ergebnis nicht in das Zielregister übertragen. Es wird nur das Flag-Register verändert (wie bei der Subtraktion).
Bei der Programmierung können durch sich bei den Vergleichen mitunter Probleme dadurch einstellen, daß sich bestimmte Auswertungen nur sehr schwer erreichen lassen. Es muß deshalb nach Wegen gesucht werden, um dies zu umgehen.

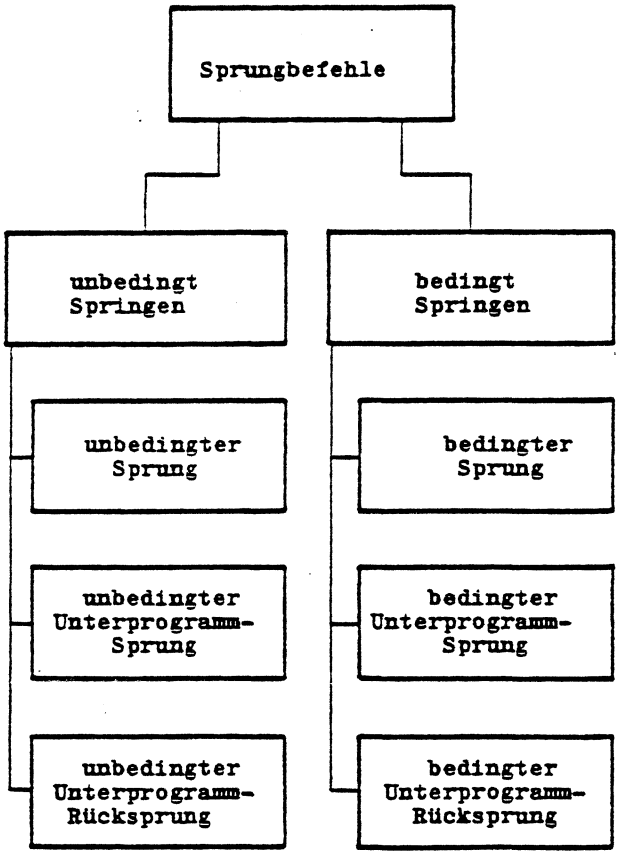
Beispiel: Der Akkumulatorinhalt soll auf größer 1Ø untersucht werden. Bei größer 1Ø soll nach Ziel gesprungen werden.
Folgender Befehl führt zum Ziel

```
CPI    11
JNC    ZIEL
```

Begründung: Das Carry-Bit wird gesetzt, wenn der 2. Operand größer ist als der Akkuinhalt, es wird rückgesetzt, wenn der 2. Operand gleich oder kleiner ist.

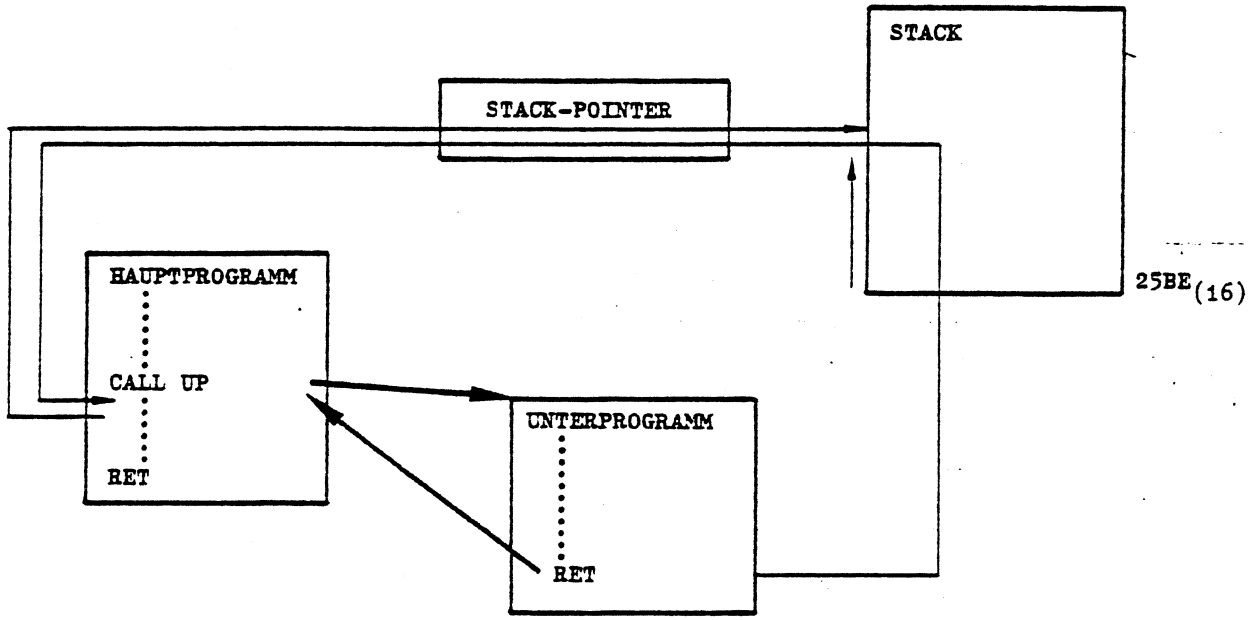
CY = 1:	2. Operand > Akkuinhalt oder Akkuinhalt < 2. Operand
CY = Ø:	2. Operand ≤ Akkuinhalt oder Akkuinhalt ≥ 2. Operand

Sprungbefehle



Bei den bedingtern Sprung-, bzw. Unterprogrammsprung- und Rücksprungbefehlen kann jedes Flag-Bit zur Sprungauslösung benutzt werden.

Bei den Unterprogrammsprungbefehlen wird die Rückkehradresse im Stack hinterlegt, bei den Rücksprüngen wird die Zieladresse dem Stack entnommen. Es muß daher sichergestellt sein, daß der Stackpointer auf dir richtige Stelle zeigt.

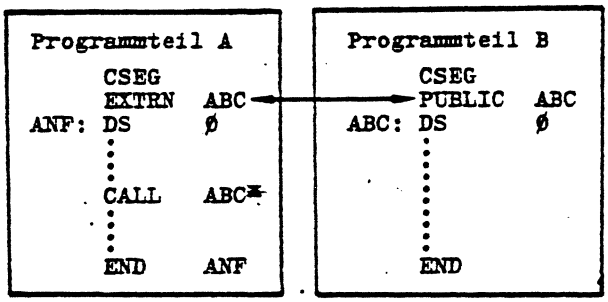


Anmerkung: Das Betriebssystem BS61Ø startet ein Anwenderprogramm durch einen CALL-Aufruf an die Startadresse. Das Programm wird daher ordnungsgemäß durch einen RET-Befehl beendet.

Komplexe Programme werden, so zeigt die Entwicklung, meist in kleinere, übersehbare Teile zerlegt. Damit ist es möglich, mehrere Programmierer eines Teams gleichzeitig mit der Lösung einer Aufgabe zu beschäftigen. Das Gesamtprogramm wird auf diese Art wesentlich früher zum Einsatz kommen.

Die einzelnen Programmteile werden getrennt übersetzt und zum Teil bereits unabhängig von den übrigen Teilen getestet. Alle übersetzten Teile (Module) werden dann mittels eines Bindeprogramms zu einem Programm zusammengebunden

An vielen Stellen muß jedoch innerhalb eines Programmteiles auf symbolische Adressen bezug genommen werden, die in einem anderen Teil definiert worden sind. Diese Verbindung ermöglicht die EXTERN-ADRESSIERUNG.



* Die sedezimale Zuordnung für die symbolische Adresse ABC wird später vom Binder durchgeführt.

Assembleranweisung für die Modulbehandlung

EXTRN symbolische Adresse;Bemerkungen

Die Anweisung EXTRN teilt dem Assembler mit, daß eine symbolische Adresse nicht in diesem Programmteil definiert ist. Bei der Übersetzung wird zunächst der Wert ØØØØ als sedezimale Adresse eingetragen. Die richtige Adresse wird später durch den Binder eingesetzt.

PUBLIC symbolische Adresse;Bemerkungen

Die Anweisung PUBLIC teilt dem Assembler mit, daß eine in diesem Programmteil definierte Adresse in anderen Programmteilen benutzt wird. Der Assembler gibt diese Information an den Binder weiter. Dieser kann nur die Extern-Adressen befriedigen, die mit PUBLIC als benutzbar erklärt werden.

CSEG

Mit CSEG werden Programmteile gekennzeichnet. Damit sind in der Hex-file Relationierungsinformationen für den Binder enthalten. Mit dem Binderkommando =C=XXXX kann die Ladeadresse eines Programmmodules festgelegt werden.

DSEG

Mit DSEG werden Datenteile gekennzeichnet. Hierbei werden in der Hex-file keine Relationierungsinformationen erzeugt. Mit dem Binderkommando =D=XXXX kann die Ladeadresse eines Datenmodules festgelegt werden.

Für die in Assembler zu erstellenden Programme beinhaltet das BS 670 eine Reihe Benutzerschnittstellen für die Ein-/Ausgabeprogrammierung.

Bedient werden derzeit:

1) Bildschirm und Tastatur
2) Floppy Disk
3) Magnet-Cartridge
4) CPU-V24 Interface
5) CPU-Current-Loop Interface (Drucker)

Die Bedienung der Peripherie ist sowohl auf "physikalischer-" als auch auf "logischer" Ebene möglich.

Physikalische Ebene:

Die physikalische Ebene ist im MONITOR realisiert. Sie erfordert vom Benutzer detailierte Kenntnisse der zur bedienenden Peripherie. Das E/A-Gerät muß zur Programmerstellungszeit fixiert werden.

Logische Ebene:

Der Benutzer kann die Wahl des E/A-Gerätes zur Laufzeit des Programmes festlegen. Er benötigt keine Detailkenntnisse über die Hardware.

PHYSIKALISCHE EIN/AUSGABE

Vom Anwender aufrufbare Routinen

Der Monitorkern enthält eine Anzahl von Routinen, die vom Anwender mit CALL aufgerufen werden können. (Ausnahme: MON, der mit JMP angesprungen werden soll). Danach wird in das Anwenderprogramm zurückverzweigt. (Ausnahmen: MON, LINK und CLINK).

ACHTUNG: Routinen, die die Kassetten ansprechen, sind im Monitor MON/C. Auf Anlagen, die nicht mit diesem Monitorzusatz ausgerüstet sind, sind diese Funktionen nicht verfügbar.

ADRESSE	ROUTINE	FUNKTION
40	TTI	Einlesen eines Zeichens von der Tastatur
43	TTO	Ausgeben eines Zeichens auf Bildschirm
46	TTONC	Ausgeben eines Zeichens auf Bildschirm (Kontrollzeichen werden wie Datenzeichen behandelt)
49	CURSOR	Positionierung des Cursors
4C	MON	Anspringen des Monitors (kein Rücksprung)
4F	ERROR	Ausgabe einer Fehlermeldung a.d. Bildschirm
52	PRINT	Ausdrucken eines Zeichens
55	DTST	Test des Ausführungszustands einer Diskettenoperation
58	DCAL	Anstoß zur Positionierung auf Spur 0 der Diskette.
5B	DSRD	Anstoß zum Lesen eines Diskettensektors
5D	DSWR	Anstoß zum Schreiben eines Diskettensekt.
61	DSDL	Anstoß zum Löschen eines Diskettensektors
64	DSRDA)*	Anstoß zum Lesen eines Diskettensektors und Konvertierung in den ASCII-Code
67	DSWRA)*	Konvertierung in den EBCDIC-Code und Anstoß zum Schreiben eines Disketten-Sektors
6A	INDISP	Lesen eines Zeichens vom Bildschirm
6D	DSWT	Warten auf den Abschluß einer Disketten-Operation
70	DCALW	wie DCAL, jedoch mit Warten auf den Abschluß der Operation (Funktion der Routine DSWT)
73	DSRDW	wie DSRD, aber mit DSWT
76	DSWRW	wie DSRW, " " "
79	DSDLW	wie DSDL, " " "
7C	DSRAW)*	wie DSRDA, " " "
7F	DSWAW)*	wie DSWRA, " " "

*) Nur benutzbar, wenn der ladbare Teil des BS 610 im RAM-Speicher steht.

ADRESSE	ROUTINE	FUNKTION
82	RCVI	Empfang eines Zeichens vom Hauptrechner (UART)
85	LOCDIR	Eintrag eines Namens im Inhaltsverzeichnis einer Diskette (INTEL-Format)
88	LOAD	Laden eines Programmes von einer Diskette (INTEL-Format)
8B	LINK	Laden und Starten eines Programmes (Achtung: Normalerweise ohne Rücksprung!)
8E	CTST	Test des Ausführungszustandes einer MB-Kassettenoperation
91	CCAL	Anstoss zum Rücksetzen auf Abschnittsmarke (AM)
94	CBRD	Anstoss zum Lesen eines Kassetten-blockes
97	CBWR	Anstoss zum Schreiben eines Kassetten-blockes
9A	CBBS	Anstoss zum Rücksetzen um einen Block
9D	CBTM	Anstoss zum Schreiben einer AM
AO	CBSF	Anstoss zum Vorwärtssuchen
A3	CBSB	Anstoss zum Rückwärtssuchen
A6	CBUN	Anstoss zum Rückspulen und Entladen
A9	CRW	Anstoss zum Rückspulen
AC	CWT	Warten auf Abschluß einer Kassetten-operation
AF	CBRDW	Lesen eines Blockes)*
B2	CBWRW	Schreiben eines Blockes)*
B5	CBEDW	Rückschreiben eines Blockes)*
B5	CBBSW	Rücksetzen um einen Block)*
B8	CBTMW	Schreiben einer AM)*
BB	CBSFW	Vorwärtssuchen)*
BE	CBSBW	Rückwärtssuchen)*

)* Mit Warten

ADRESSE	ROUTINE	FUNKTION
C1	CLOCD	Schreiben eines Inhaltsverzeichnisblokkes
C4	CLOAD	Laden eines Programmes
C7	CLINK	Laden und Starten eines Progr.
CA	XMIO	Senden zum Hauptrechner (UART)

Alle mit "C" beginnenden Routinen bedienen die MB-Kassette.

Zur Beachtung: Der Ansprung von Fehlerroutinen aus den MB-Kassettenroutinen und der Rücksprung dorthin ist nur möglich, wenn eine Diskette im BS 610-Format vorhanden ist, da die Texte der Fehlermeldungen dort gelesen werden.

TTI, TTO, TTONC, CURSOR, MON, ERROR, PRINT, INDISP, RCVI und XMIO sind in allen Systemen vorhanden.

Nach dem Rücksprung ins Anwenderprogramm ist das Carrybit gesetzt, wenn die aufgerufene Routine nicht fehlerfrei abgelaufen ist:

Carrybit nicht gesetzt: normaler Ablauf
Carrybit gesetzt : fehlerhafter Ablauf, der Fehlercode steht im A-Register

Alle übrigen Register sind unverändert (SAVE beim Ansprung, RESTORE beim Rücksprung)

```
SIEMENS 610 ASSEMBLER V2.2        *****  HARDCOPY ** BILDSCHIRM AUF DRUCKER  *****02.07.79*CNDL*

                          1 $TITLE      *****  HARDCOPY ** BILDSCHIRM AUF DRUCKER  *****02.07.79*CNDL*
                          2 ;                              MIT PHYSIKALISCHER E/A
                          3 $PW110
                          4 $PL48
                          5           CSEG
006A                      6 INDISP    EQU     6AH
0049                      7 CURSOR    EQU     49H
0052                      8 PRINT     EQU     52H
0043                      9 TTO       EQU     43H
000D                     10 CR        EQU     13
000A                     11 LF        EQU     10
000C                     12 FF        EQU     12
001D                     13 HOME      EQU     29
0000 C  2A2A2A2A         14 ENDZ:     DB      '************************************************'
0032 C  2A2A2A2A         15           DB      '********************************',00
0051 C  3E1D             16 ANF:      MVI     A,HOME
0053 C  CD4300           17           CALL    TTO         ;CURSOR AN BILDANFANG
0056 C  0618             18           MVI     B,24        ;ZEILENZAEHLER LADEN
0058 C  0E50             19 LOOP1:    MVI     C,80        ;ZEICHENZAEHLER LADEN
005A C  CD6A00           20 LOOP2:    CALL    INDISP      ;EIN ZEICHEN VOM BILDSCHIRM LESEN
005D C  FE20             21           CPI     20H         ;ZEICHEN < BLANC ?
005F C  D26400    C      22           JNC     AUSG        ;NEIN ----> DRUCKEN
0062 C  3E20             23           MVI     A,20H       ;JA  ZCHN ==> SPACE
0064 C  CD5200           24 AUSG:     CALL    PRINT       ;ZEICHEN DRUCKEN
0067 C  0D               25           DCR     C           ;ZEICHENZAEHLER - 1
0068 C  C25A00    C      26           JNZ     LOOP2       ;---> LOOP2 SOLANGE ZEICHENZAEHLER NICHT 0
006B C  3E0D             27           MVI     A,CR
006D C  CD5200           28           CALL    PRINT       ;WAGENRUECKLAUF AUSGEBEN
0070 C  3E0A             29           MVI     A,LF
0072 C  CD5200           30           CALL    PRINT       ;ZEILENVORSCHUB AUSGEBEN
0075 C  05               31           DCR     B           ;ZEILENZAEHLER - 1
0076 C  C25800    C      32           JNZ     LOOP1       ;---> LOOP1 SOLANGE ZEILENZAEHLER NICHT 0
0079 C  3E0C             33           MVI     A,FF
007B C  CD5200           34           CALL    PRINT       ;FORMULARVORSCHUB AUSGEBEN
007E C  210000    C      35           LXI     H,ENDZ      ;H/L = TEXTADRESSE
0081 C  7E               36 LOOP3:    MOV     A,M         ;ZEICHEN ---> AKKU
0082 C  B7               37           ORA     A           ;ZEICHEN = 0 ?
0083 C  CA8D00    C      38           JZ      ZEIV        ;JA ---> ZEIV
```

```
SIEMENS 610 ASSEMBLER V2.2          *****  HARDCOPY ** BILDSCHIRM AUF DRUCKER  *****02.07.79*CNDL*

0086 C  CD4300          39          CALL    TTO         ;ZEICHEN AUSGEBEN
0089 C  23              40          INX     H           ;AUSGABE - ADR. + 1
008A C  C38100    C     41          JMP     LOOP3       ;---> NAECHSTES ZEICHEN AUSGEBEN
008D C  3E0A            42 ZEIV:    MVI     A,LF
008F C  CD4300          43          CALL    TTO         ;ZEILENVORSCHUB AUSGEBEN
0092 C  C9              44          RET                 ;RUECKSPRUNG ---> BS 610
                        45          END     ANF
NO ERROR MESSAGES
```

```
                          1 $TITLE  ZEICHENVORRAT AUF DRUCKER AUSGEBEN ******** 04.07.79**CNDL**
                          2 $PW 110
                          3 $PL 48
                          4 ;
                          5 ;*************************************************************
                          6 ;                                                            *
                          7 ; AUSGABE DES GESAMTEN ZEICHENVORRATES (CODE 20 - 7F)        *
                          8 ; AUF DRUCKER 6324 , ABWECHSELND NORMAL/KURSIV               *
                          9 ; ROUTINE 10 MAL WIEDERHOLEN - ZEILENTRENNUNG (CR +LF)       *
                         10 ;                                                            *
                         11 ;*************************************************************
                         12 ;
                         13 ; EQU TABELLE:
                         14 ;
0052                     15  PRINT  EQU     52H     ;Adresse PRINT
                         16 ;
                         17         CSEG
0000 C  0600             18 ANF:    MVI     B,0     ;B - Reg. loeschen, B=Zeilenzaehler
0002 C  0EFF             19 SCHK:   MVI     C,0FFH  ;Schalter setzen
0004 C  3E7F             20         MVI     A,7FH   ;Ausgabe von DEL an den Drucker
0006 C  CD5200           21         CALL    PRINT   ;
0009 C  3E1B             22         MVI     A,1BH   ;Ausgabe von ESC an den Drucker
000B C  CD5200           23         CALL    PRINT   ;
000E C  3E33             24         MVI     A,33H   ;Kursiv
0010 C  CD5200           25         CALL    PRINT   ;
0013 C  3E1F             26 SCHN:   MVI     A,1FH   ;AKKU laden mit (Anfangscode minus 1)
0015 C  3C               27 SCH1:   INR     A       ;AKKU plus 1
0016 C  FE80             28         CPI     80H     ;AKKU = 80H ?
0018 C  CA2100    C      29         JZ      SCH2    ;wenn ja----> springe zu SCH 2
001B C  CD5200           30         CALL    PRINT   ;AKKU = kleiner 80H, dh. ausdrucken AKKU - Inh
001E C  C31500    C      31         JMP     SCH1    ;unbed. Sprung nach SCH1,- (naechstes Zeichen)
0021 C  3E0D             32 SCH2:   MVI     A,0DH   ;Wagenruecklauf (CR) nach AKKU
0023 C  CD5200           33         CALL    PRINT   ;ausgeben CR
0026 C  3E0A             34         MVI     A,0AH   ;Zeilenvorschub (LF) nach AKKU
0028 C  CD5200           35         CALL    PRINT   ;ausgeben LF
002B C  04               36         INR     B       ;Zeilenzaehler erhoehen
002C C  78               37         MOV     A,B     ;Zeilenzaehlerstand nach AKKU
002D C  FE0A             38         CPI     10      ;Zeilenzaehler = 10 ?
```

```
SIEMENS 610 ASSEMBLER V2.2        ZEICHENVORRAT AUF DRUCKER AUSGEBEN ******** 04.07.79**CNDL**

002F C  C23300   C     39          JNZ     SCHAB   ;Zeilenzaehler kleiner = 10
0032 C  C9               40          RET             ;Ruecksprung ---> BS 610
0033 C  79               41 SCHAB:   MOV     A,C     ;Schalter nach Akku
0034 C  FEFF             42          CPI     0FFH    ;Gesetzt ?
0036 C  C20200   C     43          JNZ     SCHK    ;Schalter ist rueckgesetzt , zurueck zu KURSIV
0039 C  0E00             44          MVI     C,00H   ;Schalter ruecksetzen
003B C  3E7F             45          MVI     A,7FH   ;DEL
003D C  CD5200           46          CALL    PRINT   ;zum Drucker
0040 C  3E1B             47          MVI     A,1BH   ;ESC
0042 C  CD5200           48          CALL    PRINT   ;zum Drucker
0045 C  3E34             49          MVI     A,34H   ;NORMAL
0047 C  CD5200           50          CALL    PRINT   ;zum Drucker
004A C  C31300   C     51          JMP     SCHN    ;zurueck zu NORMAL - Ausgabe
                         52          END     ANF

NO ERROR MESSAGES
```

```
                        1 $TITLE  DRUCKER SCHRAEGLAUF-TEST      ***** 04.07.79 **CNDL**
                        2 $PW 110
                        3 $PL 48
                        4 ;
                        5 ;*************************************************************
                        6 ;                                                            *
                        7 ; AUSGABE DES GESAMTEN ZEICHENVORRATES ( CODE 20 BIS 7F )    *
                        8 ; AUF DRUCKER. ZEILENBEGINN IN AUFSTEIGENDEM ASCII - CODE    *
                        9 ; PROGRAMMSTOP DURCH BELIEBIGE TASTATUREINGABE.              *
                       10 ; ABSCHLIESSEND FORMULARANFANG                               *
                       11 ;                                                            *
                       12 ;*************************************************************
                       13           CSEG;                  BEGINN DES CODESEGMENTES
                       14 ;
0052                   15 PRINT     EQU    52H;            MONITOR-PRINTER-DRIVER-ADRESSE
2780                   16 TAPUFF    EQU    2780H;          SYSTEM - TASTATURPUFFER
000D                   17 CR        EQU    0DH;            WAGENRUECKLAUF
000A                   18 LF        EQU    0AH;            ZEILENVORSCHUB
000C                   19 FF        EQU    0CH;            FORMULARANFANG
                       20 ;
0000 C                 21 START:    DS     0
0000 C  0600           22           MVI    B,0;            ZEILENZAEHLER = 0
0002 C  0E60           23 LOOP0:    MVI    C,96;           ZEICHENZAEHLER = 96
0004 C  3E1F           24           MVI    A,1FH;          AKKU = ANFANGSCODE MINUS 1
0006 C  80             25           ADD    B;              ZEILENZ. + AKKU = STARTCODE
0007 C  3C             26 LOOP1:    INR    A;              CODE + 1
0008 C  FE80           27           CPI    80H;            HOECHSTER CODE ?
000A C  CA3600    C    28           JZ     LOOP3;          JA ---> LOOP3
000D C  CD5200         29 LOOP2:    CALL   PRINT;          NEIN - DRUCKEN EIN ZEICHEN
0010 C  0D             30           DCR    C;              ZEICHENZAEHLER MINUS 1
0011 C  C20700    C    31           JNZ    LOOP1;          SOLANGE NICHT = 0 ---> LOOP1
0014 C  3E0D           32           MVI    A,CR;           CR
0016 C  CD5200         33           CALL   PRINT;          AUSGEBEN
0019 C  3E0A           34           MVI    A,LF;           LF
001B C  CD5200         35           CALL   PRINT;          AUSGEBEN
001E C  04             36           INR    B;              ZEILENZ. + 1
001F C  78             37           MOV    A,B;            ZEILENZAEHLER ---> AKKU
0020 C  FE5F           38           CPI    95;             96 ZEILEN AUSGEGEBEN
```

```
SIEMENS 610 ASSEMBLER V2.2        DRUCKER SCHRAEGLAUF-TEST     ***** 04.07.79 **CNDL**

0022 C  CA0000    C     39           JZ    START;          JA ---> NEUER ZYKLUS
0025 C  3A8027          40           LDA   TAPUFF;         TASTATUR - ZEICHENZ. ---> AKKU
0028 C  B7              41           ORA   A;              EIN ZEICHEN ALS STOP-KRITERIUM EIGEG.?
0029 C  CA0200    C     42           JZ    LOOP0;          NEIN - - FORTSETZEN
002C C  3E0C            43           MVI   A,FF;           FORMULARANFANG
002E C  CD5200          44           CALL  PRINT;          AUSGEBEN
0031 C  97              45           SUB   A;              AKKU LOESCHEN
0032 C  328027          46           STA   TAPUFF;         TASTATURZEICHEN ELIMINIEREN
0035 C  C9              47           RET   ;               ---> BS610
0036 C  3E20            48 LOOP3:    MVI   A,20H;          BLANC ---> AKKU
0038 C  C30D00    C     49           JMP   LOOP2;
                        50           END   START;

NO ERROR MESSAGES
```

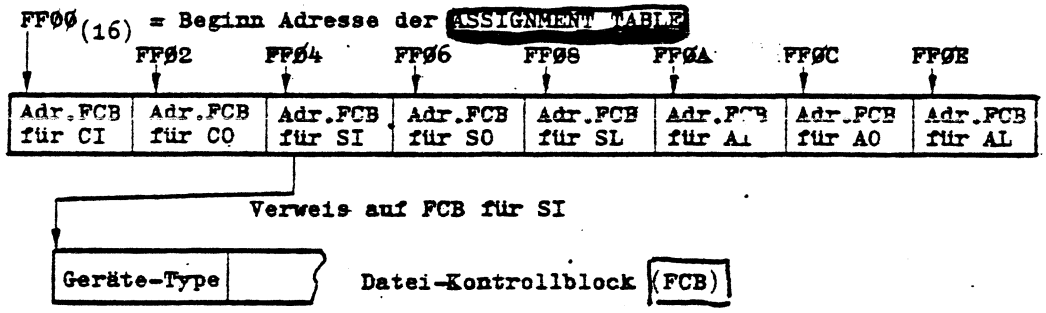
LOGISCHE EIN/AUSGABE

Die Verwendung der logischen Schnittstelle für die Ein-/Ausgabe erfordert zur Programmerstellungszeit keinerlei Wissen um die zu verwendeten Geräte und Dateinamen. Diese werden erst bei Ablauf des Programmes zugewiesen.

Wird bei Laden des Programmes die Zuordnung einer Datei vom "ASSIGNMENT INTERPRETER" des BS 610 durchgeführt, so darf kein OPEN und CLOSE im Programm für diese Datei gegeben werden, da das BS 610 vor Start des Programms die Datei eröffnet, und bei Rückkehr über RET die Datei schließt.

Über die logische Schnittstelle können die logischen Geräte
CO, CI, SI, SO, SL, AI, AO, AL
verwendet werden.

In der "ASSIGNMENT TABLE" werden für diese 8 Geräte die Adressen der Datei-Kontroll-Blöcke (FCB=File-Control-Block) hinterlegt.

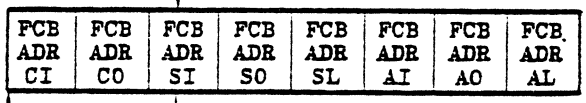


Die Adresse des FCB muß als Eingangsbedingung für einen logischen E/A-Aufruf im H/L-Register vorhanden sein! Weitere Eingangsbedingungen je nach verwendeter Routine erforderlich.

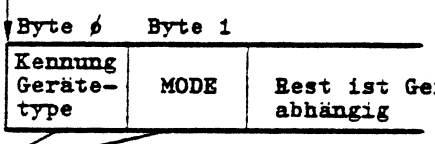
Eintrag der FCB-Adresse erfolgt bei

1) ASSIGN SI=:mn:

2) PROG SI=:mn:DATEI.NAM



FFØØ = ASSIGNMENT-TABELLE



1 = Zeichenorientiertes Gerät im "Interrupt-Betrieb"

2 = Zeichenorientiertes Gerät im "Abfrage-Betrieb"

3 = Echo-Gerät

4 = Zeilenorientiertes Gerät

5 = Blockorientiertes Gerät

6 = Doppel-Gerät

2^0 = 1 : eröffnet für Eingabe

2^1 = 1 : eröffnet für Ausgabe

2^2 = 1 : Line-Feed wird unterdrückt

2^3 = 1 : dz. nicht verwendet

2^4 = 1 : bei nächstem Lesen wird EOF erkannt

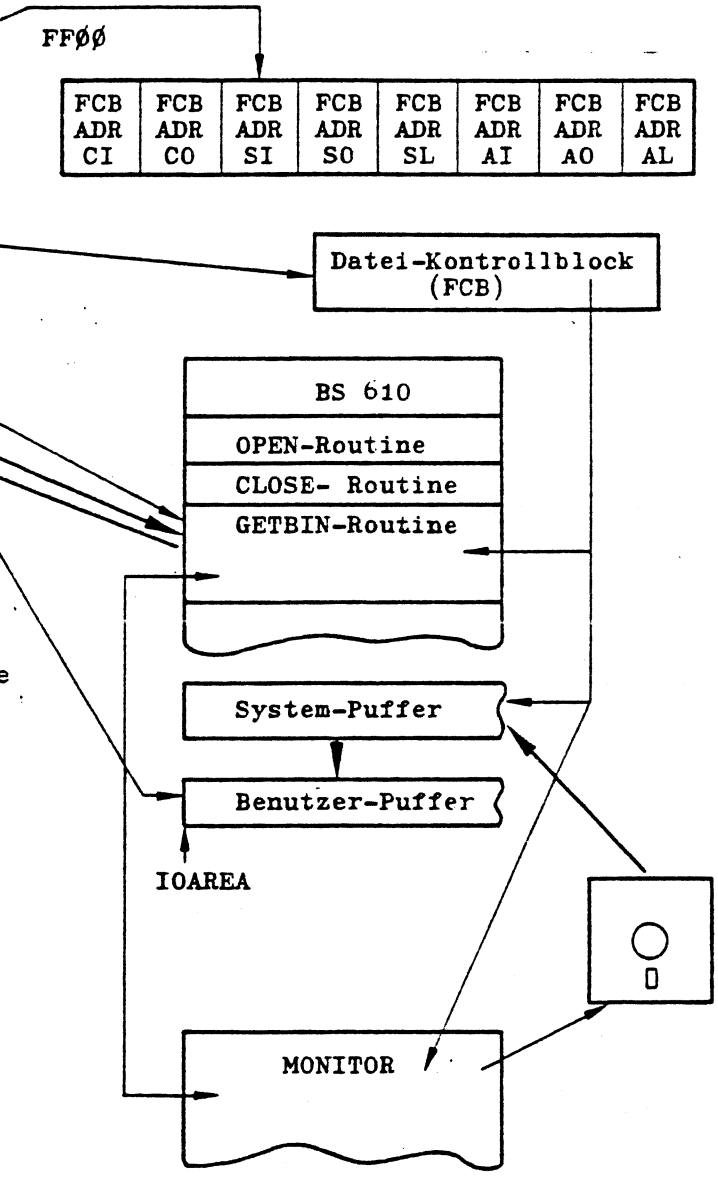
2^5 = 1 : kein Austrag aus Assignment-Tabelle
(Open wurde durch Kommando-interpreter durchgeführt)

2^6 = 1 : es soll kein CLOSE durchgeführt werden
(Zuordnung erfolgte durch ASSIGN)

2^7 = 1 : EOF erkannt

Zusammenhang: Anwenderprogramm - BS610-Monitor

```
FCBSI    EQU   OFF04H
LSSCHR:  ORG   4000H

LESEN:   LHLD  FCBSI
         LXI   D,EABER
         MVI   B,80

GETBIN   EQU   OFD18H

         CALL  GETBIN
         JNC   VERARB
         CPI   0
         JNZ   FEHLER

;*****     EOF bei Lesen
;*****     Reg B = Restlänge

VERARB:  LXI   D,EABER

;*****     Reg H/L = FCB-Adresse

         JMP   LESEN

FEHL:    DS    0

;*****     Reg A = Fehler


EABER:   DS    80
```



Benutzung der BS 610-Funktionen

Eine BS 610-Funktion wird aufgerufen durch ein CALL-Statement mit der zugehörigen Sprungtabellen-Adresse. Vor dem CALL müssen die zugehörigen Parameter in die entsprechenden Register gebracht werden. Nach dem Rücksprung zeigt das Carrybit an, ob die Routine normal (Carry = 0) oder fehlerhaft (Carry gesetzt) abgelaufen ist. Im letzteren Fall enthält das A-Register den Fehlercode (vergl. Anhang A). Carry gesetzt und A = 0 bedeuten bei Eingabe Dateiende, bei Ausgabe Überlauf.

Routinen, die E/A-Operationen enthalten, verlangen in HL als Parameter die Adresse eines Datei-Kontrollblockes (FCB). Falls die normalen Dateizuweisungen verwendet werden, braucht sich der Anwender mit den Formaten der FCBs nicht abzugeben. Für spezielle Zwecke befindet sich eine grobe Darstellung der Formate im Anhang B. Üblicherweise verwendet man einen LHD-Befehl, um den FCB einer speziellen Einheit aus der Gerätezuweisungstabelle zu holen (FF00 - FF0F).

Im folgenden wird eine kurze Beschreibung der verfügbaren Systemfunktionen gegeben. Wenn nicht anders angegeben, werden alle Register gerettet und wiederhergestellt mit Ausnahme derer, die anschließend ein Ergebnis enthalten.

FD00 BS 610 Rückspruch in den Kommandomodus des BS 610.
Ansprung mit JMP oder CALL, beim Rücksprung zum Anwender. Keine Parameter.

Zur Beachtung:

Normalerweise sollte der Rücksprung ins BS 610 mit RET erfolgen, weil dadurch alle durch eine Zuweisung eröffneten Dateien wieder geschlossen werden. Der Rücksprung über FD00 läßt diese Dateien offen.
Nch einem Fücksprung über RET zeigt das Carrybit den Verlauf der Operation an. Bei Auftreten eines Fehlers (Carry gesetzt) steht der Fehlercode in A.

FD03 ERROR Gibt eine Fehlermeldung ERROR XX auf CO und springt ins Anwenderprogramm zurück, falls das Carrybit nicht bereits beim Ansprung gesetzt war.

FD06 LOAD lädt ein Programm von einer INTEL-formatierten Diskette. Parameter: s. LOAD- Funktion des Monitors.

Die Sprungtabelleneinträge von FD09 - FD27 enthalten grundlegende E/A- Funktionen.

Es wird vorausgesetzt, daß die logischen Einheiten (CI, CO, SI, SO, SL, AI, AO, AL) durch das BS 610 zugewiesen und damit auch gleichzeitig eröffnet wurden.

Folgende Funktionen sind verfügbar:

- Lesen des nächsten Zeichens (INCHAR)
- Schreiben eines Zeichens (OUTCHAR)
- Einlesen eines Textes bis zum nächsten CR LF in einen vom Anwender zugewiesenen Puffer (GET)
- Schreiben eines Textes bis zum nächsten CR aus einem vom Anwender angegebenen Puffer (PUT, PUTSTR)
- Einlesen einer bestimmten Anzahl von Zeichen in einen
- Anwenderpuffer (GETBIN)
- Schreiben einer bestimmten Anzahl von Zeichen aus einem Anwenderpuffer (PUTBIN)
- Lesen eines Blockes (READ)
- Schreiben eines Blockes (WRITE)
- Schließen einer Datei (CLOSE)
- Eröffnen einer Datei (OPEN), falls dies nicht vom System besorgt wurde.

Alle vorstehenden Routinen benötigen die Adresse eines FCB in HL. Für normale Dateien und bei vorheriger Definition entsprechender EQUs kann dies mit LHLD erledigt werden.

- Darstellung eines Bytes oder Wertes in hexadezimaler Form im ASCII-Code (EDBH bzw. EDWH). Beispiel für EDBH: Byte enthält ASCII-A = 41 H. Nach EDBH enthält das Empfangsfeld 3431H = 41 (ASCII)
- Umwandlung einer Hexadezimal-Darstellung in ein Byte bzw. Wort (EDHB bzw. EDHW, umgekehrte Funktion von EDBH/EDWH)
- Dynamische Pufferzuweisung bzw. Löschung (BUFALLOC/ BUFDEALLOC).

- Setzen eines Zeit-Intervalles (STIMER).

Routine	Einsprung-Parameter	Resultate
FD09 INCHAR	HL - Dateikontrollblock (FCB)	Zeichen in A. Falls Carry gesetzt, Fehler oder EOF.
FD0C OUTCHAR	HL - FCB	Carry gesetzt: Fehler oder EOE.

Routine	Einsprung-Parameter	Resultate
EDOF GET	HL - FCB	Carry gesetzt: Fehler oder EOT.
	DE - Pufferadresse	DE verändert.
FD12 PUT	HL - FCB	Carry gesetzt: Fehler oder EOE.
	DE - Pufferadresse	DE verändert.
FD15 PUTSTR	wie PUT	
FD18 GETBIN	HL - FCB	Carry gesetzt: Fehler oder EOF.
	DE - Pufferadresse	DE verändert
	B - Zeichenzähler	
FD1B PUTBIN	HL - FCB	Carry gesetzt: Fehler oder EOE.
	DE - Pufferadresse	DE verändert
	B - Zeichenzähler	
FD1E OPEN	HL - FCB	
	A - OPEN Code	Carry gesetzt: Fehler.
	(1 = Eingabe, 2 = Ausg.)	
FD21 READ	HL - FCB	Carry gesetzt: Fehler oder EOE.
FD24 WRITE	HL - FCB	Carry gesetzt: Fehler oder EOE.
FD27 CLOSE	HL - FCB	Carry gesetzt: Fehler.

Unterschied zwischen PUT und PUTSTR: PUT fügt CR LF ein, PUTSTR nicht.

EOE Ausgabedateiende. (Ende des vorgesehenen Dateibereichs erreicht.)
EOF Ende der Eingabedatei
EOT Textende

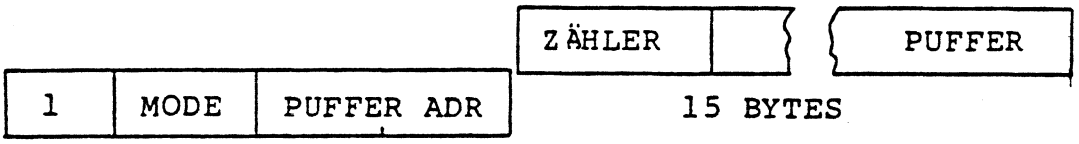
Routine	Einsprung-Parameter	Resultate
FD2A EDBH	A - umzuwandelndes Byte	Hexdarstellung im Empfangsfeld.
	HL - Adresse eines 2-Byte-Empfangsfeldes	
FD2D EDWH	DE - umzuwandelndes Wort	Hex-Darstellung im Empfangsfeld
	HL - Adresse eines 4-Byte-Empfangsfeldes	

Routine	Einsprungparameter	Resultate
FD30 EDHB	HL - Adresse eines 2-Byte-Bereichs	Byte in A
FD33 EDHW	HL - Adresse eines 4-Byte-Breichs	Wort in DE
FD36 BUFALLOC	HL - gewünschte Puffergröße	HL-Pufferadresse.
FD39 BUFDEALLOC	HL - Pufferadresse	
FD3C STIMER	DE - Gewünschtes Zeitintervall in 20ms-Einheiten HL - Adresse, die nach Ablauf des Zeitintervalles angesprungen wird.	

Anhang B

Formate der Datei-Kontrollblöcke (FCBs)

ZEICHENORIENTIERTES GERÄT (INTERRUPT-Betrieb)



MODE kann folgende Werte annehmen:

1 - eröffnet für Eingabe
2 - eröffnet für Ausgabe
4 - LF wird nicht unterdrückt

10H - LB-Bit (beim nächsten Lesen wird Datei- oder Bereichende erreicht)

20H - Auto (Datei wurde durch den ASSIGNMENT-Interpreter angelegt bzw. eröffnet. Beim Rücksprung in den Monitor wird sie geschlossen und der zugehörige Puffer freigegeben)

40H - kein CLOSE (Datei soll nicht geschlossen werden)

80H - EOF-Bit (Datei oder Bereichsende wurde erreicht)

ZEICHENORIENTIERTES GERÄT (Abfragebetrieb)

2	MODE	INCHAR	OUTCHAR

INCHAR und OUTCHAR sind Adressen von Subroutinen
INCHAR liest aus einer Eingabedatei ein Zeichen nach A.
Bei fehlerhaftem Ablauf ist das Carrybit gesetzt, sonst nicht.

OUTCHAR schreibt ein Zeichen aus A in eine Ausgabedatei.
Carryfunktion wie bei INCHAR.
Alle Register (Ausnahme: PSW bei INCHAR) müssen gerettet oder wiederhergestellt werden.
Normalerweise enthält nur eines der beiden Felder eine Adresse.

ECHO-Gerät

3	MODE	INPUT DEV.	OUTPUT DEV.

INPUT und OUTPUT DEV sind Adressen anderer Kontrollblöcke. Sobald ein Zeichen vom Eingabegerät gelesen wird, wird es auf dem Ausgabegerät geechot.

ZEILENORIENTIERTES GERÄT

4	MODE	BUFR PTR	BUFR EXT	BUFFER ADR

CHAR ODER BLK DEV

BUFR PTR	Zeichenzähler, der auf das nächste Zeichen im Puffer verweist (beginnt mit 0)
BUFR EXT	Pufferlänge
BUFR ADR	Pufferadresse
CHAR ODER BLK DEV	Zeiger zu einem anderen Kontrollblock, der zum Empfang weiterer Zeichen aufgerufen wird.

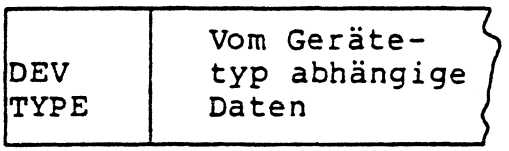
BLOCKORIENTIERTES GERÄT (Diskette und MB-Kassette)

Der gesamte File-Control-Block wird hier wegen seiner Länge für die Beschreibung in drei Teile zerlegt und dann Teil für Teil erklärt.

1.Teil (Byte 1 - 16) DISKETTE UND MB-KASSETTE

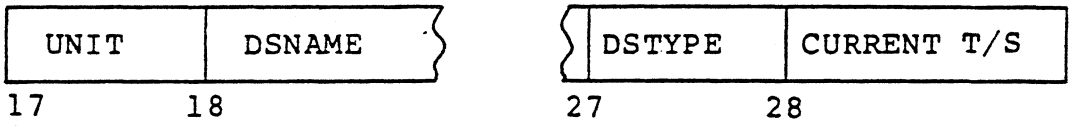
5	MODE	BUFFER PTR	BUFFER EXT	BUFFER ADR
0	1	2	4	6

READ	WRITE	OPEN	CLOSE
8	10	12	14



BUFFER PTR	Position des nächsten Zeichens innerhalb ds Puffers
BUFFER EXT	Pufferlänge
BUFFER ADR	Pufferadresse
READ WRITE OPEN	Geräteabhängige Adressen von Routinen zum Lesen, Schreiben, Eröffnen und Schließen. Normalerweise sind entweder READ oder WRITE = 0
CLOSE	Die OPEN- und CLOSE-Adressen sind vom Dateityp abhängig (Ein- oder Ausgabedatei).
DEV TYPE	Gerätetyp 0 - Diskette 1 - MB-Kassette

2.Teil (Byte 17 - 29) DISKETTE



UNIT	Physikalische Gerätenummer (0-3)
DSNAME	Dateiname. Bei INTEL-formatierten Disketten sind die ersten 6 Zeichen der Name, die nächsten 3 dessen Erweiterung. Bei IBM-formatierten Disketten sind die ersten 8 Zeichen der Name. Das 9. ist für zukünftige Erweiterungen reserviert (Zugriffskennung).
DSTYPE	80H EBCDIC 40H 512 Bytes *) 20H 256 Bytes *) 10H unbenutzt 8 binary seg *) 4 Setzen der I und S-Flags 2 IBM 1 INTEL

*) Zukünftige Erweiterung

CURRENT T/S	Spur und Sektor des aktuellen Dateikettungsblocks (bei INTEL-Format) bzw. des nächsten Satzes (bei IBM-Format).

3.Teil (Byte 30 - 33) DISKETTE

IBM-Format ≈ ECMA TC 19

EOD oder EOE	Fehlerspuren	HDR-Sektor
30	32	34

EOD oder EOE — Spur und Sektor des letzten geschriebenen Blockes bei Eingabedateien bzw. des letzten reservierten Blockes bei Ausgabedateien.

INTEL-Format:

CHAR LAST BLK	DIR CNT	DIR ADR
30	31	32

CHAR LAST BLK — Anzahl der Zeichen (Nutzdaten) im letzten Block (bei Eingabedateien)

DIR CNT — relativer Zeiger, der auf die Spur/Sektor-Adresse des nächsten Datenblockes zeigt. Diese Adressen werden im Dateikettungsblock-Puffer gespeichert.

DIR ADR — Adresse des Dateikettungsblock-Puffers

2.Teil (Bytes 17 - 29) MB-KASSETTE

UNIT	DSNAME		DSTYPE	OVERRUN ADR
17	18		27	28

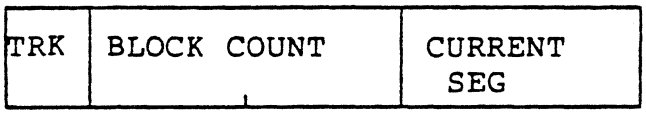
UNIT — Laufwerknummer

DSTYPE

80H	EBCDIC
40H, 20H, 10H, 8, 4	unbenutzt
2	ECMA-Format
1	BS 610-Format

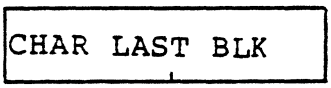
OVERRUN ADR wird zur Prüfung auf Leseüberläufe benutzt. Vor dem Lesen werden zwei Bytes mit besonderem Inhalt hinter den Lesepuffer plaziert. Nach dem Lesen wird geprüft, ob die Bytes unverändert sind.

Teil 3 MB-KASSETTE BS 610 - FORMAT



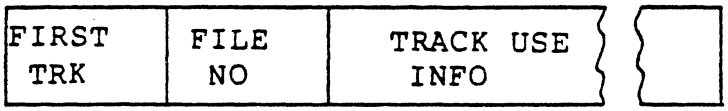
TRK	Spur von der/auf die momentan gelesen/geschrieben wird
BLOCK COUNT	wird für Ausgabedateien inkrementiert, für Eingabedateien dekremetiert
CURRENT SEG	gibt die Nummer des Dateisegmentes an, das momentan gelesen/geschrieben wird

Teil 4 MB-KASSETTE, EINGABE



CHAR LAST BLK enthält die Anzahl von Informationszeichen im letzten Block

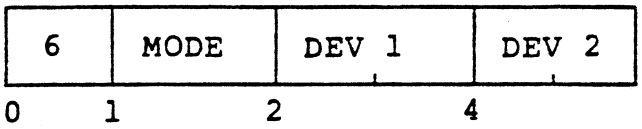
Teil 5 MB-KASSETTE, AUSGABE



FIRST TRK	Nummer der Spur, in die das erste Dateisegment geschrieben wurde
FILE NO	Dateinummer des ersten Segmentes auf dieser Spur

TRACK USE INFO ist ein Duplikat der Bytes 1 - 15 des Kassetten-Belegungsblockes (vergl. Anhang F)

DOPPEL-GERÄT



DEV1 und DEV2 sind Verweise auf andere Kontrollblöcke (beides Ein- oder Ausgabedateien)

```
SIEMENS 610 ASSEMBLER V2.2        *****  HARDCOPY ** BILDSCHIRM AUF DRUCKER  *****02.07.79*CNDL*

                          1 $TITLE      *****  HARDCOPY ** BILDSCHIRM AUF DRUCKER  *****02.07.79*CNDL*
                          2 ;                              MIT LOGISCHER E/A
                          3 ;                              AUSNAHME LESEN VOM BILDSCHIRM
                          4 $PW110
                          5 $PL48
                          6           CSEG
FF00                      7 FCBCI     EQU     0FF00H
FF02                      8 FCBCO     EQU     0FF02H
FF08                      9 FCBSL     EQU     0FF08H
FD09                     10 INCHAR    EQU     0FD09H
FD0C                     11 OUTCHAR   EQU     0FD0CH
006A                     12 INDISP    EQU     6AH        ;PHYSIKALISCH
000D                     13 CR        EQU     13
000A                     14 LF        EQU     10
000C                     15 FF        EQU     12
001D                     16 HOME      EQU     29
0000 C  2A2A2A2A         17 ENDZ:     DB      '**************************************************'
0032 C  2A2A2A2A         18           DB      '********************************',00
0051 C  3E1D             19 ANF:      MVI     A,HOME     ;
0053 C  2A02FF           20           LHLD    FCBCO      ;H/L = ADRESSE DES FCB FUER CO
0056 C  CD0CFD           21           CALL    OUTCHAR    ;ZEICHENAUSGABE
0059 C  0618             22           MVI     B,24       ;ZEILENZAEHLER LADEN
005B C  0E50             23 LOOP1:    MVI     C,80       ;ZEICHENZAEHLER LADEN
005D C  CD6A00           24 LOOP2:    CALL    INDISP     ;EIN ZEICHEN VOM BILDSCHIRM LESEN
0060 C  FE20             25           CPI     20H        ;ZEICHEN < BLANC ?
0062 C  D26700    C      26           JNC     AUSG       ;NEIN ----> DRUCKEN
0065 C  3E20             27           MVI     A,20H      ;JA  ZCHN ==> SPACE
0067 C  2A08FF           28 AUSG:     LHLD    FCBSL      ;H/L = ADRESSE DES FCB FUER SL
006A C  CD0CFD           29           CALL    OUTCHAR    ;ZEICHEN DRUCKEN
006D C  0D               30           DCR     C          ;ZEICHENZAEHLER - 1
006E C  C25D00    C      31           JNZ     LOOP2      ;---> LOOP2 SOLANGE ZEICHENZ.NICHT 0
0071 C  3E0D             32           MVI     A,CR       ;
0073 C  2A08FF           33           LHLD    FCBSL      ;
0076 C  CD0CFD           34           CALL    OUTCHAR    ;WAGENRUECKLAUF AUSGEBEN
0079 C  3E0A             35           MVI     A,LF       ;
007B C  2A08FF           36           LHLD    FCBSL      ;
007E C  CD0CFD           37           CALL    OUTCHAR    ;ZEILENVORSCHUB AUSGEBEN
0081 C  05               38 .         DCR     B          ;ZEILENZAEHLER - 1
```

```
SIEMENS 610 ASSEMBLER V2.2          *****  HARDCOPY ** BILDSCHIRM AUF DRUCKER  *****02.07.79*CNDL*

0082 C  C25B00   C      39           JNZ    LOOP1      ;---> LOOP1 SOLANGE ZEILENZAEHLER NICHT 0
0085 C  3E0C             40           MVI    A,FF       ;
0087 C  2A08FF           41           LHLD   FCBSL      ;
008A C  CD0CFD           42           CALL   OUTCHAR    ;FORMULARVORSCHUB AUSGEBEN
008D C  210000   C       43           LXI    H,ENDZ     ;H/L = TEXTADRESSE
0090 C  7E               44 LOOP3:    MOV    A,M        ;ZEICHEN ---> AKKU
0091 C  B7               45           ORA    A          ;ZEICHEN = 0 ?
0092 C  CAA100   C       46           JZ     ZEIV       ;JA ---> ZEIV
0095 C  E5               47           PUSH   H          ;TEXTADRESSE SICHERN
0096 C  2A02FF           48           LHLD   FCBCO      ;H/L =ADRESSE DES FCB FUER CO
0099 C  CD0CFD           49           CALL   OUTCHAR    ;ZEICHEN AUSGEBEN
009C C  E1               50           POP    H          ;TEXTADRESSE AUS STACK HOLEN
009D C  23               51           INX    H          ;AUSGABE - ADR. + 1
009E C  C39000   C       52           JMP    LOOP3      ;---> NAECHSTES ZEICHEN AUSGEBEN
00A1 C  2A02FF           53 ZEIV:     LHLD   FCBCO      ;
00A4 C  3E0A             54           MVI    A,LF       ;
00A6 C  CD0CFD           55           CALL   OUTCHAR    ;ZEILENVORSCHUB AUSGEBEN
00A9 C  C9               56           RET               ;RUECKSPRUNG ---> BS 610
                         57           END    ANF

NO ERROR MESSAGES
```

```
                        1 $TITLE EIN/AUSGABE - PROGRAMM MIT GETBIN UND PUTBIN  ***04.07.79**CNDL**
                        2 $PW 110
                        3 $PL 48
                        4 ;*********************************************************************
                        5 ;*                                                                   *
                        6 ;*  DATENTRANSFER MIT GETBIN/PUTBIN                                  *
                        7 ;*                                                                   *
                        8 ;*********************************************************************
                        9 START:  CSEG
FF04                   10 FCBSI   EQU    0FF04H;   ADRESSE DES FILE-CONTROL-BLOCKES FUER SI
FF06                   11 FCBSO   EQU    0FF06H;   ADRESSE DES FILE-CONTROL-BLOCKES FUER SO
FD18                   12 GETBIN  EQU    0FD18H;   ADRESSE DER ROUTINE GETBIN
FD1B                   13 PUTBIN  EQU    0FD1BH;   ADRESSE DER ROUTINE PUTBIN
0000 C  00             14 SCHALT: DB     0;        ENDESCHALTER
0001 C                 15 PUFFER: DS     80;       PUFFERBEREICH
                       16 ;
                       17 ;*******LESEN IN DEN PUFFER MIT GETBIN*****************************
                       18 ;
0051 C  2A04FF         19 LESEN:  LHLD   FCBSI;    H/L = ADR. DES FCB FUER SI
0054 C  110100    C    20         LXI    D,PUFFER; D/E = ADR. DES PUFFERS
0057 C  0650           21         MVI    B,80;     B = ZEICHENZAEHLER FUER GETBIN
0059 C  CD18FD         22         CALL   GETBIN;   LESEN IN DEN PUFFER
005C C  D26800    C    23         JNC    SCHREI;   KEIN CARRY = KEIN FEHLER --> SCHREIBEN
005F C  B7             24         ORA    A;        A = 0 ABER CARRY = 1 ?
0060 C  C28200    C    25         JNZ    LSFEHL;   NEIN --> FEHLER BEIM LESEN
                       26 ;
                       27 ;*******END OF FILE ERKANNT  ( CARRY = 1 UND  A = 00 )************
                       28 ;
0063 C  3E01           29         MVI    A,01H;
0065 C  320000    C    30         STA    SCHALT;   ENDESCHALTER SETZEN
                       31 $E
```

```
                        32 ;*******SCHREIBEN AUS PUFFER AUF SO MIT PUTBIN*********************
                        33 ;
0068 C  2A06FF          34 SCHREI: LHLD   FCBSO;    H/L = ADR. DES FCB FUER SO
006B C  110100    C     35         LXI    D,PUFFER; D/E = ADR. DES PUFFERS
006E C  3E50            36         MVI    A,80;     A = MAX. LESEANZAHL
0070 C  90              37         SUB    B;        TATSAECHL. LESEANZ. MINUS 80 = SCHREIBANZ.
0071 C  47              38         MOV    B,A;      B = SCHREIBANZAHL
0072 C  CD1BFD          39         CALL   PUTBIN;   SCHREIBEN AUS PUFFER
0075 C  DA8200    C     40         JC     SRFEHL;   WENN CARRY = 1 ---> FEHLER BEIM SCHREIBEN
0078 C  3A0000    C     41         LDA    SCHALT;   SCHALTER NACH AKKU
007B C  FE01            42         CPI    01H;      SCHALTER GESETZT ? (LETZTER BLOCK)
007D C  C25100    C     43         JNZ    LESEN;    NEIN  --> LESEN NAECHSTEN BLOCK
                        44 ;
                        45 ;*******PROGRAMMENDE*********************************************
                        46 ;
0080 C  B7              47         ORA    A;        CARRY RUECKSETZEN
0081 C  C9              48         RET    ;         ----> BS610
                        49 ;
                        50 ;*******SCHREIB/LESEFEHLERBEHANDLUNG*******************************
                        51 ;
0082 C                  52 LSFEHL: DS     0;        LESEFEHLER
0082 C                  53 SRFEHL: DS     0;        SCHREIBFEHLER
0082 C  C9              54         RET    ;         ---> BS 610 MIT FEHLERMELDUNG
                        55         END    START

NO ERROR MESSAGES
```

▶ ASM

Fehlermeldung	Bedeutung
1	Falscher Befehl (erstes Zeichen)
2	Falscher Befehl (Datenfeld)
3	Falscher Befehl (Opcode-Feld)
4	Syntaxfehler in Pseudoinstruktion
5	Symbol doppelt definiert
6	Symboltabelle voll
7	Ende einer Zeichenkette fehlt
8	Syntaxfehler in einem Ausdruck
9	Falsche Endekennung eines Ausdrucks
10	Fehlender Operand
11	Fehlende Konstante bei DB oder DW
12	Undefiniertes Symbol in einem Ausdruck
13	Fehlendes END
*→ 50	Eingabesatz zu lang
51	Falsche logische Einheit (Source)
52	Falsche logische Einheit (List)
53	Falsche logische Einheit (Objekt)
54	Falsches Assembler-Kommando
55	Zu wenig Platz auf Diskette oder Kassette
56	Disketten- oder Kassettenlaufwerk nicht bereit
57	Schreibsperre auf der Kassette
* 14	Wie Fehler-Nr. 5

▶ RASM

RASM kann die folgenden Fehlermeldungen ausgeben:

01	Syntax-Fehler im Quellcode (nicht zulässiges Zeichen, Feld zu lang)
02	Unbekannter Befehls-Code
03	Zu viele Entries in der Zeile
04	Symbol-Tabelle ist voll
05	Unzulässiger Ausdruck beim ersten Kompilier-durchgang. Symbole in ORG-Ausdrücken müssen definiert sein, bevor sie verwendet werden.
06	EQÜ-Anweisung mit unzulässigem Operanden
07	Unzulässiger Ein-Byte-Ausdruck
08	Fehler bei Sprungmarke (s.Fehler 05)
09	Unzulässiger Ausdruck
10	Fehlender Ausdruck
11	Undefiniertes Symbol
12	Symbol mehrfach definiert
13	Falsche Register-Angabe

Schnittstelle STIMER

Das Betriebssystem bietet die Möglichkeit, bestimmte Programmteile eines Anwenderprogrammes nach Ablauf einer vorgewählten Zeit anzustoßen. Wenn dies gefordert wird, muß die Schnittstelle des BS61Ø

STIMER (CALL ØFD3CH)

benutzt werden.

Eingangsbedingung:

H/L = Adresse des Programmteiles, das nach Ablauf des gewünschten Zeitintervalls ablaufen soll

D/E = Zeitintervall in Vielfachen von 2Øms
(z.B.: LXI D,50 für 1 sec)

Anwendung:
==========

```
STIMER   EQU   OFD3CH

;*****     Programm - Vorlauf

         LXI   H,ZEITA
         LXI   D,100
         CALL  STIMER

;*****     Hauptprogramm Ablauf

;*****     Hauptprogramm Ende

ENDE:    DI
         LXI   H,0
         LXI   D,0
         CALL  STIMER
         EI
         RET

;*****     Start nach Ablauf des Zeitintervalles

ZEITA:   DI
         LXI   H,0
         LXI   D,0
         CALL  STIMER
         EI

;*****     Bearbeitung nach Ablauf des Zeitintervalles

         LXI   D,100
         LXI   H,ZEITA
         CALL  STIMER
         RET
```

Benutzung des Timers

Ein Anwenderprogramm kann sich die interne Uhr der 6.610 zunutze machen, indem es alle 20ms oder Vielfache von 20ms eine vorgegeben Programmroutine (Interruptroutine) durchlaufen läßt. Dazu muß das Anwenderprogramm beim Aufruf der Routine STIMER (Set Timer, FD3CH) im Doppelregister H/L die Adresse seiner Interruptroutine, im Doppelregister DE die Anzahl der 20ms-Einheiten übergeben. Dabei ist folgendes zu beachten:

- In der Interruptroutine müssen die Inhalte der Register und des Akkumulators, die zuvor im Programm benutzt wurden, auf dem Stack gesichert werden, wenn Register und Akkumulator in der Routine benötigt werden. Selbstverständlich müssen die Inhalte am Ende der Unterbrechung wieder zurückgeladen werden.

- Beim Sichern und Rückladen der Register muß man andere Unterbrechungen mit DI verhindern, sollte diese Unterbrechungssperre jedoch möglichst bald wieder mit EI zurücknehmen.

- Die STIMER-Routine muß zu Beginn der Unterbrechungsroutine mit CALL STIMER ,H/L=0, D/E=0 zurückgesetzt werden.

- Am Ende der Unterbrechungsroutine kann STIMER mit neuen Werten, sofern gewünscht, aufgerufen werden.

- Am Ende des Programmes muß STIMER mit H/L=0, D/E=0 aufgerufen werden, um weitere Unterbrechungen zu verhindern.

Außer in der oben beschriebenen Art kann der Timer auch zur automatischen Versorgung der Tageszeit benutzt werden. Zuvor müssen die folgenden Speicher hexadezimal mit den folgenden Werten belegt werden:

27B6H	Monat
27B5H	Jahr
27B7H	Tag
27B8H	Tageszähler (auf 0 setzen)
27B9H	Stunde - 24
27BAH	Minute - 60
27BBH	Sekunde - 60

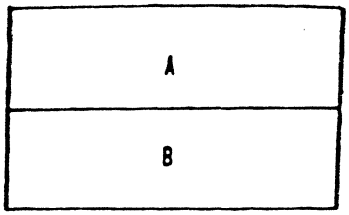
Monat, Jahr und Tag werden nicht verändert, der Tageszähler kann theoretisch bis 255 gezählt werden.

Wenn der 20ms-Zähler auf 50 gesetzt wird, erfolgt jede Sekunde ein Zeitinterrupt und die oben aufgeführten Zähler werden aktualisiert.

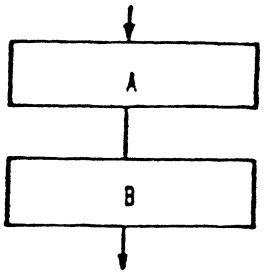
Zur Benutzung des Timers s. Beispiel 3 auf Seite 119.

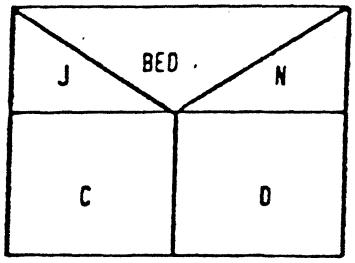
STRUKTOGRAMM

DIN 66001

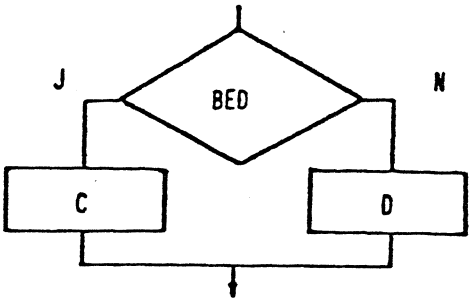


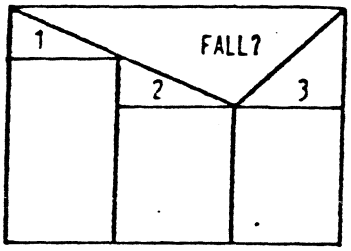
SEQUENZ(FOLGE)



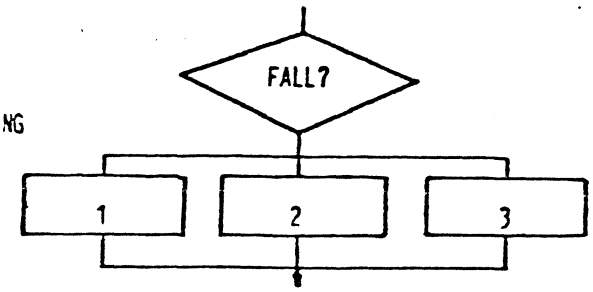


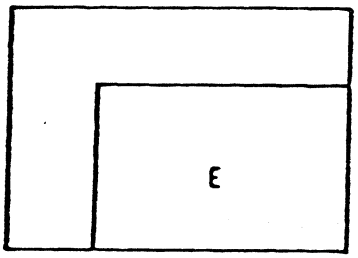
VERZWEIGUNG(AUSWAHL)



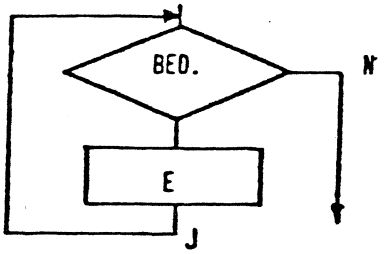


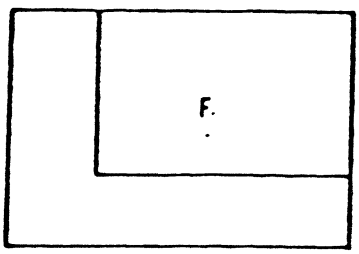
FALLUNTERSCHEIDUNG
(AUSWAHL)



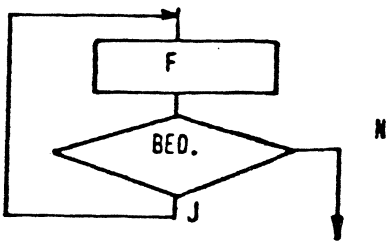


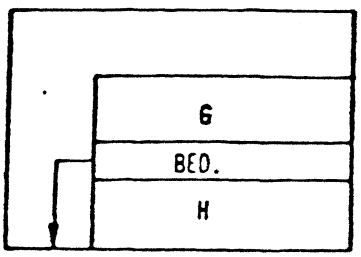
BEDINGTE WIEDER-
HOLUNG



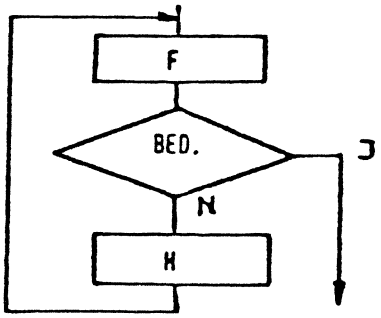


BEDINGTE WIEDER-
HOLUNG



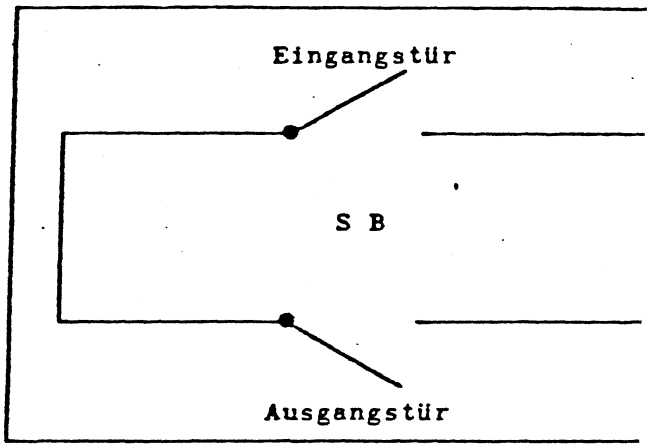


BEDINGTE WIEDER-
HOLUNG

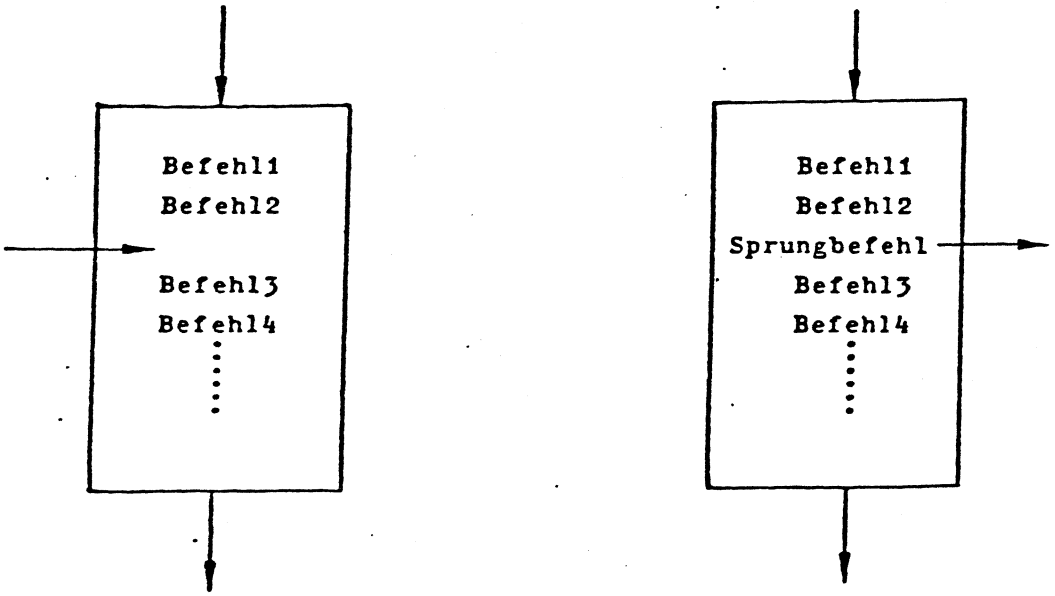


Grundsatz der Zweipoligkeit

Jeder Strukturblock hat genau einen Eingang (dynamischer Anfang) und genau einen Ausgang (dynamisches Ende).

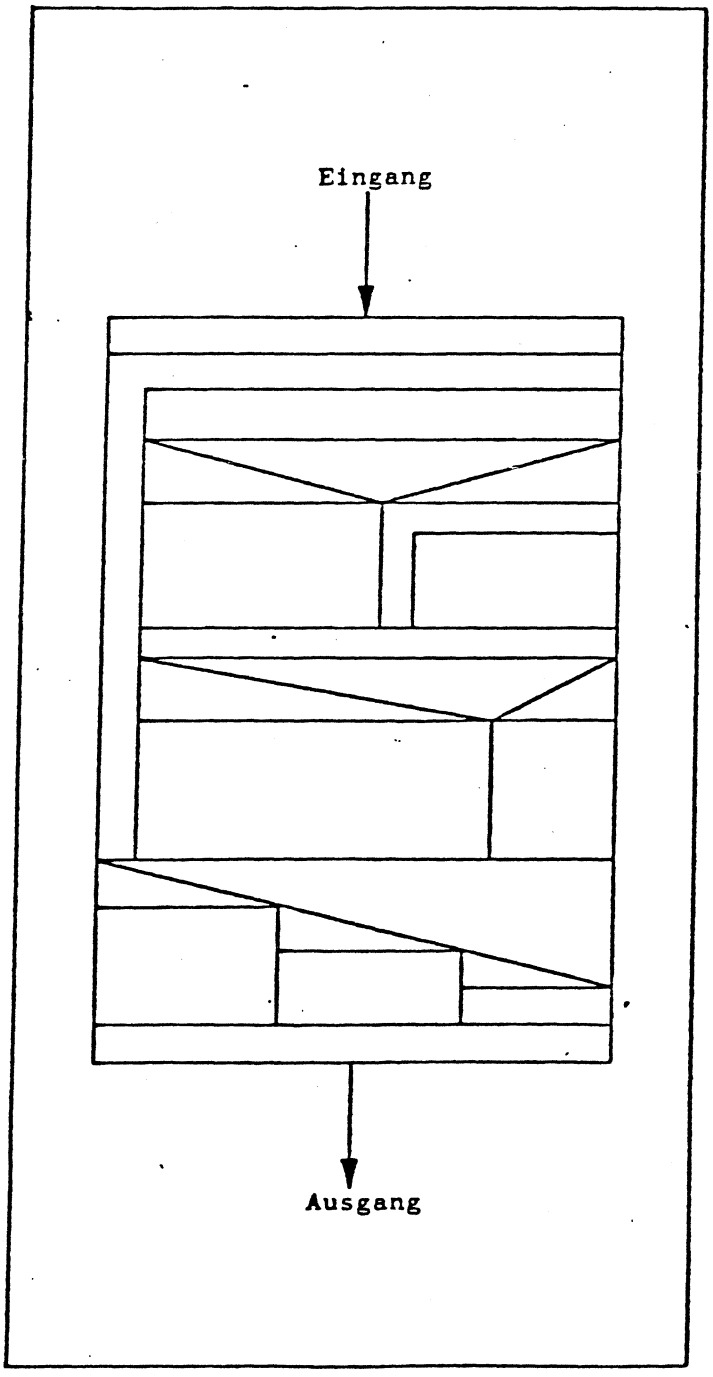


Gegenbeispiele:



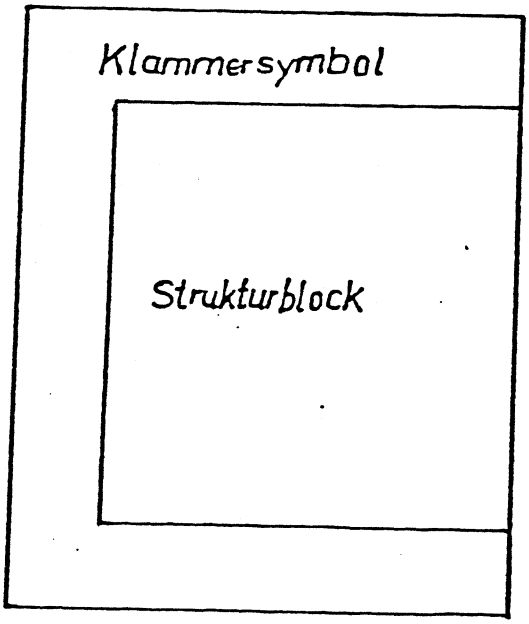
Zusammensetzung von Strukturblöcken

Strukturblöcke einer Ebene dürfen nur durch Aneinanderreihung oder Ineinanderschachtelung zusammengesetzt werden.
Das Ergebnis ist wieder ein Strukturblock und damit wieder ein Zwei-Pol.

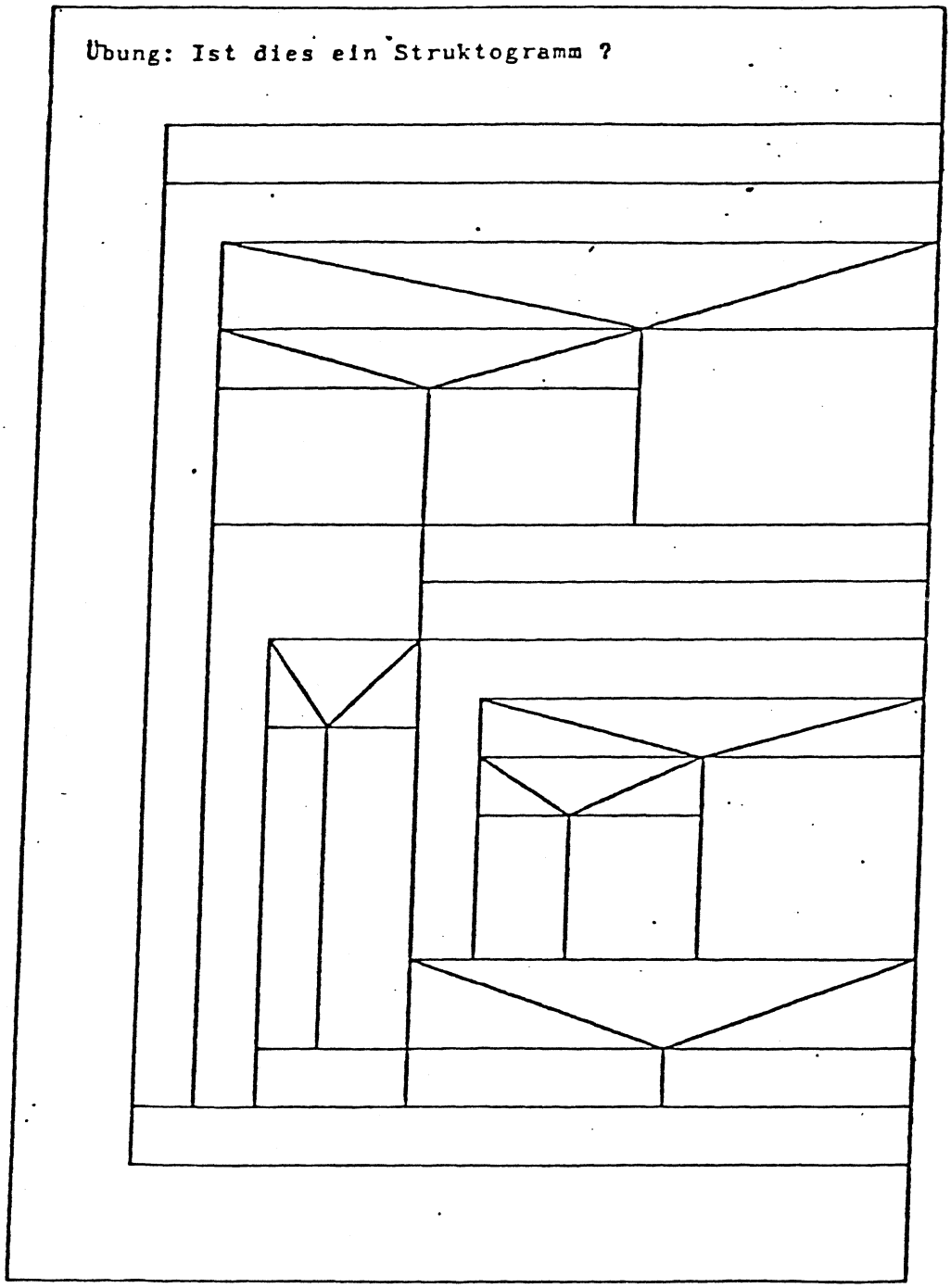


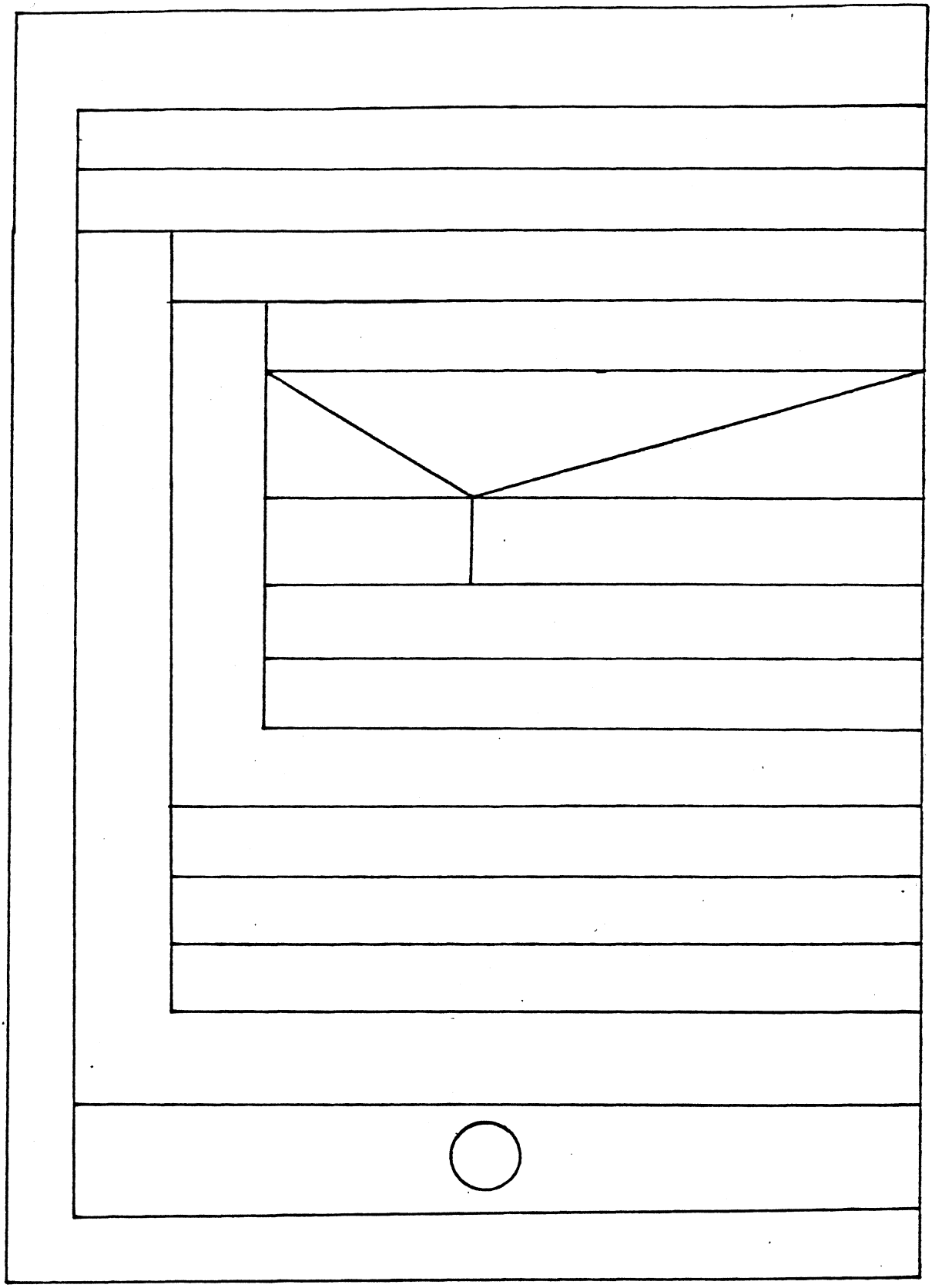
b. Struktogramme

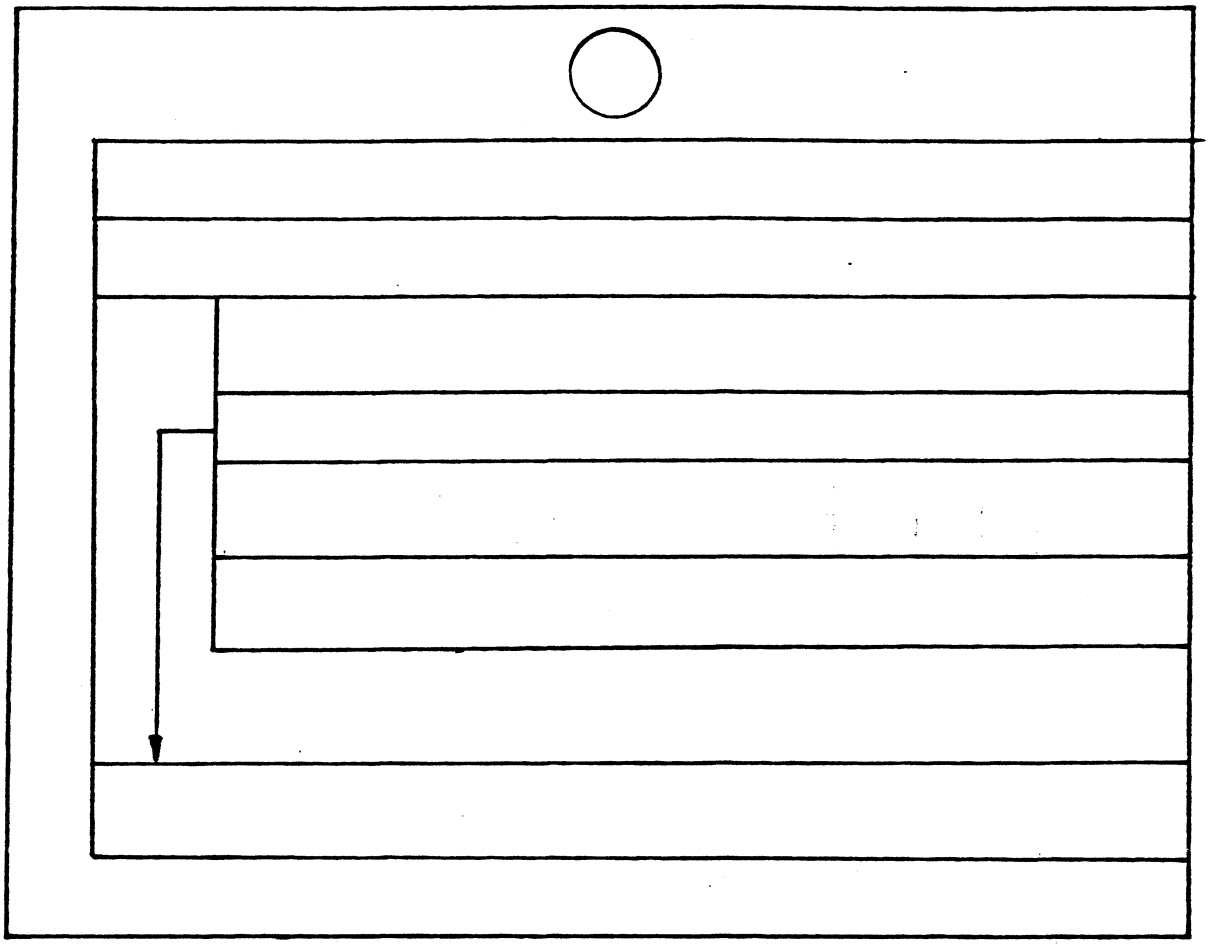
Ein Struktogramm ist eine freie Kombination der vorher beschriebenen logischen Grundstrukturen, wobei das Struktogramm immer von oben nach unten durchlaufen wird. Ein Strukturblock kann in einem anderen vollständig enthalten sein. Verbindungen oder Überlappungen zwischen benachbarten Strukturen sind nicht erlaubt. Ein Struktogramm sollte nicht länger als eine Seite sein. Bei Platzmangel kann durch Lupen das betreffende Programmstück auf einer gesonderten Seite dargestellt werden. Strukturblöcke können mit einem Klammersymbol umschlossen werden, damit für den ganzen Klammerinhalt geltende Angaben dargestellt werden können.



Übung: Ist dies ein Struktogramm ?







```
IEMENS 610 ASSEMBLER V2.2          *****  HARDCOPY ** BILDSCHIRM AUF DRUCKER  *****02.07.79*CNDL*

                          1 $TITLE      *****  HARDCOPY ** BILDSCHIRM AUF DRUCKER  *****02.07.79*CNDL*
                          2 ;                                MIT PHYSIKALISCHER E/A
                          3 $PW110
                          4 $PL48
                          5           CSEG
06A                       6 INDISP    EQU     6AH
049                       7 CURSOR    EQU     49H
052                       8 PRINT     EQU     52H
043                       9 TTO       EQU     43H
00D                      10 CR        EQU     13
00A                      11 LF        EQU     10
00C                      12 FF        EQU     12
01D                      13 HOME      EQU     29
000 C  2A2A2A2A          14 ENDZ:     DB      '*****************************************************'
032 C  2A2A2A2A          15           DB      '*********************************',00
051 C  3E1D              16 ANF:      MVI     A,HOME
053 C  CD4300            17           CALL    TTO         ;CURSOR AN BILDANFANG
056 C  0618              18           MVI     B,24        ;ZEILENZAEHLER LADEN
058 C  0E50              19 LOOP1:    MVI     C,80        ;ZEICHENZAEHLER LADEN
05A C  CD6A00            20 LOOP2:    CALL    INDISP      ;EIN ZEICHEN VOM BILDSCHIRM LESEN
05D C  FE20              21           CPI     20H         ;ZEICHEN < BLANC ?
05F C  D26400    C       22           JNC     AUSG        ;NEIN ----> DRUCKEN
062 C  3E20              23           MVI     A,20H       ;JA  ZCHN ==> SPACE
064 C  CD5200            24 AUSG:     CALL    PRINT       ;ZEICHEN DRUCKEN
067 C  0D                25           DCR     C           ;ZEICHENZAEHLER - 1
068 C  C25A00    C       26           JNZ     LOOP2       ;---> LOOP2 SOLANGE ZEICHENZAEHLER NICHT 0
06B C  3E0D              27           MVI     A,CR
06D C  CD5200            28           CALL    PRINT       ;WAGENRUECKLAUF AUSGEBEN
070 C  3E0A              29           MVI     A,LF
072 C  CD5200            30           CALL    PRINT       ;ZEILENVORSCHUB AUSGEBEN
075 C  05                31           DCR     B           ;ZEILENZAEHLER - 1
076 C  C25800    C       32           JNZ     LOOP1       ;---> LOOP1 SOLANGE ZEILENZAEHLER NICHT 0
079 C  3E0C              33           MVI     A,FF
07B C  CD5200            34           CALL    PRINT       ;FORMULARVORSCHUB AUSGEBEN
07E C  210000    C       35           LXI     H,ENDZ      ;H/L = TEXTADRESSE
081 C  7E                36 LOOP3:    MOV     A,M         ;ZEICHEN ---> AKKU
082 C  B7                37           ORA     A           ;ZEICHEN = 0 ?
083 C  CA8D00    C       38           JZ      ZEIV        ;JA ---> ZEIV
```

```
IEMENS 610 ASSEMBLER V2.2          *****  HARDCOPY ** BILDSCHIRM AUF DRUCKER  *****02.07.79*CNDL*

086 C  CD4300          39           CALL   TTO        ;ZEICHEN AUSGEBEN
089 C  23              40           INX    H          ;AUSGABE - ADR. + 1
08A C  C38100    C     41           JMP    LOOP3      ;---> NAECHSTES ZEICHEN AUSGEBEN
08D C  3E0A            42 ZEIV:     MVI    A,LF
08F C  CD4300          43           CALL   TTO        ;ZEILENVORSCHUB AUSGEBEN
092 C  C9              44           RET               ;RUECKSPRUNG ---> BS 610
                       45           END    ANF

0 ERROR MESSAGES
```

Indexsequentielles Zugriffssystem

ISAM/DA

Allgemeines

Wie bereits bekannt ist, bietet das logische Ein-/Ausgabesystem (ladbarer Teil) Schreiben und Lesen auf Dateiebene. Dabei können alle Geräte (Tastatur, Bildschirm, Drucker, Floppy-Disk und Magnetkassette) bedient werden. Dies gilt für Programme, die in den drei dz. vorhandenen Programmiersprachen ASSEMBLER, BASIC oder COBOL erstellt wurden.
Bei sehr vielen Aufgaben müssen die Daten, die sich auf externen Massenspeichern befinden, einer Änderung unterworfen werden (z.B.: Fortschreibung der Bruttogehälter bei Lohnabrechnungsprogrammen). Dies ist jedoch mit dem logischen EAS des BS61Ø nur sehr umständlich möglich, da ein gelesener Datensatz nicht mehr auf die selbe Stelle zurückgeschrieben werden kann. Eine Lösung wäre nur in 4 Schritten möglich

1) Kopieren aller Datensätze bis zum zu ändernden Datensatz (Programmschleife: Lesen über SI=alte Datei, Schreiben über SO=neue Datei)
2) Lesen des zu ändernden Datensatzes, Ändern, Schreiben in neue Datei
3) Kopieren aller Datensätze bis EOF
4) Löschen der alten Datei

Ohne den Einsatz von Zusatzsoftware oder dem unzulässigen Umgehen der Betriebssystemsoftware ist eine andere Arbeitsweise nicht möglich.
Um jedoch auch derartige Anwenderprobleme mit der 6.61Ø lösen zu können, wurde ein Zusatzsoftwarebaustein entwickelt:

ISAM/DA

ISAM/DA (Indexsequential acces method/ Direct Acces) ermöglicht je nach Wunsch den Zugriff zu Datensätzen in einer FD-Datei auf zwei verschiedenen Wegen:

1) Zugriff über einen Schlüssel (Index)
2) Zugriff über eine Satznummer (DA)

Direktzugriff (DA)

Der Direktzugriff ermöglicht das Zugreifen zu einem Datensatz über die Satznummer. Damit kann jeder beliebige Datensatz mit nur einem FD-Zugriff geschrieben, gelesen oder gelöscht werden.

Indexzugriff (ISAM)

Der Indexzugriff ermöglicht sowohl sequentiellen (fortlaufenden) Zugriff zu Datensätzen, als auch den Zugriff über einen Satzschlüssel (Index). Es sind je nach Dateiaufbau u.U. mehrere FD-Zugriffe notwendig, um Sätze zu lesen, zu löschen oder zu schreiben.

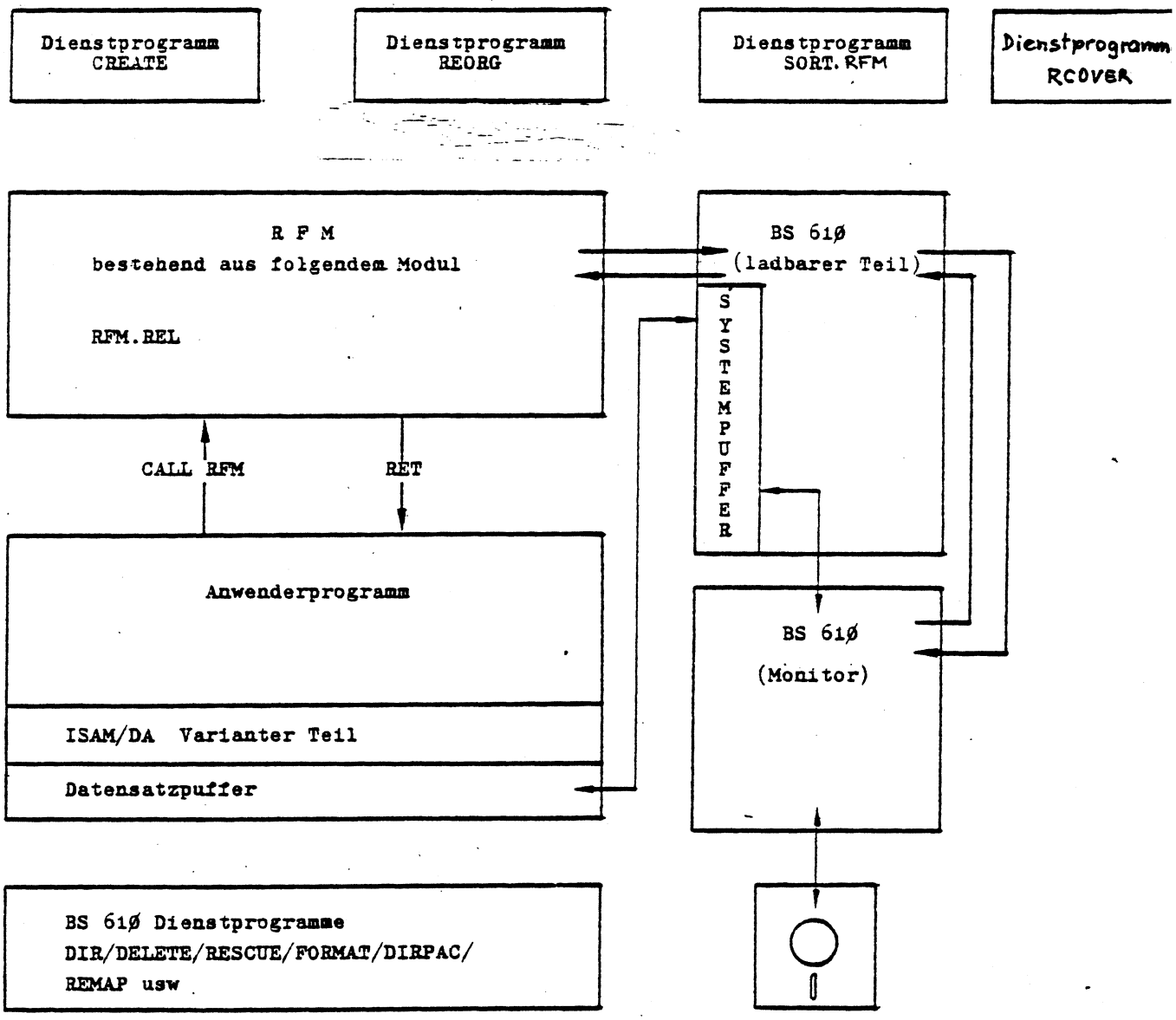
In beiden Fällen wird jedoch sichergestellt, daß ein gelesener Datensatz verändert oder unverändert zurückgeschrieben werden kann.

Wesentliche Merkmale:

ISAM/DA
- Mehrteilige Dateien (auch über mehrere Discketten)
- Feste Satzlänge (1 - 250 Byte)
- Sätze werden auf der Diskette geblockt (Platzersparnis)
- Fortlaufender Dateibereich (Spuren)

Bei ISAM
- Schlüssellänge 1 - 24 Byte an beliebiger Stelle im Satz
- Teilqualifizierter Zugriff ist möglich
- Wiederherstellung gelöschter Sätze
- Dateireorganisation notwendig

Grundsätzlicher Aufbau ISAM/DA



Aufbau eines Benutzerprogrammes

Allgemeiner Vorlauf
Anschluß an ISAM/DA-Baustein
Eröffnen einer Datei
ISAM/DA Zugriff und Verarbeitung Schreiben Lesen Löschen Zurückschreiben (gelöschten Satz wieder herstellen)
Datei schließen
allgemeiner Nachlauf
Allgemeiner Konstanten und Variablen Bereich
ISAM/DA Parameterblöcke
ISAM/DA Variable
ISAM/DA Varianter Teil

DEMONSTRATIONSPROGRAMM für RFM-(BASIC)

Auf Systemscheibe enthalten:

RFM.BAS
BRFM (Anschlußbaustein BASIC-RFM)
BSTART

Arbeitsablauf:

1/ Datei(en) anlegen

2/ Anwenderprogramm (Demonstrationspr.) laden

3/ Durchführen von Operationen

4/ Beenden des Anwenderprogrammes

5/ Event. Datei(en) reorganisieren (Index-Datei)

6/ Event. Datei(en) sortieren

Arbeitsablauf BEISPIEL

LW Ø : Systemscheibe BS 610 Vers. 1.54
BASIC Vers. 1.4
RFM Vers. 1.43

LW 1 : Datenscheibe (Intel-Form.)

1.) Anlegen von Dateien mit Dienstpr. CREATE

1a.) DIREKTDATEI : BOB2 (Teilnehmerliste 2er-BOB)

```
CREATE $ D, :F1: BOB2, 81, 30, 1
```

Systemmeldung : CREATE COMPLETED

1b.) INDIZIERTE DATEI : TEL (Telefonliste)

```
CREATE $ I, :F1: TEL, 81, 20, 5, 12, Ø
```

SYSTEMMELDUNG : CREATE COMPLETED

2./ Laden des Anwenderprogrammes :

```
BSTART
   :
START   RFM.BAS
```

3./ Auf Bildschirm erscheint eine Maske mit den Operationscodes der ausführbaren Operationen

Durchführen von Operationen

opcode Ø
opcode 6 —— Datei eröffnen
:
schreiben
lesen
:
opcode 7 —— Schließen Datei(en)

} Anwender-HB. Kap. 4

4.) Beenden des Anwenderprogrammes:

(Nach Schließen der Dateien opcode 7)

opcode 99

F O R M U L A R S P R A C H E

Inhaltsverzeichnis:

FORMULARSPRACHE (allgemeiner Teil)

z.Zt. gibt es zwei freigegebene Versionen

1. Formularsprache (alt)

```
==========================
=    PROD-NR.:  610 - 80 =
==========================

FORMULARSPRACHE 1 VERS. 1.2
(unveraendert)
--------------------------
FORMS            56     7030 W
```

2. Formularsprache 2 (neu)

```
==========================
=    PROD-NR.:  610 - 85 =
==========================

FORMULARSPRACHE 2 VERS. 1.1
---------------------------
FORM              69     8498 W
DM                89    11108 W
FORM   .DAT       98    12200 W
FORM   .D          2       84 W
CREATE            25     3003 W
REORG             72     8871 W
RCOVER           101    12576 W
```

Beim Arbeiten mit der Formularsprache muß sich der Bildschirm der 6.610 im PAGE-MODE befinden.
(Display-Logic 1 S9→OFF)

FORMULARSPRACHE UNVERÄNDERT (alt)

Sie steht als Dienstprogramm zur Verfügung.
Sie wird mit 'FORMS' aufgerufen.

Die Sprache hat zwei Ebenen:

Formularebene: In der Formularebene werden die Formulare erstellt, eventuell geändert und gespreichert.

Datenebene: In der Datenebene wird mit den Formularen gearbeitet:
Sie werden ausgefüllt und die Daten werden geschrieben.

Nach dem Laden der Formularsprache (Aufruf: FORMS) verzweigt das Programm in die Formularebene.

Der Bildschirm gliedert sich in 2 Teile:

1. Zeile - Statuszeile
2. Zeile - } Formularbereich
⋮ - }
25. Zeile - }

Die Statuszeile zeigt u.a. die Ebene, in der sich das Programm befindet ('FORM' bzw. 'DATA')

FORMULARSPRACHE 2

Sie besteht aus mehreren Modulen. (Dateien)

Das Formularerfassungssystem und das Datenerfassungssystem sind getrennte Programme.

Formularerfassungssystem

In dieser Ebene werden Formulare erstellt, evt. geändert und gespeichert.

Sie ist unterteilt in Formularerstellungsebene
und in Korrekturebene.

Das Formularerfassungssystem wird mit dem Aufruf 'FORM' geladen.

Nach dem Laden verzweigt das Programm automatisch in die Formularerstellungsebene.

Der Bildschirm gliedert sich in 2 Teile:

Zeile 1, 2	-	Statuszeilen
Zeile 3-25	-	Formularbereich

Jetzt können Formulare erstellt, aufgerufen bzw. abgespeichert werden. Pro Fomularsatz können z.Zt. max. 5 Formulare erstellt werden. Der Formulardateiname darf keine Erweiterung enthalten.

Ausgehend von der Formularerstellungsebene ist es möglich, für Änderungen oder Korrekturen in die sog. Korrekturebene zu verzweigen.

ERSTELLEN EINES FORMULARS: siehe Praxisleitfaden
und Bedienungsanleitung

Datenerfassungssystem

Das Datenerfassungssystem besteht aus den Dienstprogrammen:

'DM'

'FORM.D'

'FORM.DAT'

Das Datenerfassungssystem arbeitet mit einem Formularsatz, der mit dem Formularerfassungssystem erstellt wurde.

Es bietet die Möglichkeit der

- Datenerfassung
- Änderung von bereits erfaßten Daten
- Datenverwaltung, d.h. Löschen von best. Datensätzen, sowie das Suchen von
  - bestimmten Sätzen
  - Strings innerhalb von Datensätzen

Das Datenerfassungssystem wird mit dem Aufruf 'FORM.D' geladen und gestartet.

Das Datenerfassungssystem arbeitet mit Dateien, die mit dem Dienstprogramm 'CREATE' erstellt wurden, d.h. vor dem Laden des Erfassungssystems muß eine indexsequentielle Datei eingerichtet werden.

Das Dienstprogramm 'CREATE' wird wie folgt aufgerufen:

CREATE$D,(physik. Einheit)(Dateiname·Erweiterung),250,(Nr.),1

- Nr.: gibt die Anzahl der Datensätze an.
- Dateiname.Erweiterung: muß gleich dem Namen des entsprechenden Formulars sein und muß zusätzlich eine Erweiterung, die nur aus Zahlen bestehen darf, haben.

ÜBUNG MIT DEM DATENERFASSUNGSSYSTEM

siehe Praxisleitfaden und Bedienungsanleitung

Einschränkungen, sowie Besonderheiten der jeweiligen Programmversion entnehmen Sie bitte den Lieferfreigabe-Mitteilungen.

FORMULARSPRACHE 2

Praxisleitfaden

Verwenden Sie dazu auch die Broschüre Bedienungsanleitung Formularsprache 2 (Anwenderhandbuch, Best.-Nr. D41/1220)

A) Formularerfassungssystem

Ein Formular ist nach nachstehendem Entwurfsblatt zu erfassen, abzuspeichern und auszudrucken.

B) Datenerfassungssystem

Erstellen Sie eine indexsequentielle Datei für die Datenerfassung.

Erfassen Sie mehrere Datensätze in das in A) erstellte Formular.

Nach Abschluß der Übungen im Datenerfassungsmodus ist die Datei zu schließen.

Laden Sie anschließend das Datenerfassungssystem neu und üben Sie mit dem Suchmodus und Änderungsmodus.

Bildschirm-Computer 6.610

Bildschirm - Entwurfsblatt

Programm **ÜBUNGSBEISPIEL - Formular - Entwurf**

Bearbeiter _____ Datum _____

Blatt ___

```
1 [T] normal ~>
2
3 AN                          AUFTRAG
4 SCHULE FUER D
5 EWZ FUER LEHRMEDIEN     VON: [EV] invers/alpha/muß/dupl. [ET] BEARBEITER: [EV] invers/alpha/muß [ET] TEL. [EV] inv./num. [E]
6 [T] normal ~>
7
8 AUFTRAGSART: [EV] invers/alphanum/muß                                    [E]
9 [T] normal
10                [EV] invers/alphanum/                                    [E]
11 [T] normal
12                [EV] invers/alphanum/                                    [E]
13 [T] normal
14
15 AUFTRAGSUMFANG: [EV] invers/alphanum/muß                                [E]
16 [T] normal
17                [EV] invers/alphanum                                     [E]
18 [T] normal
19
20 GEWUENSCHTER
21 LIEFERTERMIN: [EV] invers/alphanum/muß/links                            [E]
22 [T] normal
23 DATUM [EV] invers/alphanum/links          [ET] normal
24
25 BLATTNR. [EV] invers/num/muß/rechts/Nullauff. [ET]                      [E]
```

[T] - Geschütztes Feld

[V] - Ungeschütztes Feld

[E] - Feldendezeichen

14/80