

SIEMENS

Betriebs-
system

Dienst-
programme

Sprach-
systeme

Subsysteme

DFÜ-Software

Hardware

Anhang

AMBOSS 1

Systemtaschenbuch

Bestell-Nr. U5-J-Z25-
Printed in the Federal
Republic of Germany
27 AG 481.5 (4600)

VORWORT

Was enthält das 6.610 Systemtaschenbuch?

In diesem Taschenbuch sind in konzentrierter Form, alle wesentlichen Fakten zu einem Katalogwissen zusammengestellt, die zur Bedienung des Bildschirm-Computer 6.610 und der dazugehörigen System- und systemnahen Software benötigt werden. Es enthält unter anderem eine Darstellung des Formats und eine kurze Beschreibung der Funktion für jedes Kommando, jeden Aufruf, und jede Anweisung.

Das Taschenbuch besteht aus den Registern:

1. Betriebssystem
2. Dienstprogramme
3. Sprachsysteme
4. Subsysteme
5. DFU-Software
6. Hardware
7. Anhang

An wen wendet sich dieses Systemtaschenbuch?

Es soll denjenigen als Nachschlagewerk dienen, die bereits die ausführliche Dokumentation für den Bildschirm-Computer 6.610 kennen.

Wie ist der Aufbau dieses Systemtaschenbuches?

Jedem Register ist ein Inhaltsverzeichnis vorangestellt. Ein Druckregister erleichtert das Auffinden der zu jedem Kapitel gehörenden Seiten die nach Register/Kapitel/Seite benummert sind. Zudem enthält jede Seite die Paragraphen- und evtl. die Kapitelüberschrift als Kolumnentitel. Die letzte Seite des Systemtaschenbuches ist ein aufklappbares Faltblatt, das Auskunft über die Notation des Buches gibt.

Wie wird das Systemtaschenbuch aktualisiert?

Es werden Nachträge erstellt, bestehend aus den Seiten die entsprechend der Seitennumerierung zu ersetzen bzw. hinzuzufügen sind. Diese Nachträge werden bei Erscheinen bekannt gegeben und können über das ZVW Lager Fürth, unter der entsprechenden Bestellnummer, bestellt werden.

Anregungen, Korrekturvorschläge oder Änderungswünsche richten Sie bitte formlos an:

SIEMENS AKTIENGESELLSCHAFT
D Bi VS 22
Otto-Hahn-Ring 6
8000 München 83.

D Bi VS 22

März 1981

INHALT**REGISTER 1****BETRIEBSSYSTEM**

KAPITEL 1	MONITOR	Seite
1.	MON/D	1-1-1
1.1	Kommandos	.
1.2	Aufrufbare Routinen	1-1-3
1.3	Dynamischer Pufferaufbau	1-1-7/8

KAPITEL 2 **LADEBARES BETRIEBSSYSTEM**

1.	BS1	
1.1	Laden des Betriebssystem	1-2-1
1.2	Reservierter Systembereich	.
1.3	Aufrufbare Routinen	1-2-2
1.4	Interne System-Routinen	1-2-6
1.5	Tabellenzeiger	1-2-7
1.6	Systemzeiger	.
1.7	Logische Einheiten	1-2-8
1.8	Physikalische Einheiten	.
1.9	Datei-Kontrollblock	1-2-9
1.10	Fehlermeldungen	1-2-14

MONITOR/D

COMMANDOS

1. MON/D VERS. 6.0
1.1 Kommandos

Kommando	Syntax	Bedeutung
S	Sadr <u>SP</u> Sadr <u>SP</u> hexwert CR	<ul style="list-style-type: none"> Ausgabe einer Speicherstelle <ul style="list-style-type: none"> Weitere Ausgabe der Speicherstellen mit SP Rücksprung in Eingabemodus "." mit CR Änderung einer ausgegebenen Speicherstelle
X	X reg reg: X : Alle Register A B C D E H L : Einzelregister P : Programmzähler S : Kellerzeiger (Stackpointer) F : Statusregister S=Sign (1=minus) Z=Zero (1=Zero) A=Hilfscarry (1=gesetzt) P=Parity (1=Parität) C=Carry (1=gesetzt) F=Flagregister X reg hexwert CR	<ul style="list-style-type: none"> Ausgabe eines Registers (oder aller Register) Änderung eines angezeigten Registers

Kommando	Syntax	Bedeutung
G	G Gadr1 CR GSPadr2 CR Gadr1 SPadr2 CR GSPadr2 SPadr3 CR Gadr1 SPadr2 SPadr3 CR	<ul style="list-style-type: none"> Starten eines Programmes adr1 = Startadresse (wenn nicht angegeben = aktuelle Adresse) <p style="text-align: center;">$\left.\begin{matrix} \text{adr2} \\ \text{adr3} \end{matrix}\right\} = \text{Haltepunkte}$</p>
D	Dadr,l CR	<ul style="list-style-type: none"> Ausgabe eines Speicherbereiches der angegebenen Länge
F	Funt.adr,ob.adr,wert F1,0,0	<ul style="list-style-type: none"> Füllen des Speicherbereiches mit dem angegebenen Wert Löschen des gesamten Speicherbereiches (Füllen mit 0) und Neuladen des Systems.
L	L:Fn:programm CR n = Laufwerknummer (0 → 3)	<ul style="list-style-type: none"> Laden eines binären Programmes
R	Rn n = Laufwerknummer (0 → 3)	<ul style="list-style-type: none"> Laden des BS I
A	A CR	<ul style="list-style-type: none"> Füllen des Bildschirmes mit allen darstellbaren Zeichen (=Hardware-Testhilfe)
T	T CR	Teletype Funktion
H	H CR	Ausgabe der Monitor-Version

1.2 Aufrufbare Routinen

Adresse (Hex)	Routine	Einsprung-Parameter	Ergebnis
40	TTI	-	Einlesen eines Zeichens von Tastatur und Ablieferung in A
43	TTO	A: Zeichen	Ausgabe des Zeichens aus A auf Bildschirm. Zeichen <LFH werden als Kontrollzeichen interpretiert.
46	TTONC	A: Zeichen	Wie TTO; jedoch werden <u>alle</u> Zeichen als Datenzeichen interpretiert.
49	CURSOR	H: Zeile (0 → 24) L: Spalte (0 → 79)	Positionierung auf die gewünschte Stelle.
4C	MON	-	Ansprung des Monitors (ohne Rücksprung) (mit JMP)
4F	ERROR	A=XX: Fehlercode	Fehlermeldung: ERRORXX auf CO Rücksprung ins Anwenderprogramm, wenn CY = rückgesetzt.
52	PRINT	A: Zeichen	Ausgabe des Zeichens aus A auf Drucker.
55	DTST	-	Test des Ausführungszustandes einer Disketten-Operation Ergebnis im A-Register bzw. Carrybit: Carrybit nicht gesetzt : Operation läuft noch Carrybit gesetzt : Operation beendet A-Register gleich Null : Operation normal beendet A-Register ungleich Null: Operation fehlerhaft beendet Fehlercode im A-Register

Adresse (Hex)	Routine	Einsprung-Parameter	Ergebnis
58	DCAL	A:Laufwerknummer	Anstoß zur Positionierung auf Spur 0 der Diskette. Die Kontrolle geht an das Anwenderprogramm zurück, <u>bevor</u> die Operation beendet ist. Vor Anstoß einer neuen Diskettenoperation muß die Routine DSWT aufgerufen werden.
5B	DSRD	A :Laufwerknummer B :Spur C :Sektor H/L:Pufferadresse	Anstoß zum Lesen eines Diskettensektors Die Kontrolle geht ans Anwenderprogramm zurück, <u>bevor</u> die Operation beendet ist. Vor Anstoß einer neuen Disketten-Operation muß die Routine DSWT aufgerufen werden.
5E	DSWR	wie DSRD	Anstoß zum Schreiben eines Diskettensektors Wie DSRD, jedoch Schreiboperation.
61	DSDL	A:Laufwerknummer B:Spur C:Sektor	Anstoß zum Löschen eines Diskettensektors Wie DSWR, jedoch wird die Kennung DELETED DATA (Daten gelöscht) in die Datenadreßmarke des betreffenden Sektors geschrieben.
64	DSRDA*)	wie DSRD	Anstoß zum Lesen eines Sektors und Konvertierung von EBCDIC nach ASCII. Wie DSRD, jedoch mit Codeumwandlung nach dem Lesen.

Adresse (Hex)	Routine	Einsprung-Parameter	Ergebnis
67	DSWRA ^{*)}	wie DSRD	Konvertierung von ASCII nach EBCDIC und Anstoß zum Schreiben eines Disketten-Sektors. Wie DSWR, jedoch mit Konvertierung vor dem Schreiben.
6A	INDISP	-	Lesen des Zeichens vom Bildschirm, auf dem der Cursor steht. Letzter wandert ein Feld weiter.
6D	DSWT	-	Warten auf den Abschluß einer Disketten-Operation Status wird im A-Register abgeliefert. Normaler Ablauf der Operation: Carrybit und A-Register sind 0. Fehler bei der Operation: Carrybit gesetzt, Fehlercode in A.
70	DCALW	A:Laufwerknummer	Wie DCAL (Positionieren auf Spur 0), jedoch mit Warten auf den Abschluß der Operation (Funktion der Routine DSWT)
73	DSRDW	Wie DSRD	Wie DSRD (Lesen eines Disketten-Sektors), aber mit anschließendem DSWT.
76	DSWRW	Wie DSRD	Wie DSWR (Schreiben eines Sektors), aber mit anschließendem DSWT.
79	DSDLW	Wie DSDL	Wie DSDL (Löschen eines Sektors), aber mit anschließendem DSWT.

Adresse (Hex)	Routine	Einsprung-Parameter	Ergebnis
7C	DSRAW ^{*)}	Wie DSRD	Wie DSRDA (Lesen eines Sektors und Konvertierung nach ASCII), aber mit anschließendem DSWT.
7F	DSRAW ^{*)}	Wie DSRD	Wie DSWRA (Konvertierung nach EDCDIC und Schreiben eines Sektors), jedoch mit anschließendem DSWT.
82	RCVI	-	Empfang eines Zeichens vom Hauptrechner (UART) und Ablieferung im Register A. Das höchstsignifikante Bit (80H) des Zeichens wird auf 1 gesetzt, wenn während der Übertragung ein Fehler festgestellt wurde (Parity, Überlauf, Zeichenrahmenfehler)
85	LOCDIR	A : Laufwerknummer H/L: Adresse eines 9 Zeichen langen Namenfeldes (nicht benutzte Stellen = 00)	Lokalisiert den Eintrag eines Dateinamens im Inhaltsverzeichnis einer INTEL-formatierten Diskette H/L-Adresse aus BUF2 (2600H) wo der Eintrag im Inhaltsverzeichnis steht TRCK(2731H)-Adresse aus BUFI (2680H) wo Spur, Sektor des nächsteinzulesenden ISIS-DIR-Sektors steht
88	LOAD	Wie LOCDIR	Laden eines Programmes von einer INTEL-formatierten Diskette H/L: Startadresse des Programmes.
8B	LINK	Wie LOCDIR	Laden und Starten eines Programmes von einer INTEL-formatierten Diskette. Rücksprung erfolgt nur im Fehlerfall.
CA	XMIO	A: Zeichen	Senden des Zeichens aus dem UART Sendepuffer zum Hauptrechner A wird verändert.

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



1.3 Dynamischer Pufferaufbau

- Tastaturpuffer : 2780H - 279FH
- UART-Empfangspuffer : 25E0H - 25FFH
- UART-Seridepuffer : 25C0H - 25DFH

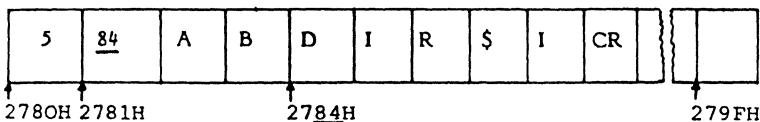
PUFFERAUFBAU

Byte 1 - Zeichenzähler (wie bei MON 4 und MON 5)

Byte 2 - Pufferzeiger = Niedrigere Doppelbytehälfte
des Zeigers auf das nächste abzuholende Zeichen.
Dieses Byte wird binär zu 2700H addiert, um die Adresse des
nächsten abzuholenden Zeichens zu erhalten. Nur gültig, wenn
Bytezähler nicht null. (Unterschiedlich zu MON 4 und MON 5).

Byte 3 -
32 Gepufferte Zeichen in zyklischer Folge.

Bsp Tastaturpuffer



d.h. das erste abzuholende Zeichen ist D.



1. BS 11.1 Laden des Betriebssystem

Rn n = Laufwerknummer (0 → 3)

Druckertreiber

Kennung	Treibervor- spannr.	Treiberroutine	Drucker
/A	MA22.80	DM22.80	80-spaltiger PT80(6322)
/B	MA22.132	DM22.132	132-spaltiger PT80(6322)
/C	MA22.S	DM22.S	PT80 (6322) mit Sparschrift
/D	MABIND	DMBIND	UD3 (6327)
/E	TY24.INT	DT24.INT	DIABLO (6324) mit intern. Zeichenv.
/F	TY24.DEU	DT24.DEU	DIABLO (6324) mit deutsch. Zeichenv.
/G	MA25	DM25	LOGABAX (6325)

- unter Datei DRUCKR-Druckertreibervorspannroutine abspeichern
- fehlt Datei DRUCKR-Standard Druckertreiber für UD3(6327)
wird angenommen (/D)

1.2 Reservierter Systembereich

FD00H - FD3FH Aufrufbare BS610-Routinen

FD40H - FDCCH Sprungtabelle für Monitor Grundfunktionen

FDCDH - FDE4H Interne System-Routinen

FDE6H - FDEBH Tabellenzeiger

FDECH - FDFFH Systemzeiger

FF00H - FF0FH ASSIGNMENT-Tabelle
(log. Einheiten)

FF10H - FFFFH Systemarbeitsspeicher

1.3 Aufrufbare Routinen

Adresse (Hex)	Routine	Einsprung-Parameter	Ergebnis
FD00	BS610	-	Rücksprung in den Kommandomodus des BS610; Dateien werden geschlossen.
FD03	ERROR	-	Fehlermeldung: ERROR XX mit Erklärung auf C0. Rücksprung ins Anwenderprogramm, wenn das CY-Bit rückgesetzt wurde, ansonsten Sprung ins BS1.
FD06	LOAD	H/L: Adresse eines 9 Zeichen langen Namenfeldes (nicht benutzte Stellen = 00) A: Laufwerknummer	$\left\{ \begin{array}{l} CY = 0 \\ H/L: Startadresse des Programmes \end{array} \right.$ $\left\{ \begin{array}{l} CY = 1 \\ A = Fehlercode \end{array} \right.$
FD09	INCHAR	H/L: FCB-Adresse	$\left\{ \begin{array}{l} CY = 0 \\ \text{Zeichen in A} \end{array} \right.$ $\left\{ \begin{array}{l} CY = 1 \\ A = Fehlercode (A = 0 \rightarrow EOF) \end{array} \right.$
FDOC	OUTCHAR	H/L: FCB-Adresse A: Zeichen	$\left\{ \begin{array}{l} CY = 0 - \text{fehlerfreier Ausgang} \\ CY = 1 \\ A = Fehlercode (A = 0 \rightarrow EOF) \end{array} \right.$
FDOF	GET	H/L: FCB-Adresse D/E:Pufferadresse	$\left\{ \begin{array}{l} \text{Einlesen aus Datei bis CR.} \\ DE = \text{verändert} \\ CY = 0 - \text{fehlerfreier Ausgang} \end{array} \right.$ $\left\{ \begin{array}{l} CY = 1 \\ A = Fehlercode (A = 0 \rightarrow EOF) \end{array} \right.$

Adresse (Hex)	Routine	Einsprung-Parameter	Ergebnis
FD12	PUT	H/L: FCB-Adresse D/E: Pufferadresse	Schreiben in Datei bis CR;CR,LF werden angehängt DE = verändert CY = 0 fehlerfreier Ablauf { CY = 1 A = Fehlercode oder A = 0 → EOF
FD15	PUTSTR	H/L: FCB-Adresse D/E: Pufferadresse	Schreiben in Datei bis CR;CR,LF werden nicht angehängt. DE = verändert CY = 0 fehlerfreier Ablauf { CY = 1 A = Fehlercode oder A = 0 → EOF
FD18	GETBIN	H/L: FCB-Adresse D/E: Pufferadresse B : Zeichenzähler	Einlesen einer gewissen Anzahl von Zeichen aus Datei CY = 0 fehlerfreier Ablauf { CY = 1 A = Fehlercode oder A = 0 → EOF
FD1B	PUTBIN	H/L: FCB-Adresse D/E: Pufferadresse B : Zeichenzähler	Schreiben einer gewissen Anzahl von Zeichen D/E = Adresse des letzten Datenbytes +1 B = Rest CY = 0 normales Ende { CY = 1 A = Fehlercode
FD1E	OPEN	H/L: FCB-Adresse A: { 1 Eingabe 2 Ausgabe	Eröffnen einer Datei CY = 0 fehlerfreier Ablauf { CY = 1 A = Fehlercode

Adresse (Hex)	Routine	Einspi...-Parameter	Ergebnis
FD21	READ	H/L: FCB-Adresse	Lesen eines Sektors in den FCB-Puffer CY = 0 - fehlerfreier Ablauf CY = 1 A = Fehlercode
FD24	WRITE	H/L: FCB-Adresse	Schreiben eines Sektors aus dem FCB-Puffer CY = 0 - fehlerfreier Ablauf CY = 1 A = Fehlercode
FD27	CLOSE	H/L: FCB-Adresse	Schließen einer Datei CY = 0 - fehlerfreier Ablauf CY = 1 A = Fehlercode
FD2A	EDBH	A: umzuwendelndes Byte H/L: Adresse eines 2 Byte-Empfangsfeldes	Der Hex-Wert in A wird in ASCII aufbereitet.
FD2D	EDWH	D/E: umzuwendelndes Wort H/L: Adresse eines 4 Byte-Empfangsfeldes	Der Hex-Wert in DE wird in ASCII aufbereitet.
FD30	EDHB	H/L: Adresse eines 2 Byte-Eingabefeldes	A = Hex-Byte Umkehrung von EDBH
FD33	EDHW	H/L: Adresse eines 4 Byte-Eingabefeldes	D/E = Hex-Wort Umkehrung von EDWH

Adresse (Hex)	Routine	Einsprung-Parameter	Ergebnis
FD36	BUFALLOC	H/L: gewünschte Pufferlänge	Pufferbereich zuweisen H/L = Adresse des zugewiesenen Puffers
FD39	BUFDEALLOC	H/L: Pufferadresse	Pufferbereich freigeben
FD3C	STIMER	D/E: gewünschter Zeitintervall in 20 ms-Einheiten H/L: Adresse, die nach Ablauf des Zeitintervales ange- sprungen wird.	

Adresse (Hex)	Routine	Einsprung-Parameter	Ergebnis
FDD3	RELN	A : Anzahl H/L: Adresse einer Liste von Sektoren UNIT(2730H) = Gerätenummer MAP1(2739H) => Spur/Sektor der MAP2(273BH) ISIS.MAP	Freigabe einer Anzahl von Disketten- sektoren
FDD6	RELBLK	B/C: Spur/Sektor UNIT(2730H) = Gerätenummer MAP1(2739H) => Spur/Sektor der MAP2(273BH) ISIS.MAP	Freigabe eines Diskettensektors
DD9	GETN	A : Anzahl H/L: gewünschte Adresse des 1. Sektors UNIT(2730H) = Gerätenummer MAP1(2739H) => Spur/Sektor MAP2(273BH) ISIS.MAP	Reservierung einer Anzahl von Disketten- sektoren
FDDC	GETBLK	B/C: Spur/Sektor UNIT(2730H) = Gerätenummer MAP1(2739H) => Spur/Sektor MAP2(273BH) ISIS.MAP	Reservierung eines Diskettensektors

1.4 Interne System-Routinen

INTERNE SYSTEM-ROUTINEN



1.5 Tabellen-Zeiger

- FDEAH Zeiger auf Tabelle für Disketten im INTEL-Format
- FDE8H Zeiger auf Tabelle für Disketten im ECMA TC15 (IBM)-Format.
- FDE6H Zeiger auf Tabelle für MB-Kassetten im BS1-Format.

Tabellenaufbau:

- | | |
|-----|------------------------------|
| 0 | Eröffnen für Eingabe |
| 3 | Eröffnen für Ausgabe |
| 6 | Lesen eines Blockes |
| 9 | Schreiben eines Blockes |
| 0CH | Schließen einer Eingabedatei |
| 0FH | Schließen einer Ausgabedatei |

1.6 Systemzeiger

- FDECH nicht benutzt
- FDEEH :LP:
- FDF0H :NF:
- FDF2H :BB:
- FDF4H :CO:
- FDF6H :CI:
- FDF8H :XM:
- FDFAH :RC:
- FDFCH RAM-Speicher-START
(54kB → 2800H)
(48kB → 4000H)
- FDFEH SYSTEM-START

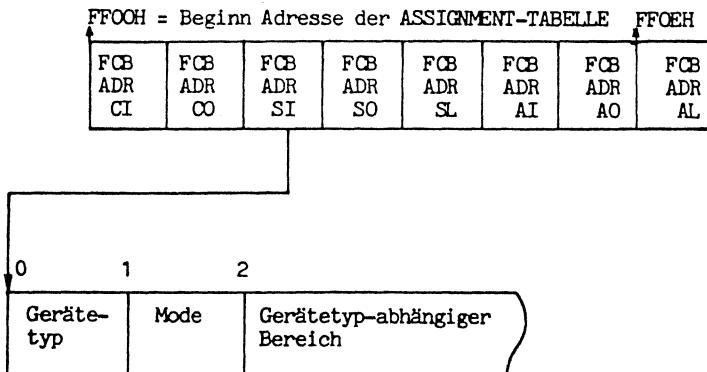
1.7 Logische Einheiten (ASSIGNMENT-Tabelle)

<u>FCB-Zeiger</u>	<u>log. Einheit</u>
FF00H	CI - Tastatur (Bildschirm) Eingabe
FF02H	CO - Bildschirmausgabe
FF04H	SI - Standardeingabe
FF06H	SO - Standardausgabe
FF08H	SL - Standardauflistung
FF0AH	AI - Hilfseingabe
FF0CH	AO - Hilfsausgabe
FF0EH	AL - Hilfsauflistung

1.8 Physikalische Einheiten

:LP:	Sequentielles Ausgabegerät (Drucker)
:BB:	Ausgabeeleerdatei (Byte Bucket File)
:NF:	Eingabeeleerdatei (Null File)
:CO:	Bildschirm
:CI:	Tastatur
:LP:	Drucker
:XM:	Sendung zu einer abgesetzten Station
:RC:	Empfang von einer abgesetzten Station
:CP:	(:CI:/ (:CO:, :LP:))
:Fn:dateiname	Diskette im INTEL-Format
:Mn:dateiname	MB-Kassette im BS1 Format
:Jn:ibmname	Diskette im ECMA TC15 (IBM)-Format (EBCDIC-Code)
:Bn:ibmname	Diskette im ECMA TC15 (IBM)-Format (ASCII-Code)

n = 0 → 3
 dateiname = xxxxxxx.xxx }
 ibmname = xxxxxxxx } x = alphanumerisch

1.9 Datei-Kontrollblock (FCB)Gerätetyp

- 1 - Zeichenorientiertes Gerät im Interrupt-Betrieb
- 2 - Zeichenorientiertes Gerät im Abfrage-Betrieb
- 3 - Echo-Gerät
- 4 - Zeilenorientiertes Gerät
- 5 - Blockorientiertes Gerät, Diskette und MB-Kassette
- 6 - Doppel Gerät

Mode

- 1 - eröffnet für Eingabe
 - 2 - eröffnet für Ausgabe
 - 4 - BINFILE (LF wird nicht unterdrückt)
 - 8 - nicht verwendet
- 10H - LB-Bit (beim nächsten Lesen wird EOF erkannt)
- 20H - Auto-Bit (Datei wurde durch den ASSIGNMENT-Interpreter eröffnet)
- 40H - kein CLOSE
- 80H - EOF-Bit

Aufbau der FCB's in Abhängigkeit des Gerätetyps

Gerätetyp 1 (Zeichenorientiertes Gerät im Interrupt-Betrieb)

0 - Gerätetyp (1)

1 - Mode

2 - Pufferadresse

4 - Zeichenzähler

6 - Puffer

20

Gerätetyp 2 (Zeichenorientiertes Gerät im Abfrage-Betrieb)

0 - Gerätetyp (2)

1 - Mode

2 - LESEROUTINE (z.B. TTI)

4 - SCHREIBEROUTINE (z.B. TTO)

6

Gerätetyp 3 (Echo-Gerät)

0 - Gerätetyp (3)

1 - Mode

2 - EINGABE (Adresse FCB)

4 - AUSGABE (Adresse FCB)

6

Gerätetyp 4 (Zeilenorientiertes Gerät)

0 - Gerätetyp (4)

1 - Mode

2 - Zeichenzähler

4 - Pufferlänge

6 - Pufferadresse

8 - Zeiger auf nächsten FCB

10

Gerätetyp 5 (Blockorientiertes Gerät-Diskette und MB-Kassette)

• Teil 1:

- 0 - Gerätetyp (5)
 - 1 - Mode
 - 2 - Zeichenzähler
 - 4 - Pufferlänge
 - 6 - Pufferadresse
 - 8 - READ
 - 10 - WRITE
 - 12 - OPEN
 - 14 - CLOSE
 - 16 - GERÄTETYP (0 = Diskette)
(1 = MB-Kassette)
- } Routineadressen

• Teil 2 (INTEL und IBM-Diskette):

- 17 - Laufwerknummer
- 18 - DATEINAME
- 27 - DATENTYP:
 - 80H EBCDIC
 - 40H BLOCKLINE
 - 20H
 - 10H
 - 8 } unbenutzt
 - 4 }
- 2 ECMA TC15 (IBM)
- 1 INTEL

- 28 - Sektor/Spur (INTEL: Sektor/Spur des aktuellen Dateikettungsblockes (nur bei SO))
(IBM: Sektor/Spur des nächsten Satzes)

- Teil 3 (INTEL-Diskette)

30 - Zeichenzahl/letzter Block (nur bei Eingabedateien)

31 - relativer Zeiger auf Sektor/Spur
 des nächsten Datenblockes aus Dateikettungs-
 blockpuffer bei SI
 des aktuellen Datenblockes aus Dateikettungs-
 blockpuffer bei SO

32 - Adresse des Dateikettungsblock-Puffers

34

- Teil 3 (IBM-Diskette)

30 - EOD oder EOE - Sektor/Spur
 des letzten geschriebenen Blockes bei SI
 des letzten reservierten Blockes bei SO

32 - Fehlerspuren

34 - HDR-Sektor

35

- Teil 2 (MB-Kassette BS610-FORMAT)

17 - Laufwerknummer

18 - DATEINAME

27 - DATENTYP: 80H EBCDIC
 40H
 20H }
 10H }
 8 } unbenutzt
 4 }
 2 }
 1 }
 BS610-Format

28 - ÜBERLAUF ADR

- Teil 3 (MB-Kassette BS610-FORMAT)

30 - AKTUELLE SPUR

31 - BLOCK-ZÄHLER

33 - AKTUELLES DATEISEGMENT



- Teil 4 (MB-Kassette BS610-Format, Eingabe)

34 - ZEICHENANZAHL/LETZTEN BLOCK

36

- Teil 4 (MB-Kassette BS610-Format, Ausgabe)

34 - ERSTE SPUR

35 - DATEINUMMER

36 - BELEGUNGSVERZEICHNIS: Byte 1 → 15

51

Gerätetyp 6 (Doppel-Gerät)

0 - Gerätetyp (6)

1 - Mode

2 - FCB-Zeiger 1

4 - FCB-Zeiger 2

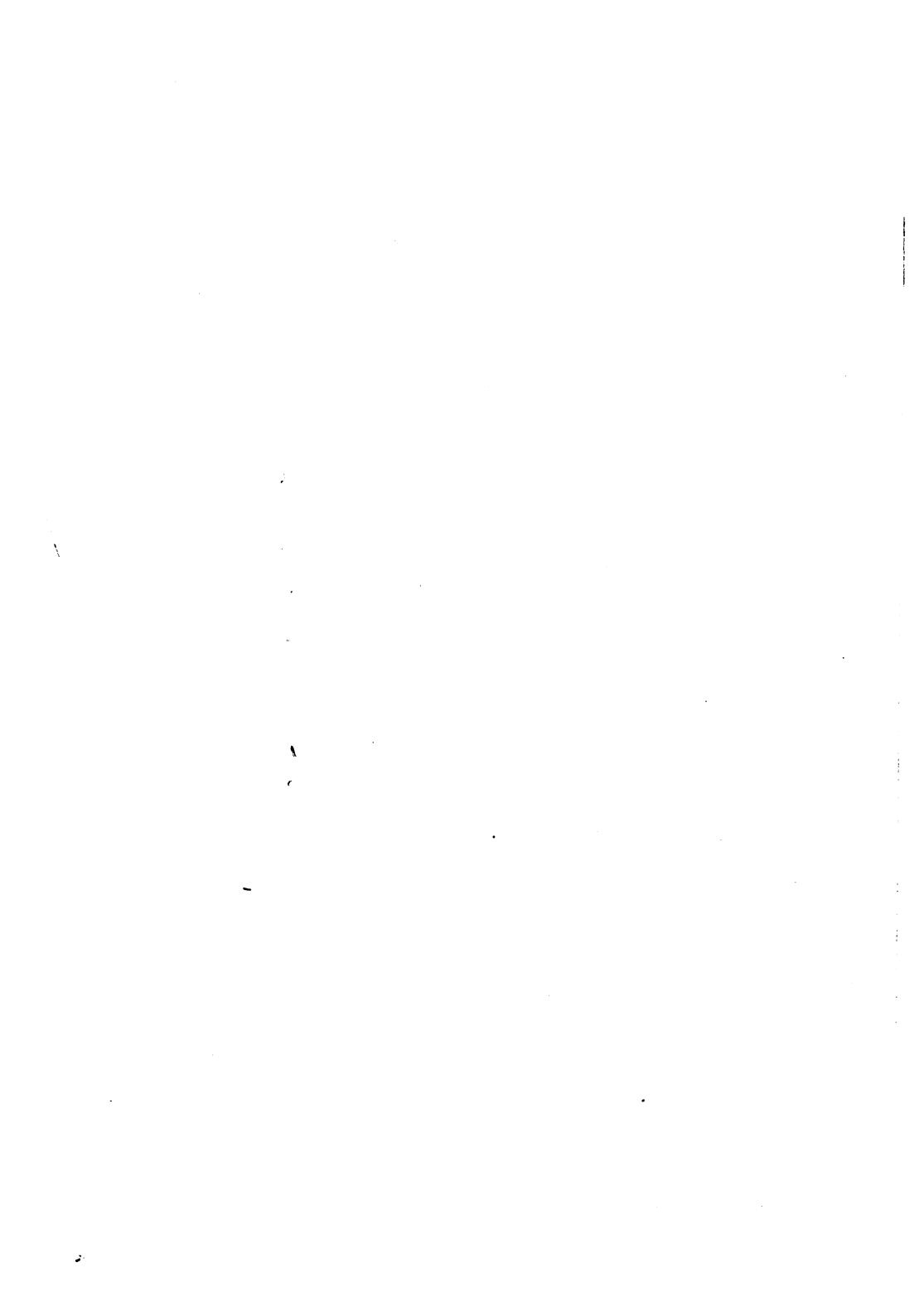
6

1.10 BS1 Fehlermeldungen

Fehlercode (Hex)	Bedeutung
00	Datei oder Bereichsende (EOF, EOE)
01	Fehlender Sektor auf einer Diskette
02	CRC-Fehler auf einer Diskette
03	Zeitfehler während einer READ/WRITE-Datenübertragung
04	Fehlende Adrefsmarke
05	Timeout für Diskettenoperation
06	Zeitfehler bei Leseoperation
07	Busy-Bit Timeout
08	Gerät unklar
09	Name nicht im Inhaltsverzeichnis und auch kein Platz dafür vorhanden
0A	Name nicht im Inhaltsverzeichnis
0B	Programm nicht gefunden
0C	Falscher Eintrag im Inhaltsverzeichnis
0D	Falsche Länge des Inhaltsverzeichnisses beim Laden
0E	Datensatz-Längenfehler beim Laden
0F	Disketten-Belegungsliste nicht auffindbar
10	Ungültiges Disketten-Kommando
11	Falsche Gerätenummer
12	Konvertierungstabelle fehlt
13	Gerätenummer nicht 0, 1, 2, 3
14	nicht verwendet
15	Versuchte Umwandlung eines Zeichens, das nicht Hex-Format hat
16	Versuchte Eingabe aus einer Datei, die keine Eingabedatei ist
17	Leseversuch nach EOF
18	Schreibversuch in eine Datei, die nicht Ausgabedatei ist
19	Nicht zugewiesenes Gerät
1A	Unbekannter Gerätetyp
1B	Kommando oder Dateiname beginnt nicht mit Buchstabe oder Zahl
1C	Kommando oder Dateiname länger als 6 oder Erweiterung länger als 3 Zeichen
1D	Kommando bezog sich nicht auf Diskette in Intel- oder Kassette im BS610-Format
1E	BS610 nicht geladen
1F	Kommando nicht durch CR, S oder SP beendet
20	DELETED-DATA-Satz gefunden

Fehlercode (Hex)	Bedeutung
21	Name einer logischen Einheit unbekannt
22	Schreibversuch für eine Datei, die nicht vom Typ 5 ist
23	Keine Schreib- oder Lese-Routine im Kontrollblock angegeben
24	Leseversuch für eine Datei, die nicht vom Typ 4 oder 5 ist
25	Zeilenüberlauf für Datei vom Typ 4
26	Falscher Parameter
27	Datei bereits eröffnet
28	OPEN-Routine fehlt im Kontrollblock
29	Eine nicht eröffnete Datei soll geschlossen werden
2A	CLOSE-Routine fehlt im Kontrollblock
2B	Pufferverkettung zerstört
2C	Falsches Pufferetikett
2D	Falsche Dateibezeichner
2E	Falsches Zuweisungsformat
2F	Diskette voll
30	nicht verwendet
31	Dateiname für Eingabedatei existiert nicht
32	Ausgabedatei auf der Diskette bereits vorhanden
33	Diskette voll
34	Dateiname länger als 8 Zeichen
35	Spurnummer > 76
36	Sektornummer ≤ 0 oder > 26
37	Pufferadresse ≤ 0
38	Routine fehlt
39	Inhaltsverzeichnisblock auf Kassette nicht 32 Bytes lang
3A	Datenträger-Kennsatz fehlt
3B	Fehler in einem Block des Inhaltsverzeichnisses
3C	Ausgabedatei bereits auf der Kassette vorhanden
3D	Kassette voll
3E	Dateiende einer Eingabedatei überlaufen
3F	Überzählige oder falsch plazierte Abschnittsmarke
40	Bandende-Fehler
41-4F	E/A - Fehler (MB-Kassette)
41	CRC - Fehler (MB-Kassette)
42	Schwerwiegender Kassettenfehler
43	41 + 42
44	Ungültiges Kassettenkommando oder Schreibschutz
45	41 + 44

Fehlercode (Hex)	Bedeutung
46	Leseversuch auf einem gelöschten Band
47	41 + 42 + 44
48	Abschnittsmarke gefunden
49	41 + 48
4A	42 + 48
4B	41 + 42 + 48
4C	44 + 48
4D	41 + 44 + 48
4E	42 + 44 + 48
4F	41 + 42 + 44 + 48
50	Falsche Gerätenummer (Kassette)
51	Falsche Spurnummer "
52	Falsche Blocklänge "
53	Spurwechselfehler "
54	Gerät unklar "
55	Falsche Pufferadresse "
56	Fremdes Kassettenformat
57	Zusätzliche Blöcke in einer Kassettendatei
58	Folgeetikett fehlt
59	Falscher Blocktyp im Kennsatz (Kassette)
5A	Falscher Name im Kennsatz (Kassette)
5B	Falsche Blocklänge während des Ladens (Kassette)
5C	Blockende (nur bei B- und J-Format)
5D	Unzulässiger Direktzugriff



INHALT**REGISTER 2****DIENSTPROGRAMME**

KAPITEL 1 ALLGEMEINE DIENSTPROGRAMME

Seite

1. Aufrufe

2-1-1

KAPITEL 2 EDITIERPROGRAMME

1. SEDIT

2-2-1

2. EDIT

2-2-2

3. TXT

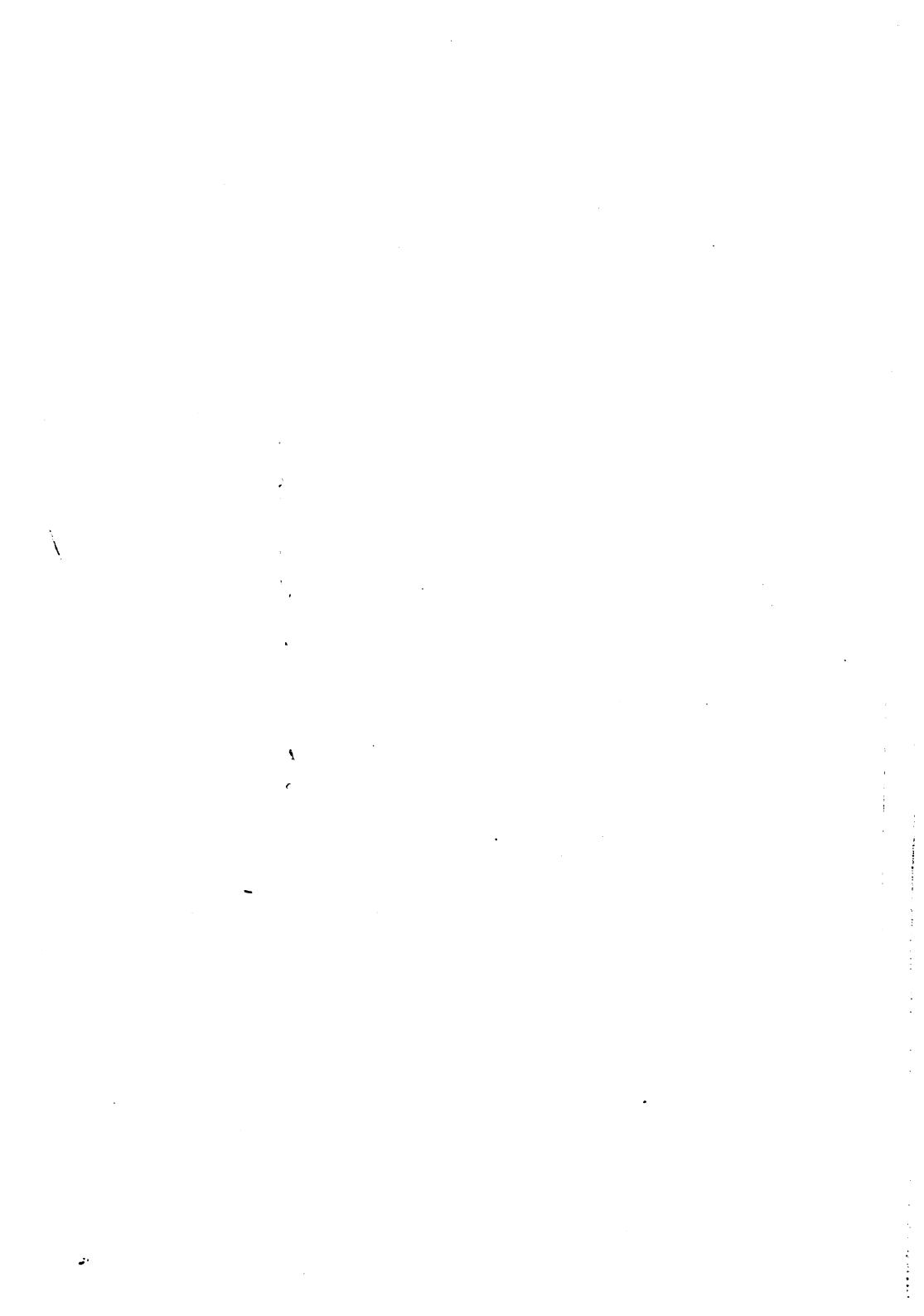
2-2-9

Name	Aufruf	Bedeutung
ALIAS	ALIAS\$:lw:name,aliasname Bsp: ALIAS\$:F1:DATEI,NAME	Definition eines zusätzlichen Namens
ALLOC	ALLOC\$:Jn:ibmname Bsp: ALLOC\$:J1:DATA1	Platzreservierung für eine Datei auf einer IBM-formatierten Floppy-Disk
ANALYZ	ANALYZ [SO=:lw:name] \$n:lw: Bsp: ANALYZ SO=:F2:FEHLER\$4:F1: ANALYZ\$:F1:	Prüfen einer Floppy-Disk auf Fehler
ASSIGN	ASSIGN zuweisungen Bsp: ASSIGN SO=:F2:AUSGAB	Zuweisung einer Datei oder Gerät zu logischen Einheiten für mehr als ein Kommando
ATTRIB	ATTRIB\$:lw:name,attributliste Bsp: ATTRIB\$:F1:ABC,+W,-S,-I	Änderung von Dateiattributen (S,W,I)
COPY	COPY SI =:lw:eingabe [,SO=:lw:ausgabe] \$ info [,SO=geraet Bsp: COPY SI=EINGAB,SO=:F1:AUSGAB COPY SI=EINGAB,SO=:LP:\$PL72	Kopieren einer einzelnen Datei
DCONVA	DCONVA	Konvertierung des Inhalts einer Floppy Disk von EBCDIC in ASCII
DCOPY	DCOPY	Kopieren einer ganzen Floppy-Disk (1:1)
DELETE	DELETE\$:lw:name,... Bsp: DELETE\$ABC,:F1:DAT	Löschen einer oder mehrerer Dateien
DGEN	DGEN	Dateiweises kopieren einer Floppy-Disk auf eine andere

Name	Aufruf	Bedeutung
DIR	DIR [SO=:lw:name] \${[info]} :lw: SO=geraet Bsp: DIR SO=:LP:\$!F2: DIR\$V:F1: DIR\$	Auflistung des Inhaltsverzeichnisses info: I - alle Dateien V - alle Dateien mit Versionsnummernausgabe ohne - nur Dateien ohne I-Attr
DIRPAC	DIRPAC\$:Fn: Bsp: DIRPAC\$:F1:	Packen des Inhaltsverzeichnisses einer Floppy-Disk
DROP	DROP SI=:lw:eingabe\$:lw: Bsp: DROP SI=:F2:FEHLER\$:F1: DROP\$:F1: (d.h. SI=/CI)	Kennzeichnung von Sektoren als "belegt"
EXEC	EXEC CI=:lw:eingabe Bsp: EXEC CI=:F1:KOMMAN.DO	Abarbeitung einer gespeicherten Kommandofolge. Bei Fehlermeldungen wird die Kommandofolge unterbrochen und ins BSI verzweigt.
FORMAT	FORMAT\$:lw:datentraeger [,S] ,L Bsp: FORMAT\$:F1:DISK FORMAT\$:F2:SYSTEM,S FORMAT\$:F2:SYSTEM.NEU,L	Einrichten der Disketten im INTEL-Format bzw. der MB-Kassetten im BS610-Format S - Kopieren aller Dateien mit S-Attribut L - Ändern des Datentraegernamens
INIT	INIT\$:lw: Bsp: INIT\$:F1: (Sektorfolge 1 14 2 15...) INIT\$:J1: (Sektorfolge 1 2 3 4 5...)	Initialisierung einer Floppy-Disk

Name	Aufruf	Bedeutung
RELEAS	RELEAS\$log.einheiten Bsp: RELEAS\$1,SO	Freigabe einer vorher mit ASSIGN zugewiesenen Datei oder Gerät
REMAP	REMAP\$:Fn: Bsp: REMAP\$:F1:	Rekonstruktion des Belegungsverzeichnisses einer Floppy-Disk
RENAME	RENAME\$:lw:name,neuname Bsp: RENAME\$:F1:ABC,DEF	Umbenennung einer Datei
RESCUE	RESCUE\$:lw:name Bsp: RESCUE\$:F3:DAT	Retten einer gelöschten Datei
SORT	<p>SORT [SO=:lw:programname] \$param SORT\$SAVE :lw:programname param</p> <p>param: I eingabe II TO ausgabe III [USING] :Fn: IV [PREPROC eingabeproz] V [POSTPROC ausgabeproz] VI SAME [DISC] I O W DISK VII KEY position laenge A (aufsteigend) D (absteigend) VIII REC { BIN } satzlaenge { ASCII }</p>	Sortieren von sequentiellen Dateien nach einem Sortierbegriff (1-10 Teile, max. 255 Bytes groß)
	<p>Bsp: SORT\$DAT1 TO DAT2 USING :F1: KEY 4 20 D ASCII 81 SORT SO=LLLL\$DAT1 TO DAT2 :F0: SAME I O W KEY 1 5 BIN 128 SORT\$SAVE :F2:PROG :F1:DAT1 TO :F1:DAT2 :F2: SAME I O KEY 5 15 BIN 128</p>	

Name	Aufruf	Bedeutung
SYS	SYS\$param	Setzen des Systemschalters (272CH) param: S - Schreibgeschützte Dateien (mit Attribut W oder S) können gelöscht oder umbenannt werden C - Bei Fehlermeldungen wird die EXEC Kommandofolge fortgesetzt
	SYS\$	Rücksetzen des Systemschalters





1. SEDIT (Zeileneditor)

Aufruf

```
SEDIT {SI=:lw:eingabe} [SO=lw:ausgabe]
      {SI=:NF:} [SO=geraet]
```

Bsp: SEDIT SI=:F1:EINGAB,SO=:LP:
 SEDIT SI=:F1:EINGAB,SO=:F2:AUSGA
 SEDIT SI=:NF:,SO=:F3:AUSGAB.NEU

Kommandos

Syntax	Bedeutung
B	Rücksetzen der Zeiger im Ein- und Ausgabepuffer auf das jeweils erste Zeichen (Kombination von BI und BO)
BI	Rücksetzen des Zeigers im Eingabepuffer auf das erste Zeichen
BO	Rücksetzen des Zeigers im Ausgabepuffer auf das erste Zeichen
C	Kopieren der restlichen Information im Eingabepuffer in den Ausgabepuffer, Rückschreiben nach SO, Lesen der nächsten Eingabezeile, Löschen des Ausgabepuffers
Cn	($0 < n \leq 99$) wie C, jedoch mit ($n-1$)facher Wiederholung
C+n	($0 < n \leq 99$) kopiert die nächsten n-Zeichen vom Ein- in den Ausgabepuffer
C"text"	Kopiert bis (ausschließlich) zum nächsten Erscheinen des spezifischen Textes im Eingabepuffer
DI	Anzeige der Zeigerposition und des Inhalts des Eingabepuffers
DO	Anzeige des Ausgabepuffer-Inhalts
H	Ende des Editierens, der Rest wird <u>nicht</u> nach SO übertragen
I"text"	Einfügen von "text" in den Ausgabepuffer
I	Einfügen einer Anzahl von Zeilen in die Ausgabedatei
S	Überspringen des Restes der laufenden Zeile
Sn	($0 < n \leq 99$) Überspringen von n Zeilen
S+n	($0 < n \leq 99$) Überspringen der nächsten n Zeichen im Eingabepuffer
S"text"	Überspringen bis (ausschließlich) zum nächsten Erscheinen von "text" im Eingabepuffer
X"text"	Einfügen von "text" in den Ausgabepuffer und Überspringen ebensovieler Zeichen im Eingabepuffer

2. EDIT (Editor)

Aufruf

EDIT [SI=:lw:eingabe] [,SO=:lw:ausgabe][,SL=:lw:listet][,AI=:lw:zusatzeingbe]
[,SO=geraet][,SL=geraet]

Bsp: EDIT SI=:F1:EIN,SO=:LP:,AI=:F3:EING.ZUS
EDIT SO=:F2:AUSGAB.NEU

Textadressierung

- . Zeilennummer (50)
- . Zeichenfolge ("abc")
- . Aktuelle Zeile (g)
- . Erste Zeile (:)
- . Letzte Zeile (.)
- . Textintervall (100,200)
(:,"zz")
(:..)

Funktionswiederholungen

n {<funktion>} 1 ≤ n ≤ 255
{<funktionsfolge>}

Bsp: 5 < R;W;R;:>
150 < CL>

Retten der Ausgabedatei bei Absturz

.Q8403 CR

Befehle

Syntax	Beispiel	Bedeutung
AA"zueingabe"	AA"DATEI"	Zuweisung der zusätzlichen Eingabedatei
AI ln1 [*za] AI"zeichenfolge" [*za]	AI3*20 AI"ADRESSE"	Lesen von n Zeilen aus zusätzlicher Eingabedatei (za=Zeilenadresse). Der Puffer darf nicht leer sein.
AS [n] AS"zeichenfolge"	ASS AS AS"NAME"	Überlesen von Zeilen der zusätzlichen Eingabedatei
CUzeilennummer CU"zeichenfolge"	CU80 CU"ORT"	Kopieren in die Ausgabedatei bis zu einer angegebenen Adresse.
E[XIT]		Normaler Abschluß
K[ILL]		Löschen des Textpuffers
OA"dateiname"	OA"DAT1"	Öffnen der Ausgabedatei
OC		Schließen der Ausgabedatei
Q[UIT]		Rückkehr ins BS610
R[EAD] [n]	R5 READ	Lesen von Zeilen der Eingabedatei in den Textpuffer
SUzeilennummer SU"zeichenfolge"	SU100 SU"DATUM"	Einlesen ab einer angegebenen Adresse

Syntax	Beispiel	Bedeutung
Tn1,n2,... ni	T1,5,10	Setzen Tab-Stops ($i \leq 15$)
TA (1,9,16,26)		Setzen von Assembler-Tab-Positionen
TC (1,8,12,16...)		Setzen von COBOL-Tab-Positionen
TD		Anzeige der aktuellen Tab-Positionen
TF (1,6,7,10,13,16,...)	,	Setzen von FORTRAN-Tab-Positionen
W [RITE] [*] [n]	W*10 W	Schreiben des Textpuffers in Ausgabedatei

Syntax				
Adressbezug	Funktionsname	Funktionsparameter	Beispiel	Bedeutung
{leer zeilenadresse}	A [L]	{leer dezimalzahl}	100AL5	Zeile(n) anfügen (Zeilenanfügbungsmodus)
leer	AT	"zeichenfolge"	AT"NAME"	Zeichenfolge anfügen
{leer zeilenadresse}	C [L]	{leer textintervall}	100CL"TITLE" C	Zeile(n) kopieren
leer	D [C]	{leer dezimalkonstante "zeichenfolge"}	DC5 DC"ab" D	Zeichen löschen
{textintervall}	DL	leer	100,150DL	Zeile(n) löschen
leer	FB	"zeichenfolge"	FB"NACH"	Zeichenfolge rückwärts suchen
{leer textintervall}	F [F]	"zeichenfolge"	100,150 FF "VOR" F"VOR"	Zeichenfolge vorwärts suchen
{leer zeilenadresse}	I [L]	{leer dezimalzahl}	100IL5 I	Zeile(n) einfügen (Einfügbungsmodus)
leer	IT	"zeichenfolge"	IT"NAME"	Zeichenfolge einfügen
leer	L [C]	{leer dezimalkonstante}	L LC128	Zeilen verketten

		Syntax	Beispiel	Bedeutung
Adreßlegung	Funktionsname	Funktionsparameter		
leer	LS	leer	LS	Zeilen trennen
leer	LSL	leer	LSL	Zeilen trennen und links ausrichten (bzw. Ausgabe des gesamten Satzes ≤ 128 Bytes auf 2 getrennten Zeilen)
{ leer zeilenadresse }	M[L]	{ leer textintervall }	100ML20,30 M	Textzeilen verschieben
leer	PL	leer	PL	Verschieben des Pufferzeigers auf Zeilenende
leer	P[N]	[+] dezimalzahl	PN3 PN-5	Verschieben des Pufferzeigers um die angegebene Zeichenanzahl
leer	PS	leer	PS	Verschieben des Pufferzeigers auf Zeilenanfang
{ leer textintervall }	S[T]	"text1""text2"	ST"xx""aa" 100S"bb""cc"	Zeichenfolge 1 durch 2 ersetzen
{ leer textintervall }	SW	"wort1""wort2"	SW"a""A" :,sw"b""be"	Wort 1 durch 2 ersetzen

Kontrollfunktionstaste

CURSOR LINKS



CURSOR RECHTS



CURSOR RAUF



CURSOR RUNTER



HOME - TASTE



WAGENRÜCKLAUF (CR)



ROLL UP



ROLL DOWN



(CONTROL + D)

Vorwärts-Tab



(CONTROL + G)

Rückwärts-Tab



CHAR

Lösche Zeichen



CHAR

Füge Blank ein



DEL

Zeile löschen

Fehlermeldungen

Fehler - meldung	Bedeutung
E00:	Interner Fehler
E01:	Überlauf Textpuffer
E02:	Freelist-Bereich zu klein
E03:	Zu viele Funktionswiederholungen
E04:	Unerlaubtes oder falsch geschriebenes Kommando
E05:	Unzulässiger Wert eines Adressbezugs
E06:	Fehlerhafte Syntax eines Adressbezugs
E07:	Fehlende Zahl nach + oder - Zeichen in einem Adressbezug
E08:	Unerlaubter Anfang einer Zeichenfolge
E09:	Zeichenfolge wurde nicht mit " abgeschlossen
E10:	Angegebene Zeichenfolge wurde nicht gefunden
E11:	Angegebene Zeilennummer wurde nicht gefunden
E12:	Keine Ausgabedatei zugewiesen
E13:	Systemfehler Code xxH
E14:	Keine Eingabedatei zugewiesen
E15:	Unzulässiger Tab-Wert
E16:	Angabe zu vieler Tabs
E17:	Keine kopierbare Zeile vorhanden
E18:	Keine Einfügung durchgeführt
E19:	Unzulässiger Wert der Dezimalkonstante
E20:	Unzulässige Syntax des Befehls/Funktion
E21:	Zeichenfolge nicht gefunden
E22:	Kommando durch Bediener unterbrochen
E23:	Befehl/Funktion noch nicht implementiert
E24:	Textpuffer leer
E25:	Kein löschares Zeichen vorhanden
E26:	Versuch "Lesen" nach "Ende der Datei"
E27:	Senkrechter Strich in String 2
E28:	Kommandos müssen durch ; getrennt werden
E29:	Kein Platz mehr auf der Floppy Disk
E30:	Ausgabedatei existiert bereits auf Floppy Disk
E31:	Nicht erlaubter Dateiname
E32:	Ausgabedatei wurde schon zugewiesen
E33:	Eingabedatei wurde schon zugewiesen
E34:	Datei kann wegen Mangel an Hauptspeicherplatz nicht eröffnet werden.
	Fehler kann durch Schließen überflüssiger Dateien behoben werden.
E35:	Zu viele Dateien eröffnet. Bei Aufruf des Editors dürfen nur SI, SO, SL und AI zugewiesen werden.
E36:	Unzulässiges Ausgabegerät. :CO: ist als Ausgabegerät unzulässig.
E37:	Leere Zeichenfolge wurde gefunden
E38:	Austausch von Wert und Zeichenfolge überschreitet die Zeilenlänge von 80 Zeichen
E39:	Angegebene Zeilenlänge zu klein

3. TXT (Text)

Aufruf

```
TXT { SI=:lw:txtdat
      { AI=:lw:jobdat
        { ,SO=:lw:ausgabe
          ,SO:geraet
          ,SL=:lw:ausgabe
          ,SL:geraet
        }
      }
    }
```

Bsp: TXT SI=:F1:TEXT1,SL=:LP:
 TXT AI=JOB,SO=:F3:TEXT.ALL
 Datei JOB: TEXT1
 TEXT2

Steuerzeichen

Typ 1

- gültig für die ganze Datei oder bis Neuversorgung
- 1 Steuerzeichen / Zeile
- ":" erstes Zeichen dieser Zeile

Steuerzeichen	Parameter	Bedeutung/Beispiel
FL	1) Erste Textzeile der neuen Seite(Standard:Zeile 5)	Erste Zeile :FL8
HT	1) Zeile 2) Position innerhalb der Zeile 3) Gewünschter Text	Seitenüberschrift :HT3,20,TITEL
LL	1) Letzte Druckzeile der Seite (Standard: Zeile 68)	Letzte Zeile :LL30
LM	1) Erste Druckposition einer Zeile	Linker Rand :LM10
PL	1) Anzahl Zeilen/Seite	Papierlänge :PL48
PN	1) Nummer der Zeile 2) Position innerhalb der Zeile 3) Format: 1: Ohne Nullenunterdrückung 2: Seitennummer innerhalb von Strichen (z.B. -5-) 4) Beginn der Seitennummerierung (wahlweise)	Seitennummerierung :PN3,40,2
RM	1) Letzte Druckposition der Zeile	Rechter Rand :RM50
SL	1) Anzahl Leerzeilen vor jedem Absatz	Leerzeilen :SL3
TS	1) Anzahl der Tabulator-Stops	Tabulator-Stop :TS8,20

Typ 2

- gültig nur für den folgenden Paragraphen
(Paragraphen-Kennung: Ein Paragraph beginnt und endet mit einer Zeile, deren erstes und einziges Zeichen ";" ist.)

- 1 Steuerzeichen/Zeile
- ";" erstes Zeichen dieser Zeile
- RM, RA, NA, TS, TP, SL wie bei Type 1

Steuerzeichen	Parameter	Bedeutung/Beispiel
NP	Keine	Neue Seite .NP
ST	1) Nummer der Zeile	Sprung auf eine bestimmte Zeile .ST20
UF	Keine	Unformatiert .UF

Typ 3

- direkt in den Text eingefügt
- ! Neue Zeile
- ⇒ Tabulator Sprung
- Trennungszeichen
- !, >, -, gefolgt von / = keine Kontrollzeichen



KAPITEL 1	ASSEMBLER	Seite
1. Hilfsprogramme zur Assemblerprogrammierung		3-1-1
2. SAB 8080		3-1-4
2.1 Befehlsliste		.
2.2 Befehlsliste in Hex-Folge		3-1-20
2.3 Aufbau des Status Bytes		3-1-22
3. ASSMBLER Standardroutinen		3-1-23
4. Makroprozessor - Stage 2		3-1-27
5. Fehlermeldungen		3-1-28

KAPITEL 2 COBOL A, COBOL B

1. Sprachformat COBOL A	3-2-1
2. Sprachformat COBOL B	3-2-4
3. Verarbeitung von RFM-Dateien	3-2-12
4. Segmentierung	3-2-13
5. Interne DEBUG-Kommandos	3-2-14
6. Übersetzung	3-2-15
7. Laufzeitsystem	3-2-16
8. Schnittstelle Sonderaufrufe	3-2-17
9. Schnittstelle COBOL - ASSEMBLER	3-2-18
10. Schnittstelle COBOL - COBOL	3-2-20
11. Fehlermeldungen (Übersetzer COBOL A)	3-2-21
12. Fehlermeldungen (Übersetzer COBOL B)	3-2-23
13. Meldungen Laufzeitsystem COBOL A, B	3-2-26

KAPITEL 3 COMMERCIAL BASIC

1. Anweisungen	3-3-1
2. Standardfunktionen	3-3-2
3. Befehle	3-3-5
4. Schnittstelle C-BASIC - ASSEMBLER	3-3-6
5. Fehlermeldungen	3-3-9

KAPITEL 4 SCIENTIFIC BASIC

1. Darstellung von Konstanten und Variablen	3-4-1
2. Operatoren	3-4-2
3. Kommandos und Anweisungen	3-4-3
4. Funktionen	3-4-9
5. Schnittstelle S BASIC - ASSEMBLER	3-4-12
6. Fehlermeldungen	3-4-15/16

ASSEMBLER

HILFSPROGRAMME

1. Hilfsprogramme zur ASSEMBLER-Programmierung

Name	Aufruf	Bedeutung
ASM	ASM AO=;lw:quellausgabe Bsp: ASM AO:F1:PROG.S ASM AO=(PROG.S,;LP;) (Ausgabe des Quellprogrammes auf Drucker und Diskette)	Erfassung und Test von Assemblerprogrammen <u>Kommandos:</u> .GadrH - Start eines Programmes .I - Rücksetzen der Symboltabelle .Ksymbol - Löschen definierter Symbole .M - Aufruf des Monitors .S - Auflistung aller definierten Symbolen .U - Auflistung der undefinierten Symbole .X - Rücksprung ins BS610 / - Ausgabe der Speicherinhalte

Name	Aufruf	Bedeutung
RASM	<p>RASM SI=:lw:eingabe, SO=:lw:ausgabe [,SL=liste] \$[info] ,AL=fehl liste]</p> <p>info: PW PL</p> <p>Bsp: RASM SI=PROG.SRC, SO=PROG.HEX, SL=:LP:\$PW70PL72</p>	Verschiebbarer Assembler
LINK	<p>LINK SO=:lw:abs.progr [,SL=liste, AI=linkinfo]</p> <p>Bsp: LINK SO=PROG, SL=LIST, AI=LINK.INF LINK SO=PROG *C = 8000 (Ladeadresse des Codesegmentes) *D = 7000 (Ladeadresse des Datensegmentes) *S = 8000 (Startadresse des Programmes) PROG.R / Ende der Parametereingabe *Q Rücksprung ins BS610 (Abbruch der Operation)</p>	Binden durch RASM erzeugter Module und Erstellen eines ablauffähigen Programmes
XREF	<p>XREF SI=:lw:eingabe [,SL=ausgabe] \$[info] ,SO=:lw:ausgabe]</p> <p>info: PW PL</p> <p>Bsp: XREF SI=EINGAB, SL=:LP:\$PW70PL72 XREF SI=EINGAB, SO=AUSGAB</p>	Erzeugung einer Querverweisliste für ein Quellprogramm

Name	Aufruf	Bedeutung
DEBUG	DEBUG\$:lw:progr [zuweisungen] Bsp: DEBUG%COPY SI=EINGAB	Laden eines Programmes Durchführung der Dateizuweisungen Übergabe der Kontrolle an den Monitor
MEMDMP	MEMDMP SO=:lw:ausgabe\$xxxx,yyyy,zzzz xxxx = Anfangsadresse (hex) des auszugebenden Speicherbereiches yyyy = Endadresse (hex) des auszugebenden Speicherbereiches zzzz = spätere Startadresse (hex) des Programmes Bsp: MEMDMP SO=AUSGAB\$4000,8000,4000	Ausgabe eines RAM-Bereiches als Binärdatei

2. SAB 80802.1 BefehlslisteNotation

ddd = Zielregister ("Destination-Register")

sss = Senderegister ("Source-Register")

rr = Registerpaar

<u>ddd oder sss</u>	<u>Register</u>
---------------------	-----------------

000	B
001	C
010	D
011	E
100	H
101	L
110	M (Speicherzugriff)
111	A (Akkumulator)

<u>rr</u>	<u>Registerpaar</u>
-----------	---------------------

00	(B,C)
01	(D,E)
10	(H,L)
11	PSW (Register A und Flag-Register)
11	SP (Stackpointer)

Mnemonic	Binärcode	Beeinfl. Zustands- Bits	Anzahl der Bytes	Takt- zyklen	Englische Befehls- beschreibung	Funktion des Befehls
----------	-----------	-------------------------------	---------------------	-----------------	---------------------------------------	----------------------

TRANSFERBEFEHLE (Datenübertragungsfehler)

a) Register → Register

MOV r ₁ ,r ₂	01dddsss	-----	1	5	Move register to register	r ₁ ,r ₂ = A,B,C,D,E,H oder L: Lade Register r ₁ mit dem Inhalt von Register r ₂
XCHG	11101011	-----	1	4	Exchange D & E, H & L	Vertausche Inhalt der Registerpaare (D, E) und (H, L)
XTHL	11100011	-----	1	18	Exchange top of stack H & L	Vertausche Inhalt des Registerpaars (H, L) und den Inhalt des Wortes, das durch den Stackpointer adressiert ist
SPHL	11111001	-----	1	5	H & L to stack-pointer	Lade Stackpointer mit dem Inhalt des Registerpaars (H, L)

b) Speicher, Peripherie → Register

MOV r ₁ ,M	01ddd110	-----	1	7	Move memory to register	r ₁ = A,B,C,D,E,H oder L: Lade Register r ₁ mit dem Inhalt des Speicherbytes, das durch den Inhalt des Registerpaars (H, L) adressiert ist
LDA adr	00111010	-----	3	13	Load accumulator direct	Akkumulator laden mit Inhalt der Adresse adr
LDAX rp	00rr1010	-----	1	7	Load accumulator indirect	rp = B,D: Akkumulator laden mit Inhalt des Speicherplatzes, der durch den Inhalt des Registerpaars rp adressiert ist

Mnemonic	Binärcode	L. infl. Zustands- Bits	Anzahl der Bytes	Takt- zyklen	Englische Befehls- beschreibung	Funktion des Befehls
LHLD adr	00101010	- - - - -	3	16	Load H&L direct	Lade Registerpaar (H, L) mit dem Inhalt der Adressen adr und (adr + 1)
POP rp PSW	11rr0001 11110001	- - - - - Z,S,P,CY,AC	1	10	Pop register pair rp off stack	rp = B,D,H,PSW: Registerpaar rp wird mit dem Wort geladen, das durch den Stackpointer adressiert ist
IN nr	11011011	- - - - -	2	10	Input	Akkumulator wird mit dem Inhalt des Eingabekanals Nummer nr (nr ≤ 255) geladen

c) Konstante → Registerpaar

LXI rp,adr	00rr0001	- - - - -	3	10	Load register pair immediate	rp = B,D,H,SP: Lade Registerpaar rp mit Wert adr
------------	----------	-----------	---	----	------------------------------	--

d) Register → Speicher, Peripherie

MOV M,r ₁	01110sss	- - - - -	1	7	Move register to memory	r ₁ =A,B,C,D,E,H oder L: Inhalt von Register r ₁ auf den Speicherplatz abspeichern, der durch den Inhalt des Registerpaars (H, L) adressiert ist
STA adr	00110010	- - - - -	3	13	Store accumulator direct	Akkumulator-Inhalt unter Adresse adr abspeichern
STAX rp	00rr0010	- - - - -	1	7	Store accumulator indirect	rp = B,D: Akkumulator in dem Byte abspeichern, das durch den Inhalt des Registerpaars rp adressiert ist

Mnemonic	Binärkode	Beeinfl. Zustands- Bits	Anzahl der Bytes	Takt- zyklen	Englische Befehls- beschreibung	Funktion des Befehls
SHLD adr	00100010	- - - - -	3	16	Store H&L direct	Registerpaar (H, L) unter Adresse adr und (adr + 1) abspeichern
PUSH rp	11rr0101	- - - - -	1	11	Push register pair rp on stack	rp = B,D,H,PSW: Inhalt des Registerpaars rp wird in das Wort übertragen, das durch den Stackpointer adressiert ist
OUT nr	11010011	- - - - -	2	10	Output	Akkumulator-Inhalt wird auf Ausgabe-kanal (Nummer nr < 255) ausgegeben

e) Konstante → Register, Speicher

MVI M, konst.	00110110	- - - - -	2	10	Move to memory immediate	Bringe den Wert der Konstanten (konst. < 255) auf den Speicherplatz, der durch den Inhalt des Registerpaars (H, L) adressiert ist
MVI r ₁ , konst.	00ddd110	- - - - -	2	7	Move to register	r ₁ =A,B,C,D,E,H oder L: Lade Register r ₁ mit Konstante(konst. < 255)

Mnemonic	Binärcode	Be fl. Zustands- Bits	Anzahl der Bytes	Takt- zyklen	Englische Befehls- beschreibung	Funktion ... Befehls
ARITHMETISCHE BEFEHLE						
INR r ₁	00ddd100	Z,S,P,-AC	1	5	Increment register	r ₁ =A,B,C,D,E,H oder L: Zum Inhalt des Registers r ₁ wird 1 addiert
INR M	00110100	Z,S,P,-AC	1	10	Increment memory	Zum Inhalt des durch Registerpaar (H, L) adressierten Bytes wird 1 addiert
DCR r ₁	00ddd101	Z,S,P,-AC	1	5	Decrement register	r ₁ =A,B,C,D,E,H oder L: Vom Inhalt des Registers r ₁ wird 1 subtrahiert
DCR M	00110101	Z,S,P,-AC	1	10	Decrement memory	Vom Inhalt des durch Registerpaar (H, L) adressierten Bytes wird 1 subtrahiert
INX rp	00rr0011	- - - - AC	1	5	Increment register pair	rp=B,D,H,SP: Der Inhalt des Registerpaars rp wird um 1 erhöht
DCX rp	00rr1011	- - - - -	1	5	Decrement register pair	rp=B,D,H,SP: Der Inhalt des Registerpaars rp wird um 1 erniedrigt
ADD r ₁	10000sss	Z,S,P,CY,AC	1	4	Add register to accumulator	r ₁ =A,B,C,D,E,H oder L: Inhalt von Register r ₁ wird zum Inhalt des Akkumulators
ADD M	10000110	Z,S,P,CY,AC	1	7	Add memory to accumulator	Der Inhalt des Speicherbytes, das durch den Inhalt des Registerpaars (H, L) adressiert ist, wird zum Inhalt des Akkumulators addiert

Mnemonic	Binärkode	Beeinfl. Zustands- Bits	Anzahl der Bytes		Englische Befehls- beschreibung	Funktion des Befehls
			Takt- zyklen			
ADC r_1	10001sss	Z,S,P,CA,AC	1	4	Add register to accumulator with carry	$r_1 = A, B, C, D, E, H$ oder L: Inhalt des Registers r_1 und Inhalt des Carrybits werden zum Inhalt des Akkumulators addiert
ADC M	10001110	Z,S,P,CA,AC	1	7	Add memory to accumulator with carry	Inhalt des Speicherbytes, das durch den Inhalt des Registerpaars (H, L) adressiert ist und der Inhalt des Carrybits werden zum Inhalt des Akkumulators addiert
DAD rp	00rr1001	- - - CY -	1	10	Add register pair to H and L	rp = B,D,H,SP: Inhalt des Registerpaars rp, und der Inhalt des Registerpaars (H,L) werden addiert. Ergebnis in (H,L)
SUB r_1	10010sss	Z,S,P,CA,AC	1	4	Subtract register from accumulator	$r_1 = A, B, C, D, E, H$ oder L: Inhalt des Registers r_1 wird vom Akkumulator-Inhalt subtrahiert
SUB M	10010110	Z,S,P,CA,AC	1	7	Subtract memory from accumulator	Inhalt des Speicherbytes, das durch das Registerpaar (H, L) adressiert ist, wird vom Akkumulator subtrahiert
SBB r_1	10011sss	Z,S,P,CY,AC	1	4	Subtract register from accumulator with borrow	$r_1 = A, B, C, D, E, H$ oder L: Inhalt von Register r_1 und Inhalt des Carrybit werden vom Akkumulator-Inhalt subtrahiert

Mnemonic	Binärkode	Beeinf. Zustands- Bits	Anzahl der		Englische Befehls- beschreibung	Funktion des Befehls
			Bytes	Takt- zyklen		
SBB M	10011110	Z,S,P,CY,AC	1	7	Subtract memory from accumulator with borrow	Inhalt des Speicherbytes, das durch das Registerpaar (H, L) adressiert ist und Inhalt des Carrybits werden vom Akkumulator-Inhalt subtrahiert
ADI konst.	11000110	Z,S,P,CY,AC	2	7	Add immediate to accumulator	Konstante (konst. < 255) wird zum Inhalt des Akkumulators addiert
ACI konst.	11001110	Z,S,P,CY,AC	2	7	Add immediate to accumulator with carry	Zum Akkumulator-Inhalt werden konst. < 255 und Carrybit addiert
SUI konst.	11010110	Z,S,P,CY,AC	2	7	Subtract immediate from accumulator	konst. < 255 wird vom Akkumulator- Inhalt subtrahiert
SBI konst.	11011110	Z,S,P,CY,AC	2	7	Subtract immediate from accumulator with borrow	konst. < 255 und das Carrybit werden vom Akkumulator-Inhalt subtrahiert
DAA	00100111	Z,S,P,CY,AC	1	4	Decimal adjust accumulator	Akkumulator-Inhalt wird in eine 2 stellige Zahl umgewandelt

Mnemonic	Binärcode	Beeinfl. Zustands- Bits	Anzahl der Bytes		Englische Befehls- beschreibung	Funktion des Befehls
			Takt- zyklen			
LOGISCHE BEFEHLE						
CMA	00101111	- - - - -	1	4	Complement accumulator	Akkumulator-Inhalt wird negiert
ANA r ₁	10100sss	Z,S,P,CY,AC	1	4	And register with accumulator	r ₁ =A,B,C,D,E,H oder L: Akkumulator- Inhalt und der Inhalt des Re- gisters r ₁ werden UND-verknüpft
ANA M	10100110	Z,S,P,CY,AC	1	7	And memory with accumulator	Der Inhalt des durch Registerpaar (H, L) adressierten Bytes wird mit dem Akkumulator-Inhalt UND verknüpft
ANI konst.	11100110	Z,S,P,CY,AC	2	7	And immediate with accumulator	Akkumulator-Inhalt wird mit der konst. ≤ 255 UND-verknüpft
ORA r ₁	10110sss	Z,S,P,CY,AC	1	4	Or register with accumulator	r ₁ =A,B,C,D,E,H oder L: Akkumulator- Inhalt wird mit dem Inhalt des Registers r ₁ ODER-verknüpft
ORA M	10110110	Z,S,P,CY,AC	1	7	Or memory with accumulator	Inhalt des über Registerpaar (H, L) adressierten Bytes wird mit Akku- mulator-Inhalt ODER-verknüpft
ORI konst.	11110110	Z,S,P,CY,AC	2	7	Or immediate with accumulator	Akkumulator-Inhalt wird mit konst. ≤ 255 ODER-verknüpft

Mnemonic	Binärkode	Beeinfl. Zustands- Bits	Anzahl der		Englische Befehls- beschreibung	Funktion des Befehls
			Bytes	Takt- zyklen		
XRA r ₁	10101sss	Z,S,P,CY,AC	1	4	Exclusive Or register with accumulator	r ₁ =A,B,C,D,E,H oder L: Akkumulator-Inhalt wird mit dem Inhalt des Registers r ₁ Exclusive-ODER-verknüpft
XRA M	10101110	Z,S,P,CY,AC	1	7	Exclusive Or memory with accumulator	Das über Registerpaar (H, L) adressierte Byte wird durch Exclusive-ODER mit dem Akkumulator verknüpft
XRI konst.	11101110	Z,S,P,CY,AC	2	7	Exclusive Or immediate with accumulator	Der Akkumulator wird mit dem Wert konst. < 255 durch Exclusive-ODER verknüpft
CMP r ₁	10111sss	Z,S,P,CY,AC	1	4	Compare register with accumulator	r ₁ =A,B,C,D,E,H oder L: Akkumulator-Inhalt wird mit dem Inhalt des Registers r ₁ verglichen
CMP M	10111110	Z,S,P,CY,AC	1	7	Compare memory with accumulator	Akkumulator-Inhalt wird mit des durch Registerpaar (H, L) adressierten Bytes verglichen
CPI konst.	11111110	Z,S,P,CY,AC	2	7	Compare immediate with accumulator	Akkumulator-Inhalt wird mit konst. < 255 verglichen

Mnemonic	Binärcode	Beeinfl. Zustands- Bits	Anzahl der Bytes		Englische Befehls- beschreibung	Funktion des Befehls				
			Takt- zyklen							
REGISTERANWEISUNGEN										
a) Akkumulator rotieren										
RLC	00000111	- - - CY,-	1	4	Rotate accumulator left	Akkumulator-Inhalt wird zyklisch um 1 Bit nach links verschoben. Bit 2^7 wird in das Carrybit geschrieben				
RRC	00001111	- - - CY,-	1f	4	Rotate accumulator right	Akkumulator-Inhalt wird zyklisch um 1 Bit nach rechts verschoben. Bit 2^0 wird in das Carrybit geschrieben				
RAL	00010111	- - - CY,-	1	4	Rotate accumulator left through carry	Akkumulator-Inhalt wird um 1 Bit nach links verschoben. Bit 2^7 wird in das Carrybit und das Carrybit in das Bit 2^0 geschrieben				
RAR	00011111	- - - CY,-	1	4	Rotate accumulator right through carry	Akkumulator-Inhalt wird um 1 Bit nach rechts geschoben. Bit 2^0 wird in das Carrybit und das Carrybit in das Bit 2^7 geschrieben				
b) Übertragsbit-Anweisungen										
CMC	00111111	- - - CY,-	1	4	Complement carry	Carrybit wird negiert				
STC	00110111	- - - CY,-	1	4	Set carry	Carrybit wird gesetzt				

Mnemonic	Binärkode	Beeinfl. Zustands- Bits	Anzahl der Bytes		Englische Befehls- beschreibung	Funktion des Befehls
			Takt- zyklen			

SPRUNGBEFEHLE**a) Unbedingte Sprünge**

PCHL	11101001	- - - - -	1	5	H & L to program counter	Programm wird an der Adresse fortgesetzt, die im Registerpaar (H, L) steht
JMP adr	11000011	- - - - -	3	10	Jump unconditional	Programm wird an der Adresse fortgesetzt

b) Bedingte Sprünge

JC adr	11011010	- - - - -	3	10	Jump on carry	Bei Carrybit = 1 wird das Programm bei der Adresse adr fortgesetzt
JNC adr	11010010	- - - - -	3	10	Jump on no carry	Bei Carrybit = 0 wird das Programm bei der Adresse adr fortgesetzt
JZ adr	11001010	- - - - -	3	10	Jump on zero	Bei Zerobit = 1 wird das Programm bei der Adresse adr fortgesetzt
JNZ adr	11000010	- - - - -	3	10	Jump on no zero	Bei Zerobit = 0 wird das Programm bei der Adresse adr fortgesetzt
JM adr	11111010	- - - - -	3	10	Jump on minus	Bei Signbit = 1 wird das Programm bei der Adresse adr fortgesetzt

Mnemonic	Binärcode	Beeinfl. Zustands- Bits	Anzahl der Bytes		Englische Befehls- beschreibung	Funktion des Befehls
			Takt- zyklen			
JP adr	11110010	- - - - -	3	10	Jump on positiv	Bei Signbit = 0 wird das Programm bei der Adresse adr fortgesetzt
JPE adr	11101010	- - - - -	3	10	Jump on parity even	Bei Paritätsbit = 1 wird das Programm bei der Adresse adr fortgesetzt
JPO adr	11100010	- - - - -	3	10	Jump on parity odd	Bei Paritätsbit = 0 wird das Programm bei der Adresse adr fortgesetzt

UNTERPROGRAMMBEHANDLUNG**a) Unterprogrammaufrufe**

Bei allen Aufrufbefehlen wird die Rückkehradresse im Stack abgelegt

CALL adr	11001101	- - - - -	3	17	Call unconditional	Programm wird bei der Adresse adr fortgesetzt
CC adr	11011100	- - - - -	3	11/17	Call on carry	Bei Carrybit = 1 wird das Programm bei der Adresse adr fortgesetzt
CNC adr	11010100	- - - - -	3	11/17	Call on no carry	Bei Carrybit = 0 wird das Programm bei der Adresse adr fortgesetzt
CZ adr	11001100	- - - - -	3	11/17	Call on zero	Bei Zerobit = 1 wird das Programm bei der Adresse adr fortgesetzt
CNZ adr	11000100	- - - - -	3	11/17	Call on no zero	Bei Zerobit = 0 wird das Programm bei der Adresse adr fortgesetzt

Mnemonic	Binärcode	Beeinfl. Zustands- Bits	Anzahl der		Englische Befehls- beschreibung	Funktion des Befehls
			Bytes	Takt- zyklen		
CM adr	11111100	-----	3	11/17	Call on minus	Bei Signbit = 1 wird das Programm bei der Adresse adr fortgesetzt
CP adr	11110100	-----	3	11/17	Call on positiv	Bei Signbit = 0 wird das Programm bei der Adresse adr fortgesetzt
CPE adr	11101100	-----	3	11/17	Call on parity even	Bei Paritätsbit = 1 wird das Programm bei der Adresse adr fortgesetzt
CPO adr	11100100	-----	3	11/17	Call on parity odd	Bei Paritätsbit = 0 wird das Programm bei der Adresse adr fortgesetzt
RST konst.	11nnn111	-----	1	11	Restart	0 konst. 7 Programm wird an der Adresse 8x konst. fortgesetzt

b) Rücksprungbefehle

RET	11001001	-----	1	10	Return	Programm wird an der Adresse fortgesetzt, die in dem Wort steht, das durch den Stackpointer adressiert ist
RC	11011000	-----	1	5/11	Return on carry	Bei Carrybit = 1 wird das Programm an der Adresse fortgesetzt, die in dem vom Stackpointer adressierten Wort steht
RNC	11010000	-----	1	5/11	Return on no carry	Bei Carrybit = 0 wird das Programm an der Adresse fortgesetzt, die in dem vom Stackpointer adressierten Wort steht

Mnemonic	Binärcode	Beeinfl. Zustands- Bits	Anzahl der Bytes		Englische Befehls- beschreibung	Funktion des Befehls
			Takt- zyklen			
RZ	11001000	- - - - -	1	5/11	Return on zero	Bei Zerobit = 1 wird das Programm an der Adresse fortgesetzt, die in dem vom Stackpointer adressierten Wort steht
RNZ	11000000	- - - - -	1	5/11	Return on no zero	Bei Zerobit = 0 wird das Programm an der Adresse fortgesetzt, die in dem vom Stackpointer adressierten Wort steht
RM	11111000	- - - - -	1	5/11	Return on minus	Bei Signbit = 1 wird das Programm an der Adresse fortgesetzt, die in dem vom Stackpointer adressierten Wort steht
RP	11110000	- - - - -	1	5/11	Return on positiv	Bei Signbit = 0 wird das Programm an der Adresse fortgesetzt, die in dem vom Stackpointer adressierten Wort steht
RPE	11101000	- - - - -	1	5/11	Return on parity even	Bei Paritätsbit = 1 wird das Programm an der Adresse fortgesetzt, die in dem vom Stackpointer adressierten Wort steht
RPO	11100000	- - - - -	1	5/11	Return on parity odd	Bei Paritätsbit = 0 wird das Programm an der Adresse fortgesetzt, die in dem vom Stackpointer adressierten Wort steht

Mnemonic	Binärcode	Beeinfl. Zustands- Bits	Anzahl der		Englische Befehls- beschreibung	Funktion des befehls
			Bytes	Takt- zyklen		
UNTERBRECHUNGSBEFEHLE						
EI	11111011	- - - - -	1	4	Enable interrupts	INTE-Flipflop wird gesetzt; der Mikroprozessor kann Unterbrechungen annehmen
DI	11110011	- - - - -	1	7	Disable interrupts	INTE-Flipflop wird rückgesetzt; der Mikroprozessor ignoriert Unterbrechungsanforderungen
ÜBRIGE BEFEHLE						
HLT	01110110	- - - - -	1	7	Halt	Programm hält an, bis eine Unterbrechungsanforderung eintrifft
NOP	00000000	- - - - -	1	4	No operation	Leerbefehl
PSEUDO BEFEHLE						
ORG adr	---	---	-	-	Origin	Der Adreßpegel des Assemblerprogrammes wird auf den Wert des Operanden gesetzt
name EQU ausdr	---	---	-	-	Equate	Dem Symbol "name" wird der Wert "ausdr" zugeordnet. Ein Name darf nur einmal im Namensfeld einer EQU-Anweisung erscheinen.

Mnemonic	Binärcode	Beeinfl. Zustands- Bits	Anzahl der		Englische Befehls- beschreibung	Funktion des Befehls
			Bytes	Takt- zyklen		
name:DB byte 1,...	---	---	-	-	Define bytes	Definieren von Bytes
name:DW wort 1,...	---	---	-	-	Define words	Wortdefinition
name:DS wert	---	---	-	-	Define storage	Bereiche definieren
STITLE text	---	---	-	-	Title	Seitenüberschrift
ZPW dezimal- zahl	---	---	-	-	Page width	Bestimmung der Zeilenlänge
ZPL dezimal- zahl	---	---	-	-	Page length	Bestimmung der Seitenlänge
ZE	---	---	-	-	Exchange	Blattwechsel

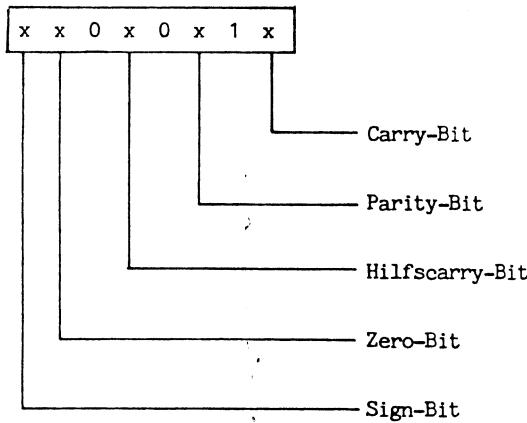
2.2 Befehlsliste in Hex-Folge

00	NOP		2A	LHLD	adr	50	MOV	D,B
01	LXI	B,adr	2B	DCX	H	51	MOV	D,C
02	STAX	B	2C	INR	L	52	MOV	D,D
03	INX	B	2D	DCR	L	53	MOV	D,E
04	INR	B	2E	MVI	L ,data	54	MOV	D,H
05	DCR	B	2F	CMA		55	MOV	D,L
06	MVI	B,data				56	MOV	D,M
07	RLC		30	---		57	MOV	D,A
08	---		31	LXI	SP,adr	58	MOV	E,B
09	DAD	B	32	STA	adr	59	MOV	E,C
0A	LDAX	B	33	INX	SP	5A	MOV	E,D
0B	DCX	B	34	INR	M	5B	MOV	E,E
0C	INR	C	35	DCR	M	5C	MOV	E,H
0D	DCR	C	36	MVI	M,data	5D	MOV	E,L
0E	MVI	C,data	37	STC		5E	MOV	E,M
0F	RRC		38	---		5F	MOV	E,A
			39	DAD	SP			
10	---		3A	LDA	adr	60	MOV	H,B
11	LXI	D,adr	3B	DCX	SP	61	MOV	H,C
12	STAX	D	3C	INR	A	62	MOV	H,D
13	INX	D	3D	DCR	A	63	MOV	H,E
14	INR	D	3E	MVI	A,data	64	MOV	H,H
15	DCR	D	3F	CMA		65	MOV	H,L
16	MVI	D,data				66	MOV	H,M
17	RAL		40	MOV	B,B	67	MOV	H,A
18	---		41	MOV	B,C	68	MOV	L,B
19	DAD	D	42	MOV	B,D	69	MOV	L,C
1A	LDAX	D	43	MOV	B,E	6A	MOV	L,D
1B	DCX	D	44	MOV	B,H	6B	MOV	L,E
1C	INR	E	45	MOV	B,L	6C	MOV	L,H
1D	DCR	E	46	MOV	B,M	6D	MOV	L,L
1E	MVI	E,data	47	MOV	B,A	6E	MOV	L,M
1F	RAR		48	MOV	C,B	6F	MOV	L,A
			49	MOV	C,C			
20	---		4A	MOV	C,D	70	MOV	M,B
21	LXI	H,adr	4B	MOV	C,E	71	MOV	M,C
22	SHLD	adr	4C	MOV	C,H	72	MOV	M,D
23	INX	H	4D	MOV	C,L	73	MOV	M,E
24	INR	H	4E	MOV	C,M	74	MOV	M,H
25	DCR	H	4F	MOV	C,A	75	MOV	M,L
26	MVI	H,data				76	HLT	
27	DAA					77	MOV	M,A
28	---					78	MOV	A,B
29	DAD					79	MOV	A,C

7A	MOV	A,D	A8	XRA	B	D5	PUSH	D
7B	MOV	A,E	A9	XRA	C	D6	SUI	data
7C	MOV	A,H	AA	XRA	D	D7	RST	2
7D	MOV	A,L	AB	XRA	E	D8	RC	
7E	MOV	A,M	AC	XRA	H	D9	---	
7F	MOV	A,A	AD	IRA	L	DA	JC	adr
			AE	XRA	M	DB	IN	kanalnr
80	ADD	B	AF	XRA	A	DC	CC	adr
81	ADD	C				DD	---	
82	ADD	D	B0	ORA	B	DE	SBI	data
83	ADD	E	B1	ORA	C	DF	RST	3
84	ADD	H	B2	ORA	D			
85	ADD	L	B3	ORA	E	E0	PRO	
86	ADD	M	B4	ORA	H	E1	POP	H
87	ADD	A	B5	ORA	L	E2	JPO	adr
88	ADC	B	B6	ORA	M	E3	XTHL	
89	ADC	C	B7	ORA	A	E4	CPO	adr
8A	ADC	D	B8	CMP	B	E5	PUSH	H
8B	ADC	E	B9	CMP	C	E6	ANI	data
8C	ADC	H	BA	CMP	D	E7	RST	4
8D	ADC	L	BB	CMP	E	E8	RPE	
8E	ADC	M	BC	CMP	H	E9	PCHL	
8F	ADC	A	BD	CMP	L	EA	JPE	adr
			BE	CMP	M	EB	XCHG	
90	SUB	B	BF	CMP	A	EC	CPE	adr
91	SUB	C				ED	---	
92	SUB	D	C0	RNZ		EE	XRI	data
93	SUB	E	C1	POP	B	EF	RST	5
94	SUB	H	C2	JNZ	adr	FO	RP	
95	SUB	L	C3	JMP	adr	F1	POP	PSW
96	SUB	M	C4	CNZ	adr	F2	JP	adr
97	SUB	A	C5	PUSH	B	F3	DI	
98	SBB	B	C6	ADI	data	F4	CP	adr
99	SBB	C	C7	RST	0	F5	PUSH	PSW
9A	SBB	D	C8	RZ		F6	ORI	data
9B	SBB	E	C9	RET		F7	RST	6
9C	SBB	H	CA	JZ	adr	F8	RM	
9D	SBB	L	CB	---		F9	SPHL	
9E	SBB	M	CC	CZ	adr	FA	JM	adr
9F	SBB	A	CD	CALL	adr	FB	EI	
A0	ANA	B	CE	ACI	data	FC	CM	adr
A1	ANA	C	CF	RST	1	FD	---	
A2	ANA	D				FE	CPI	data
A3	ANA	E	D0	RNC		FF	RST	7
A4	ANA	H	D1	POP	D			
A5	ANA	L	D2	JNC	adr			
A6	ANA	M	D3	OUT	kanalnr			
A7	ANA	A	D4	CNC	adr			

2.3 Aufbau des Status-Bytes

7 6 5 4 3 2 1 0



3 ASSEMBLER STANDARDROUTINENSyntax

	CALL	routine
Parameter	DB	länge
	DW	adr-1
	DW	adr-2

Vorzeichen-Werte

F (=POSITIV)

C (=POSITIV)

D (=NEGATIV)

Bedeutung	CALL-Routine	Parameter			Bedingungen		Ergebnis	
		max. Feldlänge (Byte)	adr-1	adr-2	EXTERNAL	zu bindende Module	fehlerloser Ablauf	Fehlermeldung im Akku.
Übertragen Zeichenfolge	MVC	255	Zielfeld	Sendefeld	MVC	{ :Fn:movmod.rel :Fn:hilmod.rel }	-	6
Übertragen m. Versetzen	MVO	15	"	"	MVO	"	-	6
Übertragen numerisch	MVN	255	"	"	MVN	"	-	6
Übertragen Zonen	MVZ	255	"	"	MVZ	"	-	6
Aufbereiten	ED	255	Maske *	gepackter Sender	ED	{ :Fn:edtmod.rel :Fn:hilmod.rel }	-	3 und 6
Umsetzen und Testen	TRT	255	Argumentenfeld	Funktionsfeld	TRT	"	Z(=0) 0(≠0) N(letztes Byte=0)	6

* Aufbereitungszeichen:

Zeichenart:	Hex. Darst.:	Funktion
Füllzeichen	beliebig z.B.*	ersetzt führende Nullen
Beginn der geltenden Ziffern	81H	beendet das Ersetzen führender Nullen und führt "Ziffernauswahl" durch
Ziffernauswahlzeichen	80H	bewirkt das Einsetzen einer Ziffer aus dem Sender
einzufügendes Zeichen	beliebig	wird im Ergebnisfeld eingefügt Interpunktion

Bedeutung	CALL-Routine	Parameter			Bedingungen		Ergebnis	
		max. Feldlänge (Byte)	adr-1	adr-2	EXTERNAL	zu bindende Module	fehlerloser Ablauf	Fehlermeldung im Akku.
Addieren gepackt	AP	15	Summand Summe	Summand	AP	{ :Fn:decmod.rel :Fn:hilmod.rel	Z(= 0) L(< 0) H(> 0) O(Uberlauf) "	3, 5, 6
Subtrahieren gepackt	SP	15	Minuend Differenz	Subtrahend	SP	"	"	"
Löschen und addieren	ZAP	15	Operand-A	Operand-B	ZAP	"	"	"
Vergleiche gepackt	CP	15	Operand-A	Operand-B	CP	"	E(gleich) L(A kleiner) H(A größer) -	3, 6
Multipliziere gepackt	MP	A=15 B= 8	Multiplikand(A) Produkt	Multiplikator(B)	MP	"	-	1, 2, 3, 6
Dividiere gepackt	DP	15	Dividend Quotient, Rest	Divisor	DP	"	-	2, 3, 6
Packen	PACK	15	gepacktes Ergebnis	entpackter Operand	PACK	"	-	6
Entpacken	UNPK	15	entpacktes Ergebnis	gepackter Operand	UNPK	"	-	6
Umwandeln in Binärform	CVB	-	Operand-A	Operand-B	CVB	"	-	3, 7
Umwandeln in Dezimalform	CVD	-	Operand-A	Operand-B	CVD	"	-	-

Bedeutung	CALL-Routine	Parameter			Bedingungen		Ergebnis	
		max. Feldlänge (Byte)	adr-1	adr-2	EXTERNAL	zu bindende Module	fehlerloser Ablauf	Fehlermeldung im Akku.
Addieren binär	AB	15	{ Summand Summe }	Summand	AB	{ :Fn:binmod.rel :Fn:hilmod.rel }	{ Z(=0) N(≠0) O(Uberlauf) }	6
Subtrahiere binär	SB	15	{ Minuend Differenz }	Subtrahend	SB	"	"	6
Vergleiche binär	COMP	15	Operand-A	Operand-B	COMP	"	{ E(gleich) L(A kleiner) H(A größer) }	6
Vergleiche gepackt	CLC	255	Operand-A	Operand-B	CLC	{ :Fn:logmod.rel :Fn:hilmod.rel }	{ E(gleich) L(A kleiner) H(A größer) }	6
logisches UND	AND	255 A=B	{ Operand-A Ergebnis }	Operand-B	AND	"	{ Z(=0) N(≠0) }	6
logisches ODER	OR	255	{ Operand-A Ergebnis }	Operand-B	OR	"	"	6
Ausschließendes ODER	ORX	255	{ Operand-A Ergebnis }	Operand-B	ORX	"	"	6
Test mit Maske	TM	-	Testbyte	Maskenbyte	TM	"	{ Z(=0) O(=1) C(gemischt) }	-

4. Makroprozessor STAGE 2

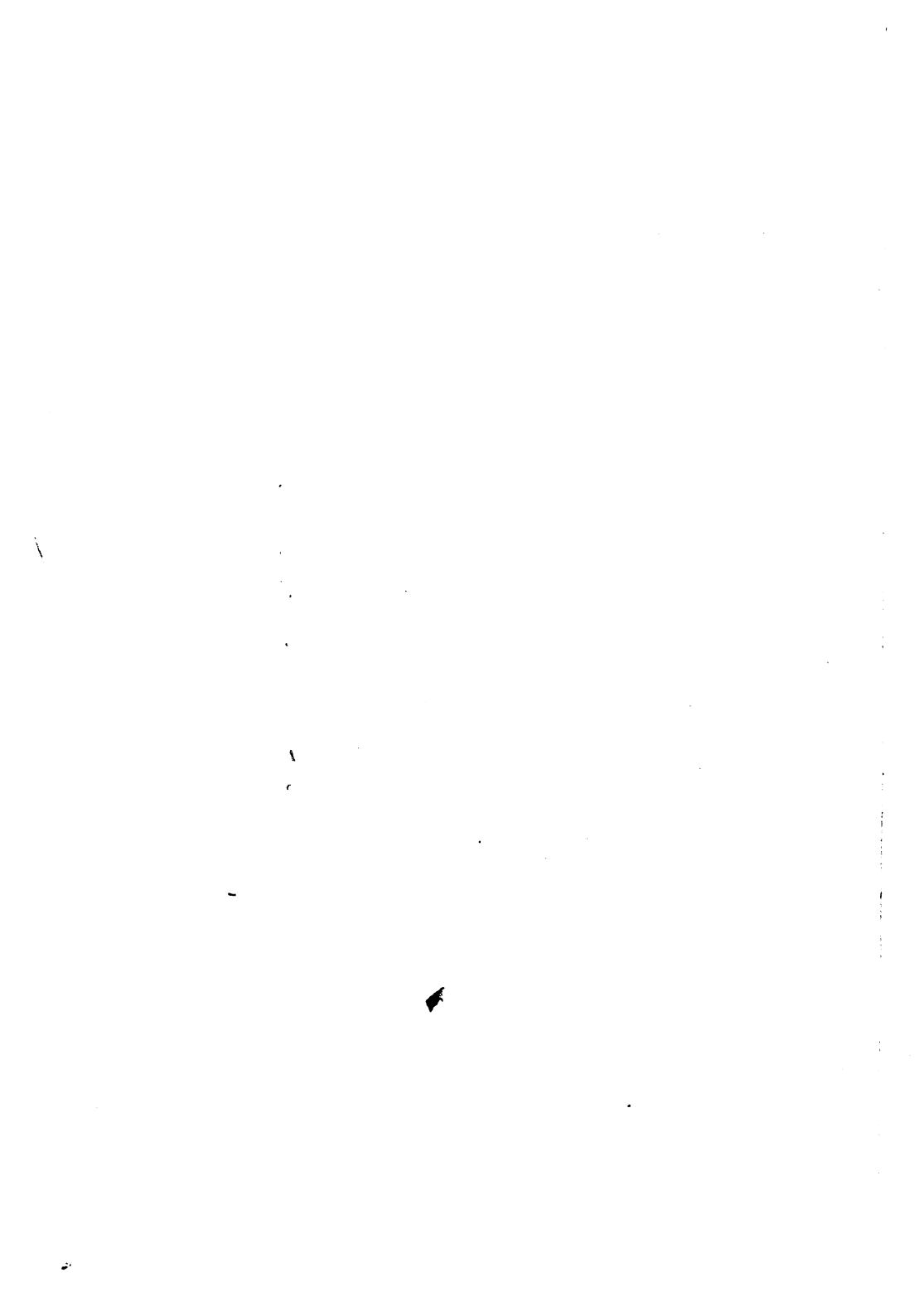
= für Programmgenerierung und Programmabwicklung

Aufruf

STAGE2 SI=:lw:eingabe,SO=:lw:ausgabe,SL=liste

Eingabe: Makrodefinition und Programm mit
Makroaufrufen

Ausgabe: Programm mit Makroexpansionen



5. FehlermeldungenASM

Fehler- meldung	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
14	Symbol doppelt definiert
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- und Kassettenlaufwerk nicht bereit
57	Schreibsperrre auf der Kassette

RASM

<u>Fehler-</u> <u>meldung</u>	<u>Bedeutung</u>
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 Kompiliertdurchgang. Symbole in ORG-Ausdrücken müssen definiert sein, bevor sie verwendet werden.
06	ECU-Anweisung mit unzulässigem Operanden
07	Unzulässiger Ein-Byte-Ausdruck
08	Fehler bei Sprungmarke (s. Fehler 05)
09	Unzulässiger Ausdruck
10	Fehlernder Ausdruck
11	Undefiniertes Symbol
12	Symbol mehrfach definiert
13	Falsche Register-Angabe

ASSEMBLER-Standardroutinen

<u>Fehler-</u> <u>meldung</u>	<u>Bedeutung</u>
1	Anzahl führender Nullen des Multiplikanden ist zu klein
2	Multiplikator/Divisor länger als 8 Bytes
3	Ein dezimaler Operand ist falsch
4	Division durch Null
5	Bei einer arithmetischer Operation trat ein Überlauf auf
6	Längenangabe für Operanden fehlt
7	Dezimaler Operand ist zu groß (bei CVB)



1. Sprachformat COBOL-A

IDENTIFICATION DIVISION.
PROGRAM-ID. programname.
[AUTHOR. kommentar.]
[INSTALLATION. kommentar.]
[DATE-WRITTEN. kommentar.]
[SECURITY. kommentar.]

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER 6.610. [DEBUGGING MODE.]
OBJECT-COMPUTER 6.610.

[SPECIAL-NAMES. [DECIMAL-POINT IS COMMA.]
[CURRENCY SIGN IS literal.]

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT dateiname

ASSIGN externdateiname

ORGANIZATION { SEQUENTIAL
INDEXED
RELATIVE } (Bei Fehlen wird
zeilen-sequentielle
Organisation angenommen)

ACCESS { SEQUENTIAL
DYNAMIC } Standardannahme:
SEQUENTIAL

STATUS datenname-1

{ RECORD
RELATIVE } KEY datenname-2.

DATA DIVISION.

FILE SECTION.

FD dateiname RECORD ganzzahl.

Datensatzerklärung... (VALUE nicht erlaubt)

WORKING-STORAGE SECTION.

Datensatzerklärung...

Datensatzerklärung:

Stufenummer [datenname-1]
 FILLER

[REDEFINES datenname-2]

[PIC zeichenfolge]

[OCCURS ganzzahl]

[VALUE literal].

PROCEDURE DIVISION..

[paragraphenname.]

anweisung

anweisung

[paragraphenname ...]

Anweisungen:

ACCEPT datenname-1 [AT datenname-2] [FROM CRT].
 literal-1] .

ADD { bezeichner-1 } TO bezeichner-2 [ROUNDED].
 literal-1 }

CLOSE dateiname .

COPY bibliotheksname .

DELETE dateiname INVALID unbedingte anweisung .

DISPLAY { datenname-1 } [AT { datenname-2 }] [UPON { CRT }
 literal-1 } { literal-2 }] [CRT-UNDER].

DIVIDE { bezeichner-1 } INTO bezeichner-2 [ROUNDED].
 literal-1 }

EXIT.

GO TO prozedurname [prozedurname-2, ...] [DEPENDING ON bezeichner].

IF bedingung THEN { anweisung-1 } [ELSE { anweisung-2 }].
 NEXT SENTENCE] .

MOVE { bezeichner-1 } TO bezeichner-2 .
 literal }

MULTIPLY { bezeichner-1 } BY bezeichner-2 [ROUNDED].
 literal-1 }

OPEN { INPUT } dateiname .
 { OUTPUT }
 I-O

PERFORM prozedurname-1 [THRU prozedurname-2]
 [UNTIL bedingung-1].

READ dateiname [NEXT] [END
 INVALID] unbedingte anweisung].

REWRITE datensatzname [INVALID] unbedingte anweisung].

STOP RUN.

SUBTRACT { bezeichner-1 } FROM bezeichner-2 ROUNDED .
 literal-1 }

WRITE datensatzname [BEFORE
 AFTER { ganzzahl }
 PAGE
 TAB] unbedingte anweisung].
 INVALID] .

2. Sprachformat COBOL B

IDENTIFICATION DIVISION.
PROGRAM-ID. programname.
[AUTHOR. kommentar.]
[INSTALLATION. kommentar.]
[DATE-WRITTEN. kommentar.]
[DATE-COMPILED. kommentar.]
SECURITY. kommentar.]

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. 6.610 [DEBUGGING MODE].
OBJECT-COMPUTER. 6.610 [,MEMORY SIZE 54 CHARACTERS].

SPECIAL-NAMES. [DECIMAL-POINT IS COMMA].
[CURRENCY SIGN IS literal].
[CONSOLE IS CRT].
[CURSOR IS datename].

INPUT-OUTPUT SECTION.

FILE-CONTROL.

```
SELECT dateiname
```

ASSIGN externdateiname

ORGANIZATION	{ SEQUENTIAL LINE SEQUENTIAL INDEXED RELATIVE }	Standardannahme: SEQUENTIAL
--------------	--	--------------------------------

ACCESS { SEQUENTIAL } Standardannahme:
 { DYNAMIC } SEQUENTIAL
 { RANDOM }

[STATUS datename-1]

[{ RECORD
RELATIVE } KEY datenname-2 .]

DATA DIVISION.
 [FILE SECTION.]
 [FD dateiname]
 [BLOCK CONTAINS ganzzahl-2 { RECORD
CHARACTERS }]
 [RECORD CONTAINS ganzzahl-3 TO ganzzahl-4 CHARACTERS]
 [LABEL { RECORD IS } { STANDARD }]
 [RECORDS ARE OMITTED]
 [Datensatzerklärung] ... (VALUE nicht erlaubt)
 [WORKING-STORAGE SECTION]
 [Datensatzerklärung] ...

Datensatzerklärung:

stufenummer datename-1
 FILLER
 [REDEFINES datename-2]
 [PIC zeichenfolge]
 [OCCURS ganzzahl-1] [TIMES] [INDEXED BY indexname ...]
 [USAGE IS] { COMPUTATIONAL
COMP
COMPUTATIONAL-3
COMP-3
DISPLAY }]
 [SIGN IS] { LEADING } { SEPERATE CHARACTERS }
 [TRAILING]
 [JUSTIFIED }]
 [JUST } RIGHT]
 [BLANK WHEN ZERO]
 [VALUE is literal]

PROCEDURE DIVISION [USING datenname-1 ...].

[DECLARATIVES.]
[sectionname SECTION] [segmentnr].
[paragraphenname]
anweisung ...
...

[sectionname SECTION] [segmentnr].
[paragraphenname]
anweisung ...
anweisung ...
[paragraphenname]
anweisung ...
anweisung ...
[sectionname SECTION] [segmentnr].
...

Anweisungen:

ACCEPT bezeichner [FROM CONSOLE]
 ACCEPT datename-1 AT {datename-2}
 {literal-1} FROM CRT
 ADD {bezeichner-1}, {bezeichner-2}.. TO bezeichner-m [ROUNDED]
 {literal-1}, {literal-2}... [, bezeichner-n [ROUNDED] ... ; ON SIZE ERROR unbedingte anweisung]
 ADD {bezeichner-1}, {bezeichner-2}, {bezeichner-3}...
 {literal-1}, {literal-2}, {literal-3}...
 GIVING bezeichner-m ROUNDED , bezeichner-n ROUNDED ...
 [, ON SIZE ERROR unbedingte anweisung]
 ALTER prozedurname-1 TO PROCEED TO prozedurname-2
 CLOSE dateiname-1 [, dateiname-2] ...
 COPY bibliotheksname.
 DELETE dateiname RECORD [; INVALID KEY unbedingte anweisung]
 DISPLAY {bezeichner-1} [, {bezeichner-2}][... UPON CONSOLE]
 {literal-1}, {literal-2}...
 DISPLAY {datename-1} [AT {datename-2}] UPON {CRT}
 {literal-3}, {literal-4}... {CRT UNDER}
 DIVIDE {bezeichner-1 INTO} bezeichner-2 [ROUNDED]
 {literal-1} [, bezeichner-3 [ROUNDED]]
 [ON SIZE ERROR unbedingte anweisung]

DIVIDE { bezeichner-1 } INTO { bezeichner-2 }
 literal-1 } BY } literal-2 }
 GIVING bezeichner-3 [ROUNDED] [bezeichner-4 [ROUNDED]]...
 [; ON SIZE ERROR unbedingte anweisung]

EXIT.

GO TO prozedurname-1 [,prozedurname-2]... [,prozedurname-n]
 [DEPENDING ON bezeichner]

IF Bedingung; { anweisung-1 } { ELSE anweisung-2 }
 { NEXT SENTENCE } { ELSE NEXT SENTENCE }

INSPECT bezeichner-1 TALLYING bezeichner-2 FOR { ALL
 LEADING CHARACTERS }
 { bezeichner-3 } { BEFORE } INITIAL { bezeichner-7 }
 literal-1 } { AFTER } { literal-5 }

INSPECT bezeichner-1 REPLACING

{ CHARACTERS BY { bezeichner-6 }
 { literal-4 } }
 { ALL } { bezeichner-5 } BY { bezeichner-6 }
 { LEADING } { literal-3 } { literal-4 }
 { FIRST }

[{ BEFORE } INITIAL { bezeichner-7 }]
 [{ AFTER } { literal-5 }]

Format 3

INSPECT bezeichner-1 TALLYING bezeichner-2 FOR { ALL
LEADING CHARACTERS }

{ bezeichner-3 } { BEFORE } INITIAL { bezeichner-4 }
 literal-1 { AFTER } { literal-2 }

REPLACING

{ CHARACTERS BY bezeichner-6
literal-4 }

{ ALL } { bezeichner-5 } BY { bezeichner-6 }
 { LEADING } { literal-3 } { literal-4 }

{ FIRST } { bezeichner-7 }
 { BEFORE } INITIAL { literal-5 }
 { AFTER }

MOVE { bezeichner-1 } TO bezeichner-2 [,bezeichner-3]...
 { literal }

MULTIPLY { bezeichner-1 } BY bezeichner-2 [ROUNDED]
 { literal-1 }

[,bezeichner-3 [ROUNDED]]... [;ON SIZE ERROR unbedingte
anweisung]

MULTIPLY { bezeichner-1 } BY { bezeichner-2 } GIVING bezeichner-3 [ROUNDED]
 { literal-1 } { literal-2 }

[,bezeichner-4 [ROUNDED]]... [;ON SIZE ERROR unbedingte
anweisung]

OPEN { INPUT dateiname-1 [,dateiname-2] ... }...
 { OUTPUT dateiname-3 [,dateiname-4] ... }...
 { I-O dateiname-5 [,dateiname-6] ... }

PERFORM prozedurname-1 $\left[\begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right]$ prozedurname-2 $\left[\begin{array}{l} \left\{ \begin{array}{l} \text{bezeichner-1} \\ \text{ganzzahl-1} \end{array} \right\} \text{TIMES} \\ \text{UNTIL bedingung-1} \end{array} \right]$
 READ dateiname[NEXT] RECORD [INTO bezeichner]
 [AT END unbedingte anweisung]
 [INVALID KEY unbedingte anweisung]
 REWRITE datensatzname [FROM bezeichner]
 [INVALID KEY unbedingte anweisung]
 SET $\left\{ \begin{array}{l} \text{bezeichner-1} \\ \text{indexname-1} \end{array} \right\} \dots \left[\begin{array}{l} \text{bezeichner-2} \\ \text{indexname-2} \end{array} \right] \dots \text{TO} \left\{ \begin{array}{l} \text{bezeichner-3} \\ \text{indexname-3} \\ \text{ganzzahl-1} \end{array} \right\}$
 SET indexname-4 [, indexname-5] .. $\left\{ \begin{array}{l} \text{UP BY} \\ \text{DOWN BY} \end{array} \right\} \left\{ \begin{array}{l} \text{bezeichner-4} \\ \text{ganzzahl-2} \end{array} \right\}$
 STOP $\left\{ \begin{array}{l} \text{RUN} \\ \text{Literal} \end{array} \right\}$
 SUBTRACT $\left\{ \begin{array}{l} \text{bezeichner-1} \\ \text{literal-1} \end{array} \right\}, \left\{ \begin{array}{l} \text{bezeichner-2} \\ \text{literal-2} \end{array} \right\} \dots \text{FROM bezeichner-m}$
 [ROUNDED][, bezeichner-n[ROUNDED]]...[;ON SIZE ERROR unbedingte anweisung]
 SUBTRACT $\left\{ \begin{array}{l} \text{bezeichner-1} \\ \text{literal-1} \end{array} \right\}, \left\{ \begin{array}{l} \text{bezeichner-2} \\ \text{literal-2} \end{array} \right\} \dots \text{FROM bezeichner-m}$
 GIVING bezeichner-n [ROUNDED][, bezeichner-o [ROUNDED]]...
 [ON SIZE ERROR unbedingte anweisung]
 USE AFTER STANDARD $\left\{ \begin{array}{l} \text{EXCEPTION} \\ \text{ERROR} \end{array} \right\}$ PROCEDURE ON $\left\{ \begin{array}{l} \text{dateiname-1} \\ \text{INPUT} \\ \text{OUTPUT} \\ \text{I-O} \\ \text{EXTEND} \end{array} \right\}$

WRITE datensatzname [FROM bezeichner-1]
[{ BEFORE } ADVANCING { ganzzahl
{ AFTER } TAB
{ PAGE
[{ LINE }
{ LINES } }]]
[INVALID KEY unbedingte anweisung]

3. Verarbeitung RFM-Dateien

Klausel Anweisung	Kombinationen
----------------------	---------------

FILE-CONTROL

ORGANIZATION	INDEXED	RELATIVE
ACCESS	DYNAMIC	DYNAMIC
KEY-Klausel	RECORD KEY	RELATIVE KEY
STATUS-Klausel	(STATUS)	(STATUS)

PROCEDURE DIVISION

Art des OPEN's	OUTPUT	INPUT	I-O
E/A Anweisung	WRITE (INVALID) .	READ NEXT END	READ NEXT (END) READ (INVALID) WRITE (INVALID) REWRITE (INVALID) DELETE (INVALID)
schließen	CLOSE		

4. Segmentierung

Segmentnummern:

- 0-49 für Root-Segmente
 50-99 für unabhängige Segmente

Fehlt die Segmentnummer wird 0 angenommen.

Einschränkungen im Programmfluß:

ALTER-Anweisung:

Eine GO TO Anweisung in einer Sektion die zum unabhängigen Bereich gehört darf nicht durch eine ALTER-Anweisung in einem anderen Segment angesprochen werden.

Perform-Anweisung

Prozedur in Perform- Anweisung in		Declarative Section	in dem gleichen Segment wie die Anweisung	im anderen Rootsegment als die Anweisung	im anderen unabhängigen Segment
Rootsegment		+	+	+	+
unabhängiges Segment		+	+	+	-

+ = möglich

- = nicht möglich

5. Interaktive DEBUG-Kommandos

A Datenadresse Wert

- Überschreiben des Inhalts von "Datenadresse" mit "Wert".

C Wert

- Ausgabe des ASCII-Zeichens, das "Wert" entspricht.

D Datenadresse

- Ausgabe von 16 Bytes, beginnend mit "Datenadresse". Die Ausgabe erfolgt in 2-Byte-Hexzeichen, denen jeweils das ASCII-Zeichen folgt, falls es abdruckbar ist.

G Proz.adresse

- Programmlauf bis "Prozeduradresse" erreicht ist (FFF: bis Programmende)

L

- Neue Zeile auf dem Bildschirm.

M Name Kommandos

- Definition des Makros "Name".

T Proz.adresse

- Programmlauf bis "Prozeduradresse" mit Ausgabe aller durchlaufenden Paragraphen.

X

- Ausführung einer Anweisung und Ausgabe des Befehlszählers (PC).

P

- Ausgabe des aktuellen PC.

S Datenadresse

- Laden des Adresszählers mit "Datenadresse" für '.' und '/' Kommandos.

S

- Ende einer Makrodefinition.

;

- Beginn des Kommentarteilers einer Zeile.

. Wert

- Überschreiben des durch den Adresszählerstand definierten Feldes mit "Wert" und Erhöhen des Adresszählers.

/

- Ausgabe des Adresszählerstandes und des zugehörigen Feldinhalts.

,

- Erhöhen des aktuellen Adresszählerstandes.

Q

- Textende, Aussprung aus dem DEBUG-Modul.

6. Übersetzung

{ COBOLA.48 } S:lw:programmdatei übersetzerkommandos
COBOLA.54 {
COBOLB }

Übersetzerkommandos:

ANS	ANSI '74 Syntaxprüfung
RESEQ	Neumerierung des Primärprogrammes
NOINT	Keine Zwischencodeausgabe
INTLIST	Zwischencode im Listing
NOLIST	Kein Listing
COPYLIST	Auflistung aller kopierten Zeilen
NOHEADING	Listing ohne Blattvorschub und Seitenüberschriften
ERRLIST	Nur Fehlerliste
INT (Externdateiname)	Name der Datei, in die der Zwischencode ausgegeben wird
LIST (Externdateiname)	Name der Datei, in welche das Listing ausgegeben wird
FORM (Ganzzahl)	Anzahl Programmzeilen pro Seite

7. Laufzeitsystem:

Temporäres Laden:

{ RUNA.48
RUNA.54 } \$ [Ladeparam] codedatei, unterprogramm ...
RUNB

Erzeugen eines ablauffähigen Programmes

{ RUNA.48
RUNA.54 } SO= programm\$ [Ladeparam]=codedatei, unterprogramm ...
RUNB

Ladeparameter

* mit DEBUG- und RFM-Modul

keine Angabe ohne DEBUG-, mit RFM-Modul

-I ohne DEBUG- und ohne RFM-Modul

8. Schnittstelle Sonderaufrufe

Aufrufendes COBOL-Programm	Aufgerufenes Programm
<p>CALL "LINK" USING datenname-1.</p> <p>Zu beachten:</p> <p>WORKING-STORAGE SECTION.</p> <p>01 datenname-1 PIC X(5) VALUE "PROG".</p> <p>Dateien werden geschlossen, es erfolgt kein Rücksprung.</p> <p>PROG ist ein ablauffähiges Programm.</p>	<p>Ablauffähiges COBOL-, ASSEMBLER- oder Basic-Programm.</p> <p>Weitere Verkettungen sind möglich</p>

9. Schnittstelle COBOL-ASSEMBLER

COBOL-Programm	ASSEMBLER-Programm
CALL literal [USING datenname-1,datenname-2, ...].	keine EXTERNAL, ORG, DSEG PUBLIC Befehlsadresse .
Literal muß eine gültige Befehls- adresse des Objektes sein.	.
Bei Übergabe von Parametern (max. 16) ist folgendes zu beachten:	.
WORKING-STORAGE SECTION. 01 datenname-1 PIC 01 datenname-2 PIC 01 datenname-n PIC Siehe Tabelle	RET END Bei Übergabe von Parametern siehe Tabelle.

COBOL

- Parameteradressen :

SCHNITTSTELLE COBOL-ASSEMBLER

Para- meter- anzahl	Registerpaar		Kellerspeicher (Stack)					
	BC	DE	letzter	letzter-1	letzter-n	letzter-n+1	
0	FF FF	FF FF	Rücksp. Adresse	FF FF				
1	1.Parameter Adresse	FF FF	Rücksp. Adresse	FF FF				
2	1.Parameter Adresse	2.Parameter Adresse	Rücksp. Adresse	FF FF				
3	2.Parameter Adresse	3.Parameter Adresse	Rücksp. Adresse	1.Parameter Adresse	FF FF			
.				
N	N-1.Parameter Adresse	N.Parameter Adresse	Rücksp. Adresse	N-2.Parameter Adresse	1.Parameter Adresse	FF FF	

rufendes Programm	gerufenes Programm
<p>LINKAGE SECTION</p> <p>WORKING-STORAGE SECTION</p> <p>01 datenname-1 PIC</p> <p>01 datenname-2 PIC</p> <p>.</p> <p>.</p> <p>01 datenname-n PIC</p> <p>PROCEDURE DIVISION</p> <p>CALL { bezeichner }</p> <p>{ literal }</p> <p>[USING datenname1, datenname2 ...]</p> <p>CANCEL { bezeichner-1 } [, { bezeichner-2 }] ...</p> <p>bezeichner bzw. literal muß den Namen der Zwischencodedatei des aufgerufenen Programmes enthalten.</p>	<p>LINKAGE SECTION</p> <p>01 datenname-1 PIC</p> <p>01 datenname-2 PIC</p> <p>.</p> <p>.</p> <p>01 datenname-n PIC</p> <p>PROCEDURE DIVISION [USING datenname-1, datenname2 ...]</p> <p>EXIT PROGRAM.</p>

10. Schnittstelle COBOL-COBOL

SCHNITTSTELLE COBOL-COBOL (COBOL B)

COBOL

11. Fehlermeldungen (Übersetzer COBOL A)

Fehler- Beschreibung
code

- 01 Datenname zu lang (oder enthält Kleinbuchstaben bei gesetztem ANS-Schalter)
02 Num. Literal zu lang oder enthält nicht num. Zeichen
03 Alphanum. Literal fehlt / falsch / unpassend
04 Datenname nicht definiert
05 Datenname mehrfach definiert
06 Zu viele Daten- und Prozedurnamen definiert
07 Unerlaubtes Zeichen in Spalte 7
08 "COPY" geschachtelt oder für unbekannte Datei
09 COBOL-Erweiterung benutzt bei gesetztem ANS- Schalter
10 "." fehlt
11 "DIVISION" fehlt
12 "SECTION" fehlt
13 "IDENTIFICATION" fehlt
14 "PROGRAM-ID" fehlt
15 "AUTHOR" fehlt
16 "INSTALLATION" fehlt
17 "DATE-WRITTEN" fehlt
18 "SECURITY" fehlt
19 "ENVIRONMENT" fehlt
20 "CONFIGURATION" fehlt
21 "SOURCE-COMPUTER" fehlt
22 Computername fehlt oder falsch
23 "MODE" fehlt
24 "OBJECT-COMPUTER" fehlt
25 "SPECIAL-NAMES" fehlt
26 "IS" fehlt
27 "COMMA" fehlt
28 "SIGN" fehlt
29 Falsches Währungssymbol
30 "." fehlt
31 "DIVISION" fehlt
32 "SECTION" fehlt
33 "INPUT-OUTPUT" fehlt
34 "FILE-CONTROL" fehlt
35 "ASSIGN" fehlt
36 "SEQUENTIAL" oder "INDEXED" oder "RELATIVE" fehlt
37 "ACCESS" fehlt bei indexsequentieller oder direkter Datei
38 "SEQUENTIAL" oder "DYNAMIC" fehlt
39 "RECORD" oder "RELATIVE" fehlt oder falsch
40 "DATA DIVISIONS" fehlt
41 "FILE SECTION" fehlt
42 "WORKING-STORAGE" fehlt
43 "PROCEDURE DIVISIONS" fehlt
44 "." fehlt
45 "DIVISION" fehlt
46 "SECTION" fehlt
47 SELECT-Klausel für Dateiname fehlt
48 Satzlänge nicht angegeben

Fehler- Beschreibung
code

66 Stufenummer nicht im Bereich 01 bis 49
67 Datensatzbereich größer als Satzlänge
68 "WORKING-STORAGE" fehlt
69 "PROCEDURE DIVISION" fehlt oder unbekannte Anweisung
70 "STATUS"-Datenname nicht auf Stufe 01 erklärt
71 "REDEFINES"-Datenname nicht definiert
72 "PICTure"-Klausel mit verbotener PIC-Zeichenfolge
73 Mit PIC beschriebenes Datenfeld zu lang (N=15, N/E=28, A/N=8192)
74 "OCCURS" geschachtelt oder Wiederholungsfaktor fehlt
75 "VALUE" fehlt oder für angegebene Datenkategorie unzulässig
76 "FILLER"-Definition ohne PICTurE-Angabe
77 Stufenummernsystem enthält Datengruppe mit Länge 0 oder
größer 8192 Bytes
78 "REDEFINES" für ungleiche Datenfelder oder unterschiedliche
Stufenummern
79 Stufenummernsystem mit mehr als 15 Schachtelungen
80 "DYNAMIC" nur erlaubt ohne ANS-Schalter und in SELECT-Klausel
81 Unbekannte Prozedurweisung
82 "." oder "ELSE" fehlt
83 Unerlaubte Datenkategorie
84 Verschachteltes IF
85 Unerlaubte Kombination von Datenkategorien in MOVE
86 "PAGE" oder "TAB" oder Ganzzahl fehlt
87 Paragraphenname identisch mit Datenname
88 Paragraphenname mehrfach definiert
89 Unerlaubte Kombination von Datenkategorien in arithm.
oder Vergleichsoperation
90 ACCEPT/DISPLAY fehlerhaft
91 Datenname für ACCEPT/DISPLAY oder AT gehört zu falscher
Datenkategorie oder nicht definiert
92 Datei nicht eröffnet
93 Operation unerlaubt für angegebene Dateiorganisation
94 Subskriptfeld unrichtig
95 Vergleichsoperator fehlt
99 Compilerfehler
101 RECORD KEY ohne Datenerklärung
102 RELATIVE KEY ohne Datenerklärung
103 STATUS ohne Datenerklärung
104 STATUS-Definition nicht standardgemäß
110 Paragraphenname nicht definiert

12. Fehlermeldungen (Übersetzer COBOL B)

Fehler- Beschreibung
code

- 01 COMPILER-Fehler
 02 Datenname zu lang oder falsch oder enthält Kleinbuchstaben
 bei gesetztem ANS-Schalter
 03 Literal fehlt oder falsch
 04 Falsches Zeichen
 05 Datenname mehrfach definiert
 06 Überlauf der Symboltabelle
 (Zu viele Daten- und Prozedurnamen definiert)
 07 Unerlaubtes Zeichen in Spalte 7
 08 COPY geschachtelt oder zu kopierende Datei fehlt
 09 COBOL-Erweiterung verwendet bei gesetztem ANS-Schalter
 10 Grenzen der Bereiche A oder B nicht eingehalten
 21 '.' fehlt
 22 'DIVISION' fehlt
 23 'SECTION' fehlt
 24 'IDENTIFICATION' fehlt
 25 'PROGRAM-ID' fehlt
 26 'AUTHOR' fehlt
 27 'INSTALLATION' fehlt
 28 'DATE-WRITTEN' fehlt
 29 'SECURITY' fehlt
 30 'ENVIRONMENT' fehlt
 31 'CONFIGURATION' fehlt
 32 'SOURCE-COMPUTER' fehlt
 33 MEMORY SIZE-Angabe falsch
 34 'OBJECT-COMPUTER' fehlt
 36 'SPECIAL NAMES' fehlt
 37 SWITCH-Klausel falsch
 38 DECIMAL-POINT-Klausel falsch
 39 CONSOLE-Klausel falsch
 40 Ungültiges Währungszeichen
 41 '.' fehlt
 42 'DIVISION' fehlt
 43 'SECTION' fehlt
 44 'INPUT-OUTPUT' fehlt
 45 'FILE-CONTROL' fehlt
 46 'ASSIGN' fehlt
 47 'SEQUENTIAL' oder 'INDEXED' oder RELATIVE fehlt
 48 'ACCESS' fehlt für indexseq./relative Datei
 49 'SEQUENTIAL/DYNAMIC' fehlt
 50 Unerlaubte Kombination ORGANIZATION/ACCESS/KEY
 51 SELECT-Klausel falsch
 54 Dateiname fehlt
 55 'DATA DIVISION' fehlt
 56 'PROCEDURE DIVISION' fehlt oder unbekannte Anweisung
 61 '.' fehlt
 62 'DIVISION' fehlt
 63 'SECTION' fehlt
 64 SELECT-Klausel für Dateiname fehlt
 65 Datensatzlänge nicht angegeben
 66 Stufenummer nicht zwischen 01 und 49 oder
 Stufe 01 erforderlich
 67 Klausel(n) in der FD-Erklärung falsch

Fehler- code	Beschreibung
68	"WORKING-STORAGE" fehlt
69	"PROCEDURE DIVISION" fehlt oder unbekannte Anweisung
70	Datenerklärung falsch oder "." fehlt
71	SIGN/USAGE unverträglich mit COMP-Datenelem. oder PICTURE ohne Vorzeichen oder andere Unverträglichkeit
72	BLANK unverträglich mit nichtnum. Datenelement
73	PICTURE-Klausel zu lang (numerisch 18, numerisch-druckaufber. 512, alphanumerisch 8192 Zeichen
74	VALUE-Klausel nicht bei Datenelement oder für angeg. Datenkategorie unzulässig
75	"VALUE" falsch oder mit PICTURE-Klausel unverträglich
76	FIGER/SYNCHRONIZED/JUSTIFIED/BLANK nicht bei einem Datenelement verwendet
77	Stufenummernsystem enthält Datengruppe mit Länge 0 oder größer 8192 Bytes
78	REDEFINES für ungleiche Datenfelder oder Stufenummern
79	Datenbereich größer 64K Bytes
80	"DYNAMIC" nur erlaubt ohne ANS-Schalter und auf Stufe 01
81	Fehler in der Datenerklärung
82	REDEFINES-Datenname nicht erklärt
83	USAGE nur mit COMP, DISPLAY oder INDEX erlaubt
84	SIGN muß LEADING oder TRAILING sein
85	Auf SYNCHRONIZED muß LEFT oder RIGHT folgen
86	Auf JUSTIFIED muß RIGHT folgen
87	Auf BLANK muß ZERO folgen
88	OCCURS-Faktor muß numerisch, ungleich 0 und ohne Vorzeichen sein
89	Auf VALUE muß ein Literal (num. oder nichtnum.) oder eine Figurative Konstante folgen
90	PICTURE-Zeichenfolge mit unerlaubtem Zeichen oder mißachtet die Vorrangregeln
91	INDEXED-Datenname fehlt oder bereits erklärt
92	Num.-druckaufber. PICTURE-Zeichenfolge zu lang
101	Unbekanntes COBOL-Verb
102	IF ... ELSE nicht paarweise
103	Falsche Datenkategorie
104	Paragraphenname mehrfach erklärt
105	Paragraphenname und Datenname gleich
106	Name fehlt
107	Falsche Kombination von Datenkategorien
108	Unbedingte Anweisung mit Bedingung verknüpft
109	Falsches Subskript
110	ACCEPT/DISPLAY falsch
111	E/A-Anweisung falsch
116	IF's zu tief geschachtelt
117	Falscher Aufbau der PROCEDURE DIVISION
118	Obligatorisches reserviertes Wort fehlt
119	Subskript außerhalb des zulässigen Bereichs
120	Zwischencodepuffer zu klein (zu großer GO TO ... DEPENDING oder Mehrfach-MOVE)
140	Sprungziel von PERFORM oder GO TO befindet sich in nicht geladenem Segment

Fehler-	Beschreibung
code	

- | | |
|-----|---|
| 142 | Überzähliges IF nach Ende der Quellcodeeingabe
(ELSE oder Punkt fehlt) |
| 143 | Falsche Datenkategorie |
| 144 | Paragraphenname undefiniert |
| 145 | Indexname mehrfach definiert |
| 146 | Falsche Cursorsteuerung |
| 147 | KEY-Erklärung fehlt |
| 148 | STATUS-Erklärung fehlt |
| 149 | STATUS-Erklärung falsch |
| 151 | PROCEDURE DIVISION falsch |
| 152 | USING-Parameter nicht erklärt in der LINKAGE SECTION |
| 153 | USING-Parameter nicht auf Stufe 01 oder 77 erklärt |
| 154 | USING-Parameter in der Parameterliste mehrfach verwendet |
| 156 | E/A-Fehler bei der Verarbeitung einer Segmentdatei |
| 157 | Falscher Aufbau der PROCEDURE DIVISION |
| 160 | Zwischencode-Puffer zu klein (s.Fehler 120) |
-

13. Meldungen Laufzeitsystem COBOL A,B

Warnung des RFM Status-1-Name = 4

- 001 Wiederholungswarnung
Es war mehr als ein Versuch nötig, um Daten auf/von Diskette zu transferieren.
- 002 Überlaufwarnung
Der letzte Überlaufsblock wurde benutzt. Weiteres Erzeugen von Sätzen kann Fehler 102 und Verlust von Sätzen produzieren. Reorganisieren.
- 004 Bereits ausgeführt
Der bezeichnete Satz ist schon gelöscht.

INVALID KEY Bedingungen Status-1-Name = 2

- 008 Leer-Satz (bei Direktdateien)
Der gerade gelesene Satz besteht nur aus Nullen.
- 103 Satz-Verdoppelung
Ein Satz mit dem bezeichneten Identifikator existiert bereits.
- 104 Satz nicht gefunden (bei Index-Dateien)
Der bezeichnete Satz kann in der Datei nicht gefunden werden.
- 105 Satz gelöscht
Der bezeichnete Satz ist als gelöscht gekennzeichnet und kann deshalb nicht gelesen werden.
- 106 Ungültiger Identifikator
Der Identifikator liegt außerhalb des Dateibereichs.

Sonstige Fehler Status-1-Name = 9

- 001 Fehlender Sektor auf der Diskette.
- 002 CRC-Fehler auf der Diskette.
- 003 Zeitfehler während einer READ/WRITE Datenübertragung.
- 004 Fehlende Adreßmarke.
- 005 Timeout bei Diskettenoperation.
- 006 Zeitfehler bei Leseoperation.
- 007 Busy-Bit Timeout.
- 008 Gerät unklar.
- 009 Dateiname nicht im Inhaltsverzeichnis und auch kein Platz dafür vorhanden.
- 010 Dateiname nicht im Inhaltsverzeichnis.
- 012 Falscher Eintrag im Inhaltsverzeichnis.
- 015 Disketten-Belegungsliste nicht auffindbar.
- 016 Ungültiges Diskettenkommando.
- 017 Falsche Gerätenummer.
- 019 Laufwerksnummer nicht 0, 1, 2, 3.
- 022 Versuchte Eingabe aus einer Datei, die keine Eingabedatei ist.
- 023 Leseversuch nach EOF.
- 024 Schreibversuch in eine Datei, die keine Ausgabedatei ist.
- 025 Nichtzugewiesenes Gerät.
- 026 Unbekannter Gerätetyp.

- 027 Dateiname beginnt nicht mit Buchstaben oder Zahl.
028 Dateiname länger als 6 oder Erweiterung länger
als drei Zeichen.
029 Kommando bezog sich nicht auf Diskette im INTEL-Format.
030 Betriebssystem nicht geladen.
032 DELETED-DATA-Satz gefunden.
039 Datei bereits eröffnet.
041 Eine nicht eröffnete Datei soll geschlossen werden.
047 Diskette voll.
050 Ausgabedatei bereits auf Diskette vorhanden.
051 Diskette voll.
052 Dateiname länger als 8 Zeichen.
053 Falscher Disketten-Typ oder Spurnummer 76.
Die Diskette ist nicht im INTEL-Format beschrieben,
oder das INTEL-Format ist zerstört.
054 Sektornummer = 0 oder 26.
056 RFM-Modul im Programm nicht vorhanden.
059 Satz kann nicht gelesen oder geschrieben werden.
096 Der RFM ist mit einer anderen Aufgabe beschäftigt.
Aufruf erneut versuchen.
099 Keine Direktzugriffsdatei.
100 Datei wurde nicht geschlossen, Fehler kann mit RCOVER
beseitigt werden.
101 Ungültiger Dateiname.
102 Überlaufeehler
Im Überlaufbereich ist kein Platz mehr für diesen Satz.
Es können Sätze verloren gegangen sein..
109 Unzulässiger Operationscode.
111 Der RFM hat nicht genügend Speicherplatz um einen weiteren
Dateikontrollblock anzulegen.(Speicherplatz wird durch
Schließen einer Datei freigegeben).
112 Versuch einen durch MEPLA gesperrten Satz zu löschen.
151 Direktzugriff auf sequentielle Datei
152 REWRITE für Datei, die nicht I/O-eröffnet ist
153 Subskript außerhalb des zulässigen Bereichs
154 PERFORM-Schachtelung mit mehr als 16 (22)* Stufen
155 Unerlaubtes Kommando an das Laufzeitsystem
156 Zwischenencode fehlerhaft oder nicht vorhanden
157 Zwischenencode zu groß
158 REWRITE für zeilensequentielle Datei
159 Unzulässiges Format in zeilensequentieller Datei
160 *Fehler beim Laden eines Compiler-Overlays
161 Falscher Zwischenencode
162 Arithmetischer Überlauf
163 AT-Cursorposition außerhalb des Bildschirms
164 Mit CALL gerufenes Programm kann nicht gefunden werden
165 Unverträgliche Versionen von Compiler und Laufzeitsystem
170 Datei nicht eröffnet (RFM)
171 Datei im falschen Modus eröffnet (RFM)
172 Nur 16 Dateien dürfen gleichzeitig eröffnet sein (RFM)
173 Datei bereits eröffnet (RFM)
174 E/A-Anweisung unerlaubt beim gegenwärtigen
Eröffnungsmodus (RFM)
175 Nicht genug Speicher verfügbar zur Eröffnung einer
weiteren sequentiellen Datei
177 Unzulässiger Inhalt eines Relocatable-Moduls

* Nur bei COBOL B



1. Anweisungennicht ausführbare Anweisungen

REM beliebige zeichenfolge
 DECIMAL liste mit bezeichnungen
 STRING [(länge)] liste mit bezeichnungen
 DEF bezeichnung [(liste mit formalen param)] = ausdruck
 DEF\$ bezeichnung [(liste mit formalen param)] = string ausdruck

ausführbare Anweisungen

LET { variable } = { ausdruck }
 { string variable } { string ausdruck }

GO TO zeilennr
 IF vergleich THEN zeilennr
 ON ausdruck GO TO liste mit zeilennr

FOR bezeichnung = ausdruck TO ausdruck STEP ausdruck
 NEXT bezeichnung

GOSUB zeilennr
 RETURN

PROC bezeichnung [(liste mit formalen param)]
 PROCEND

CALL bezeichnung [(liste mit formalen param)]

ENTER hex-adr
 STOP
 EXIT
 END

START dateiname
 OPEN log. dateibezeichnung, dateiname
 CLOSE log. dateibezeichnung
 INPUT log. dateibezeichnung, eingabeliste
 INPUTC log. dateibezeichnung, eingabeliste
 INCHAR log. dateibezeichnung, eingabeliste
 PRINT log. dateibezeichnung, ausgabeliste
 PRECISION ausdruck, variablenliste
 DISPLAY bezeichnung, zeilennr, spaltennr, feldlänge, datentyp

datentyp: 1 = Dezimalzahl (Ziffern, ., +, -)
 2 = Buchstaben (A-Z, a-z, SP)
 4 = Ziffern (0...9)
 8 = alphanum. Zeichen
 16 = automatischer CR

Funktions-name	Parameter	Ergebnis	Beschreibung
ABS	arithm.ausdruck	zahl	Absolutbetrag
INT	arithm.ausdruck	zahl	Größte ganze Zahl die kleiner gleich arithm. Ausdruck ist
SGN	arithm.ausdruck	1, -1, 0	Vorzeichenetest
IDX	string, string	zahl	Zeichenposition des erstmaligen Auftretens des 2.String im 1.String
INC	string, string	string	Verkettungsstring
LEN	string	zahl	Länge des Strings
MSK	druckmaske, arithm.ausdruck	string	Druckaufbereitung Maskenzeichen: Z = Nullenunterdrückung 9 = Ziffer Leerstelle = Einfügen Leerstelle ,
			=Komma (Ziffer trennung) - = Decimalpunkt CR = Vorzeichenposition =2 Vorzeichenstellen am Maskenende

2. Standardfunktionen

Funktions-name	Parameter	Ergebnis	Beschreibung
STR	arithm.ausdruck	string	Umwandlung Zahl → String
SUB	string, arithm. ausdruck, arithm.ausdruck	string	Unterstring der beim 1. arithm. Ausdruck beginnt und die Länge des 2. arithm. Ausdrucks hat
VAL	string	zahl	Umwandlung String → Zahl
BIN	arithm.ausdruck	string	ASCII-Zeichen dessen dezimaler CODE der arithm. Ausdruck ist
CRLF	-	Steuerzeichen	Wagenrücklauf, Zeilenvorschub
QUOTE	-	Steuerzeichen	Apostroph
TAB	-	Steuerzeichen	horizontaler Tabulator (hardwareabhängig)
ATT	-	string	Bildschirmattribute: 65 = invers 48 = blinkend 32 = halbe Intensität 81 = unterstrichen 97 = unsichtbar 113 = normal



Funktions-name	Parameter	Ergebnis	Beschreibung
CRL	-	Steuerzeichen	Bildschirm löschen
LOC	arithm.ausdruck, arithm.ausdruck	Steuerzeichen	Cursorpositionierung
NOR	-	Steuerzeichen	Setzt Bildschirm in Normalmodus
UND	-	Steuerzeichen	Setzt Bildschirm in Unter- streichungsmodus
ASCII	string	zahl	dezimale Darstellung des 1.Zeichens im String
NEW	variable	-	Anforderung der Variablen
FIXED	string variable, arithm.ausdruck	-	Zuweisung von arithm. Ausdruck Zeichen, an die Stringvariable
SKIP	arithm.ausdruck	-	Ignorierung von arithm. Ausdruck Zeichen in Inputanweisung

3 . Befehle

ABORT

BYE

C [CONTINUE] [liste mit zeilennummern]

DELETE zeilennummer [,zeilennummer]

GET dateiname

KILL dateiname

LIST [gerät] [,zeilennummer][,zeilennummer]

NEW [hex Adr] (nicht kleiner als 9D18h)

REPLACE dateiname

RUN [liste mit zeilennummern]

SAVE dateiname

SIZE [ganzzahlige konstante]

SIZE\$ [ganzzahlige konstante]

START dateiname

4. Schnittstelle C-BASIC-AssemblerFORMAT:

ENTER <hex.adr>

Beispiel

BASIC-Programm	Assembler-Programm
<u>Aufruf</u> . . . CALL proc (a\$,d) . . . PROC proc (str\$,dez) ENTER 4000 PROCEND	<ul style="list-style-type: none"> <u>Bedingungen</u> <p>1) ORG 4000 H JMP ... (<u>erster Befehl</u>)</p> <p>2) Programm im Speicher</p> <p>3) Rücksprung ins BASIC-Programm: CY=0</p> <ul style="list-style-type: none"> <u>Erhaltene Parameter</u> <p>H/L = Zeiger auf Parameterblock mit Zeigern auf die Attribut-Vektoren der Prozedur-Parameter</p> <p>D/E = Zeiger auf Interpreterroutinen-Liste.</p>

Tabellen-position	Routine	Bedeutung	Eingabe-Parameter	Ergebnis
0	TAVDV	Test des Gültigkeitsbits	H/L = Adresse des Attributvektors	CY = 0 Gültigkeitsbit gesetzt CY = 1 A=Fehlercode
1	XAVTP	Hole Typbyte	H/L = Adresse des Attributvektors	A = Typbyte des Attributvektors
2	XAVDP	Hole Datenzeiger	H/L = Adresse des Attributvektors	D/E = Datenzeigerwert des Attributvektors
3	XAVPR	Hole Genauigkeitsbyte	H/L = Adresse des Attributvektors	A = Genauigkeitsbyte des Attributvektors
4	XAVSZ	Hole Längenbyte (SIZE)	H/L = Adresse des Attributvektors	A = Längenbyte des Attributvektors
5	GPTST	Hole Zeiger auf String-datenbereich	H/L = Adresse des Attributvektors	D/E = Adresse des ersten Bytes des Stringdatenbereichs
6	SAVVD	Setze Gültigkeitsbit	H/L = Adresse des Attributvektors	Standard
15	CVID	Konvertiert eine ganze Zahl in eine Dezimalzahl	H/L = Adresse des Attributvektors des dezimalen Ergebnis D/E = Zu konvertierender Binärwert	CY = 0 Operation erfolgreich
16	CVDI	Konvertiert eine Dezimalzahl in eine ganze Zahl	H/L = Adresse des Attributvektors der Dezimalzahl	CY = 0 H/L = Binärwert CY = 1 A = Fehlercode H/L = 0

Tabellen-position	Routine	Bedeutung	Eingabe-Parameter	Ergebnis
17	TAKE	Fordert Bereich im dynamischen Speicher	A = benötigte Größe in Byte (1-255)	CY = 0 H/L = Adresse 1. Byte des Speichers CY = 1, A = Fehlercode H/L = 0
18	GIVE	Gibt einen Bereich des des dynamischen Speichers frei	H/L = Adresse des 1. Byte des freizugebenden Speicherbereichs	HL = 0 CY = 0
19	CBFSP	Kreiert Pufferbereich im Stack	H/L = Anzahl der anzulegenden Bytes	CY = 0 H/L = Adresse des niedrigstwertigen Byte des Stacks Stackpointer steht auf H/L-2. Oberes Stackende alter Wert von HL CY = 1 A = Fehlercode
20	RBFSP	Gibt durch CBFSP angelegten Stack frei	SP besetzt wie beim Ende von CBFSP	SP besetzt wie beim Eintritt in CBFSP

5. Fehlermeldungen

Schlüssel Bedeutung

83 Aufgerufene Routine nicht geladen
84 ASCII-Argument 0
86 INCHAR - nur einfache Stringvariable sind erlaubt
87 INPUT - unzulässiges Stringformat
89 MSK - keine ganze Zahl mehr
8A Unzulässige Maske
8B SUB - Startposition nicht im zulässigen Bereich
8C Division durch Null
8D Benutzerunterbrechung (CTRL/C)
8E FIXED - unzulässiger String
8F Unzulässiger Gebrauch von SKIP/FIXED

90 Falsche Parameteranzahl im Funktionsaufruf
91 ENTER - Routine muß mit JMP beginnen
92 NEW - falscher Parameter
93 Quelle kein String
96 EXIT - kein PROC aktiv
97 CALL - zu viele Parameter
98 CALL - zu wenig Parameter
99 CALL - PROC läuft bereits (keine Rekursion!)
9A Funktionsparameter hat falschen Typ
9B Keine Rekursion von Funktionen erlaubt
9C (Intern) Datei kann nicht geschlossen werden
9D NEW - Parameter keine DISPLAY Variable
9E FIXED - Satzende während der Eingabe
9F FIXED - Schlußapostroph fehlt (1. Zeichen war Apostroph!)

A0 FIXED - Länge größer als max. Stringlänge
A1 FIXED - Stringlänge zu groß
A2 Quelldaten nicht dezimal
A3 (Intern) Ende von CI
A4 VAL - String stellt keine Zahl dar
A5 Logische Datei nicht eröffnet
A6 CLOSE - logische Datei war nicht eröffnet
A7 OPEN - logische Datei bereits eröffnet
A8 DISPLAY - unzulässige Cursorposition
A9 DISPLAY - Parameter außerhalb der zulässigen Grenzen

AA DISPLAY - Variable muß vom Typ String sein
AB LET - unzulässige linke Seite der Zuweisung
AC Funktion benötigt Parameter / Feld nicht indiziert
AD Variable nicht besetzt
AE IF - unzulässiger Vergleich
AF IF - Vergleich fehlt

Schlüssel Bedeutung

B0 IF - Vergleich zwischen unterschiedlichen Typen
 B1 NEXT - arithmetischer Überlauf
 B2 FOR / NEXT - Schachtelungsfehler
 B3 NEXT - kein zugehöriges FOR
 B4 LET - linke Seite falscher Datentyp
 B5 RETURN - kein zugehöriges GOSUB
 B6 Unbekannte log. Dateibezeichnung
 B7 Log. Dateibezeichnung fehlt
 B8 Wert muß zwischen - 32767 und + 32767 liegen
 B9 LOC - Unzulässige Cursorposition
 BA STOP (kein Fehler)
 BB LEN - Stringparameter erforderlich
 BC BIN - unzulässiger Wert
 BD ATT - unzulässiger Attributwert
 BE ABS - Parameter keine Dezimalzahl
 BF Undefiniertes Symbol

 C0 SIZE (Z) - falsche Länge (1 L 255)
 C1 PRECISION - Feld bereits besetzt
 C2 PRECISION - Variable hat falschen Typ
 C3 PRECISION - falsche Dezimalstellenzahl
 (0 bis SIZE)
 C4 Funktionsparameter hat falschen Typ
 C5 Unterbrechungsstelle (kein Fehler)
 C6 Variable mehrfach definiert
 C7 SUB - Startpunkt nicht im zulässigen Bereich
 C8 Feldindex max. 32767
 C9 Variable bereits mit anderem Datentyp deklariert
 CA STRING - max. Länge ist 255
 CB STRING / DECIMAL - falsche Länge
 CC Falsches Binärliteral
 CD Operand hat falschen Typ
 CE INPUT - falsches Zahlenformat
 CF Anweisung an falscher Stelle

 D0 PROCEND fehlt
 D1 Anweisung nach PROC-PROCEND unzulässig
 D2 END fehlt
 D3 Speicherüberlauf - SAVE und Neustart
 D4 PROC Name mehrfach definiert
 D5 Indizierte Variable ist kein Feld
 D6 Komma fehlt
 D7 Unzulässiges Symbol im Ausdruck
 D8 Zuwenig Operatoren im Ausdruck
 D9 Zuwenig Operanden im Ausdruck
 DA falscher Feldindex
 DB Klammerfehler im Ausdruck
 DC LIST - unzulässige Zeilengrenzen
 DD LIST - unzulässiges Zeichen
 DE INPUT - max. Länge ist 80 Zeichen
 DF Falsches Schlüsselwort

Schlüssel Bedeutung

- E0 Schlüsselwort fehlt
E1 Dezimalkonstante fehlt
E2 Benutzersymbol erwartet
E3 Text hinter Anweisungsende
E4 Syntaktischer Fehler
E5 Arithmetischer Überlauf bei Multiplikation
E6 (Intern) Operation noch nicht vorhanden
E7 Keine Zeilenänderung vor ABORT
E8 (Intern) end-of-extent
E9 Unzulässiges Kommando
EA Variable kein Feld oder keine Funktion
EB Unbekannte Variable im Ausdruck
EC Kommando in diesem Zusammenhang unzulässig
ED Arithmetischer Überlauf bei Addition
EE Falsche Anzahl von Feldindizes
EF Falscher Feldindex
- F0 Unzulässiger Operand in der Quelle
F1 Unzulässiger Ausdruck
F2 Zeile existiert nicht
F3 Zeile außerhalb des Bereichs
F4 Unzulässige Zeilennummer
F5 Unbekanntes Schlüsselwort
F6 Variable unbesetzt
F7 Keine ganze Zahl mehr
F8 String zu kurz für empfangene Datei
F9 Unzulässiger Variablenname
FA Variablenname zu lang (6 Zeichen)
FB Unzulässiges Zeichen in Dezimalkonstante
FC Dynamischer Speicher voll
FD Schlußapostroph fehlt
FE Nicht-darstellbares Zeichen im String
FF Vorzeitiges Textende

3



1. Darstellung von Konstanten und Variablen

Stringkonstante:	Eine in Anführungszeichen eingeschlossene Folge von max. 255 alphanumerischen Zeichen. z.B. "SCIENTIFIC BASIC V.1.22"
Ganzzahlige Konstante:	Ganze Zahlen zwischen -32768 und + 32767. z.B. -3517
Festpunkt-Konstante:	Positive und negative reelle Zahlen. z.B. 34.789
Gleitpunkt-Konstante:	Positive und negative Zahlen in Exponential-darstellung z.B. 0.2524E-4
Hexadezimale-Konstante:	Hexadezimale Zahlen mit vorangestelltem &H. z.B. &H7AF
Oktal-Konstante:	Oktalzahlen mit vorangestelltem &0 oder &.z.B. &1457

Eine Konstante mit einfacher Genauigkeit ist jede numerische Konstante, die 1. sieben oder weniger Ziffern hat,
2. die Exponentialdarstellung mit E benutzt,
3. ein Ausrufezeichen am Ende hat.

Eine Konstante mit doppelter Genauigkeit ist jede numerische Konstante, die 1. acht oder mehr Ziffern hat,
2. die Exponentialdarstellung mit D benutzt,
3. ein Nummernzeichen am Ende hat.

Variablennamen können beliebige Längen haben, die ersten 40 Stellen sind von Bedeutung. Sie bestehen aus Buchstaben, Ziffern und dem Dezimalpunkt, das erste Zeichen muß ein Buchstabe sein. Kleinbuchstaben werden in Großbuchstaben umgewandelt.

Stringvariable:	nachgestelltes Dollarzeichen
ganzzahlige Variable:	nachgestelltes Prozentzeichen
Variable einfache Genauigkeit:	nachgestelltes Ausrufezeichen
Variable doppelte Genauigkeit:	nachgestelltes Nummerzeichen

Wird kein Typdeklarationszeichen angegeben, so wird einfache Genauigkeit angenommen.

2. Operatoren

Arithmetische

Operatoren: Operator Operation Beispiel

$^$	Exponentiation	3^2	=9
-	Negation	-3	
*	Multiplikation	$3*2$	=6
/	Division	$3/2$	=1.5
\	Ganzzahlige Division	$3\backslash 2$	=1
MOD	Modulo Arithmetik	$8 \text{ MOD } 3$	=2
+	Addition	$3+2$	=5
-	Subtraktion	$3-2$	=1

Vergleichs-
operatoren:

=	Gleichheit	X=Y
\neq	Ungleichheit	X \neq Y
<	Kleiner als	X<Y
>	Größer als	X>Y
\leq	Kleiner-gleich	X \leq Y
\geq	Größer-gleich	X \geq Y

logische

Operatoren: Operator Wirkung

	X	NOT X	
NOT	1	0	
	0	1	
AND	X	Y	X AND Y
	1	1	1
	1	0	0
	0	1	0
	0	0	0
OR	X	X	X OR Y
	1	1	1
	1	0	1
	0	1	1
	0	0	0
XOR	X	Y	X XOR Y
	1	1	0
	1	0	1
	0	1	1
	0	0	0
IMP	X	Y	X IMP Y
	1	1	1
	1	0	0
	0	1	1
	0	0	1
EQV	X	X	X EQV Y
	1	1	1
	1	0	0
	0	1	0
	0	0	1

String-

Operatoren: Strings können mit + verkettet werden.
z.B. "Datei" + "name" = "Dateiname"

3. Kommandos und Anweisungen

ANWEISUNG	ODER KOMMANDO BEISPIEL	FUNKTION
AUTO	[zeilennr][, schrittweite] AUTO 100,50	Automatisches Erzeugen von Zeilennummern (Standard AUTO 10,10)
CALL	variable [(param.liste)] CALL ASSUM(J,K,L)	Aufruf von Assemblerunterprogrammen
CHAIN	[MERGE] dateiname [,zeilennr][,ALL] ,DELETE bereich CHAIN MERGE "PROG",100,ALL,DELETE 5-6	Aufruf von anderen Programmen mit Übergabe von Variablen des aktuellen Programms
CLEAR	[[, ausdruck1][, ausdruck2]] CLEAR &HAAF6,514	Vorbesetzen von Variablen mit Null bzw. Leerstring ausdruck1 = Höchste Speicheradr. für SBASIC ausdruck2 = Stackgröße für SBASIC
CLOSE	[[#] dateinr], [#] dateinr]...]] CLOSE #1,#2,#3	Schließen von Dateien
COMMON	variablenliste COMMON A,B,H,P	Übergabe von Variablen an mit CHAIN geladenen Programmen
CONT		Fortsetzen der Programmausführung nach CTRL-C, STOP oder END
DATA	konstantenliste DATA 3, "MILCH" ,4,"BUTTER"	Speicherung von Konstanten, auf die die READ-Anweisung zugreift
DEF	FNname [(param)] =ausdruck DEF FNAB (X,Y)=X*3/Y-6	Definition und Bezeichnung einer Benutzerfunktion

ANWEISUNG BEISPIEL	ODER KOMMANDO	FUNKTION
DEF typ buchstabenbereich DEFSTR A-D Alle Variablenamen die mit den Buchstaben A,B,C oder D beginnen sind Stringvariable		Zuordnen von Variablenamen zu den Typen ganze Zahlen (INT), Gleitpunktzahlen einfacher (SNG) oder doppelter (DBL) Genauigkeit, bzw. String (STR)
DEF USR[ziffer]=ganzzahlalausdruck DEF USR0=2400		Festlegen der Startadresse eines Assemblerunterprogramms
DELETE [zeilennr] [- zeilennr] DELETE -400		Löschen von Programmzeilen
DIM liste indizierter variablen DIM A(10) ,B(3,5)		Festlegung der Maximalwerte von Feldvariablen-Indices
EDIT zeilennr EDIT 20 Unterkommandos: I text X iD H iSzeichen iKzeichen Czeichen CR,E,Q, L A	Einfügen eines Texts Verlängern der Zeile Löschen von i Zeichen rechts vom Cursor Löschen aller Zeichen rechts vom Cursor Sucht das i-te Auftreten des Zeichens Wie S außer daß alle gesuchten Zeichen gelöscht werden Ersetzen von dem Zeichen auf dem der Cursor steht Verlassen des Editiermodus mit/ohne Abspeicherung der Änderung Ausgabe des Restes der Zeile Neubeginn des Editierens	Aufruf des Editiermodus
END		Beendigung der Programmausführung

ANWEISUNG	ODER KOMMANDO BEISPIEL	FUNKTION
ERASE	variablenliste ERASE A,NUMMER,G	Löschen von Feldern aus einem Programm
ERROR	ganzzahlalausdruck ERROR S+T	Simulieren von Fehlersituationen Definition von Fehlerschlüsseln Die Variable ERR enthält den Fehlerschlüssel, ERL die Zeilennr. in der der Fehler auftrat
FOR NEXT	variable TO ausdruck [STEP ausdruck] variable [,variable ...] FOR I = 1 TO 5 NEXT I	Schleifenbildung
GOSUB RETURN	zeilennr	Aufruf und Rückkehr von/aus Unterprogrammen
GOTO	zeilennr	Unbedingtes Verzweigen
IF	ausdruck { THEN anweisung... } GOTO zeilennr ELSE anweisung... zeilennr IF A < B THEN A= A+5 ELSE 330	Bedingtes Verzweigen bzw. Steuerung des Programmfluss (Schachtelungen sind erlaubt)
INPUT	[;][anforderungsstring;] variablenliste INPUT "Name:";NAME\$	Anforderung von Eingaben über Tastatur
INPUT	dateinr,variablenliste INPUT #1,A,B,D,S\$	Lesen von Datenelementen aus Dateien
KILL	"dateiname"	Löschen von Diskettendateien
LINE INPUT	[;][string; stringvariable] LINE INPUT ADRESSE\$	Eingabe einer Zeile, max. 254 Zeichen in eine Stringvariable

ANWEISUNG ODER KOMMANDO BEISPIEL	FUNKTION
LINE INPUT #dateinr, stringvariable	Einlesen aus Dateien
LIST [zeilennr] [- zeilennr] LIST - 3000	Auflisten von Programmzeilen
LOAD "dateiname" [,R] LOAD "PROG" ,R	Laden eines Programms bei Angabe von R Start mit RUN
MERGE "dateiname"	Mischen von Programmdateien
MID\$ (string1,ganzzahl [,ganzzahl])=string2 MID\$ ("GARTENSCHLAUCH" ,7)="ZAUN"	Ersetzen von Stringteilen durch andere
NAME "alter dateiname" AS "neuer dateiname"	Umbenennung einer Diskettendatei
ON ERROR GOTO zeilennr ON ERROR GOTO 400	Abfangen von Fehlern und Festlegen der Fehlerbehandlungsroutine
ON ausdruck GOSUB liste mit zeilennr ON NUMMER-1 GOTO 300,310,320	Unterprogrammaufruf in Abhängigkeit vom Wert des Ausdruckes
ON ausdruck GOTO liste mit zeilennr ON NUMMER-1 GOTO 300,310,320	Programmverzweigung in Abhängigkeit vom Wert des Ausdruckes
OPEN "modus"[,#] dateinr, "dateiname" OPEN "0" ,1, ":lp:"	Eröffnen einer Datei Modus I=Input;O=Output
OPTION BASE 1 0	Festlegung der Untergrenze von Feldindices
OUT i,j i=Portnr. OUT 32,101 j=Datenbyte	Senden eines Bytes an einen Ausport des Rechners.
POKE i,j i=Speicherzelle POKE &H27BB,06 j=Datenbyte	Schreiben eines Bytes in eine Speicherzelle

ANWEISUNG	ODER KOMMANDO BEISPIEL	FUNKTION
PRINT	liste mit ausdrücken ? "SCIENTIFIC BASIC", A	Datenausgabe auf Bildschirm Statt PRINT kann auch ein Fragezeichen angegeben werden.
PRINT	USING format; liste mit ausdrücken PRINT USING "## .## ";A;B	Ausgabe von Strings und Zahlen unter Verwendung des angegebenen Formates
	Formatzeichen für Stringausdrücke: ! Nur das erste Zeichen wird ausgegeben \n SP\ 2+n Zeichen des Strings werden ausgegeben & Der String wird in seiner ursprünglichen Form ausgegeben	
	Formatzeichen für numerische Ausdrücke: # Zifferposition . Dezimalpunktposition + Vorzeichen wird immer ausgegeben - Vorzeichen wird nur ausgegeben wenn negativ ** Statt führender Nullen Ausgabe von Sternchen \$ Links von der Zahl wird ein Dollarzeichen ausgegeben ~~~ Legt Exponentialformat fest % Falls Anzahl Ziffern > Anzahl Stellen wird Prozentzeichen vor der Zahl ausgegeben.	
PRINT#	dateinr [USING format;] ausdrucksliste PRINT# 1,A,B,A\$	Ausgabe von Daten in Dateien siehe auch PRINT USING
RANDOMIZE	[ausdruck] RANDOMIZE PEEK (&h27bb)	Neustart des Zufallszahlen-generators
READ	variablenliste READ A,B\$,C,D\$	Lesen von Werten aus der DATA-Anweisung
REM	komentar	Einfügen von erläuternden Kommentaren
RENUM	[neue nr][,alte nr][,schrittweite] RENUM 200,50	Neunummerierung von Programmzeilen (Standard RENUM 10,10)

ANWEISUNG ODER KOMMANDO BEISPIEL	FUNKTION
RESTORE [zeilennr] RESTORE	Lesewiederholung für DATA-Anweisung
Resume [0 NEXT zeilennr]	Wiederaufnahme der Programmausführung nach einer Fehlerbehandlungsprozedur
RUN [zeilennr]	Programmstart
SAVE "dateiname" [,A ,P] A=ASCII-Format P=geschützt	Abspeichern eines Programms auf Diskette
STOP	Beenden der Programmausführung
SWAP variable1, variable2	Vertauschen des Variableninhalts
SYSTEM	Rückkehr zum Betriebssystem
TRON TROFF	An- und Abschalten der Verfolgung von Programmzeilen durch Ausgabe deren Nummern
WAIT portnr,ganzzahl [,ganzzahl] WAIT 32,2	Warten auf Zeichen am Eingabe-port des Rechners
WHILE bedingung . .	Ausführen einer Schleife solange die Bedingung wahr ist.
WEND	
WIDTH ganzzahlausdruck	Bestimmung der Breite der Bildschirmzeilen (15-80)
WRITE liste mit ausdrücken WRITE # dateinr,liste mit ausdrücken	Wirkung wie PRINT, bzw. PRINT# Strings werden jedoch in Anführungszeichen ausgegeben.

4. FunktionenStandardfunktionen

Funktion	Wirkung
ABS(X)	Absolutbetrag von X
ASC(X\$)	Dezimaler Wert, der dem ASCII-Code des ersten Zeichen in X\$ entspricht
ATN(X)	Arcustangens von X im Bogenmaß
CDBL(X)	Wandelt X in eine Zahl mit doppelter Genauigkeit um
CHRS(I)	String der den ASCII-Code von I hat
CINT(X)	Wandelt X durch Rundung in ganze Zahl um
COS(X)	Cosinus von X im Bogenmaß
CSNG(X)	Wandelt X in eine Zahl mit einfacher Genauigkeit um
CVI(2-byte-string)	Wandelt Stringwerte in numerische Werte um
CVS(4-byte-string)	
CVD(8-byte-string)	
EOF(dateinr)	Erkennung von EOF
EXP(X)	Berechnet "e hoch X"
FIX(X)	Ganzzahliger Teil von X
FRE(0)	Berechnet die Anzahl der von SBASIC nicht belegten Bytes
HEX\$(X).	String mit hexadezimalen Wert von X
INP(I)	Liefert das vom Port gelieferte Byte
INPUT\$(X [, # dateinr])	Liefert String mit X Zeichen die von der Tastatur bzw. aus der Datei gelesen werden
INSTR([I,] X\$,YS)	Sucht ab der I-ten Position das erste Auftreten von Y\$ IN X\$; Ergebnis: Position
INT(X)	Größte ganze Zahl die kleiner gleich X ist
LEFT\$(X\$,I)	String der aus den linken I Zeichen von X\$ besteht
LEN(X\$)	Anzahl Zeichen von X\$

<u>Funktion</u>	<u>Wirkung</u>
LOC(dateinr)	Satznummer die beim nächsten GET / PUT verwendet wird
LOG(X)	Natürlicher Logarithmus von X
MID\$(X\$; [,J])	String der Länge I, entnommen aus X\$ ab dem I-ten Zeichen
MKIS(X)	Umwandlung numerischer Werte in Stringwerte
MKS\$(Y)	X=Ganzzahliger Ausdruck
MKD\$(Z)	Y=Ausdruck mit einfacher Genauigkeit Z=Ausdruck mit doppelter Genauigkeit
OCT\$(X)	String mit oktalem Wert von X
PEEK(I)	Liefert das Byte aus Speicherzelle I
POS(0)	Liefert die aktuelle Cursorposition
RIGHT\$(X\$,I)	String der aus den rechten I Zeichen von X\$ besteht
RND [(X)]	Zufallszahl zwischen 0 und 1; Bei X=0 wird die zuletzt erzeugte Zahl wiederholt
SGN(X)	Vorzeichentest
SIN(X)	Sinus von X im Bogenmaß
SPACES\$(X)	String bestehend aus X Leerzeichen
SPC(I)	Nur bei PRINT; Gibt I Leerzeichen aus
SQR(X)	Quadratwurzel von X
STR\$(X)	Stringdarstellung von X
STRING\$(I,J)	String der Länge I, dessen Zeichen den ASCII-Code J haben oder identisch mit dem ersten Zeichen von X\$ sind
STRING\$(I,X\$)	
TAB(I)	Setzt Cursor auf die I-te Bildschirmposition
TAN(X)	Tangens von X im Bogenmaß
USR[ziffer](X)	Ruft Assemblerunterprogramm mit dem Parameter X auf
VAL(X\$)	Numerischer Wert von X\$
VARPTR(variable)	Adresse des ersten Byte der Variablen

Abgeleitete mathematische Funktionen

<u>Funktion</u>	<u>Berechnung in SBASIC</u>
Sekans	1/COS(X)
Kosekans	1/SIN(X)
Kotangens	1/TAN(X)
Arkussinus	ATN(X/SQR(-X*X+1))
Arkuskosinus	ATN(X/SQR(-X*X+1))+1.5708
Arkussekans	ATN(X/SQR(X*X-1))+SGN(SGN(X)-1)*1.5708
Arkuskosekans	ATN(X/SQR(X*X-1))*SGN(X)-1)*1.5708
Arkuskotangens	ATN(X)+1.5708
Hyperbelsinus	(EXP(X)-EXP(-X))/2
Hyperbelkosinus	(EXP(X)+EXP(-X))/2
Hyperbeltangens	EXP(-X)/EXP(X)+EXP(-X)*2+1
Hyperbelkotangens	EXP(-X)/EXP(X)-EXP(-X)*2+1
Hyperbelsekans	2/ EXP(X)+EXP(-X))
Hyperbelkosekans	2/ EXP(X)-EXP(-X))
Arkushyperbelsinus	LOG(X+SQR(X*X+1))
Arkushyperbelkosinus	LOG(X+SQR(X*X-1))
Arkushyperbeltangens	LOG((1+X)/(1-X))/2
Arkushyperbelkotangens	LOG((X+1)/(X-1))/2
Arkushyperbelsekans	LOG((SQR(-X*X+1)+1)/X)
Arkushyperbelkosekans	LOG((SGN(X)*SQR(X*X+1)+1)/X)

5. Schnittstelle SBASIC - ASSEMBLERFormat

CALL variablenname [parameter liste]
 oder
 variablenname = USR[ziffer](param)

Beispiel:

SBASIC PROGRAMM	ASSEMBLER PROGRAMM
<pre> 10 CLEAR,&HCFFF . . 50 ASSUNT = &HD001 60 I! = 10! 70 J = 3 80K# = 4.75# 90 CALL ASSUNT(I!,J,K#) . . 110 DEF USR3 = &HD001 120 A = USR3 (I) .</pre>	<pre> ORG D000 RET START:..... . . Das Assemblerprogramm muß vor dem SBASIC- Programm geladen werden, und in einem vor SBASIC geschützten Bereich liegen. </pre>

Übergabe der ParameterBeim Funktionsaufruf USR:

Akku = Typenkennzeichen des Parameters:

2 = Ganze Zahl in 2 Byte

3 = String

4 = Gleitpunktzahl mit einfacher Genauigkeit(im Microsoft-Format)

8 = Gleitpunktzahl mit doppelter Genauigkeit(im Microsoft-Format)

Ganze Zahlen:

H/L = Adresse von Byte FAC-3

Gleitpunktzahlen mit einfacher Genauigkeit (Microsoftformat):

H/L = Adresse von Byte FAC-3

Gleitpunktzahlen mit doppelter Genauigkeit (Microsoftformat):

H/L = Adresse von Byte FAC-3

Strings:

D/E = Adresse von Byte SDK-0

Beim Aufruf der CALL-Anweisung:

Werden bis zu 3 Parameter übergeben so steht in:

H/L = Adresse von FAC-3 oder FAC-7 oder SDK-0 des 1.Parameters

D/E = Adresse von FAC-3 oder FAC-7 oder SDK-0 des 2.Parameters

B/C = Adresse von FAC-3 oder FAC-7 oder SDK-0 des 3.Parameters

Werden 4 und mehr Parameter übergeben so steht in:

H/L = Adresse von FAC-3 oder FAC-7 oder SDK-0 des 1.Parameters

D/E = Adresse von FAC-3 oder FAC-7 oder SDK-0 des 2.Parameters

B/C = Adresse eines Adreßblocks der weiteren Parameter.

Aufbau Adreßblock:

Byte 0 } Adresse von FAC-3 oder FAC-7 oder SDK-0 des

3.Parameters

Byte 1 } (bytevertauscht)

Byte n } Adresse von FAC-3 oder FAC-7 oder SDK-0 des

n+3.Parameters

Byte n+1 } (bytevertauscht) (n=ganze,gerade Zahl)

Aufbau des Gleitpunktakkumulators FAC

8-stelliger Speicherbereich mit folgender Bezeichnung:

FAC-7 FAC-6 FAC-5 FAC-4 FAC-3 FAC-2 FAC-1 FAC-0

Bei ganzen Zahlen enthält die Stelle FAC-3 das niedrige Wertige Byte, und die Stelle FAC-2 das höhere Wertige Byte der Zahl.

Bei Gleitpunktzahlen mit einfacher Genauigkeit enthalten die Stellen FAC-3 bis FAC-1 die Mantisse (untere 8 Bits in FAC-3) und die Stelle FAC-0 den Exponenten zur Basis 2 plus 129.

Bei Gleitpunktzahlen mit doppelter Genauigkeit enthalten die Stellen FAC-7 bis FAC-1 die Mantisse (untere 8 Bits in FAC-7) und die Stelle FAC-0 den Exponenten zur Basis 2 plus 129.

Das obere Byte der Mantisse (FAC-1) enthält in beiden Fällen im Bit 2 hoch 7 das Vorzeichen (0=positiv, 1=negativ).

Aufbau Stringdeskriptor SDK

3-stelliger Speicherbereich mit folgender Bezeichnung:

SDK-0 SDK-1 SDK-2

SDK-0 enthält die Länge des Strings. SDK-1 und SDK-2 enthalten bytewertauscht die Adresse des Stringanfangs, also des ersten Bytes des Strings.

6. Fehlermeldungen

<u>Code</u>	<u>Nummer</u>	<u>Meldung und Bedeutung</u>
NF	1	NEXT without FOR (NEXT ohne FOR)
SN	2	Syntax error (Syntaxfehler)
RG	3	RETURN without GOSUB (RETURN ohne GOSUB)
OD	4	Out of DATA (Zu wenig Daten)
FC	5	Illegal function call (Unzulässiger Funktionsaufruf)
OV	6	Overflow (Ergebnis Rechenoperation zu groß)
OM	7	Out of memory (Zu wenig Speicher vorhanden)
UL	8	Undefined line (Undefinierte Zeile)
BS	9	Subscript out of range (Indexbereichüberschreitung)
DD	10	Duplicate Definition (Mehrfachdefinition)
/0	11	Division by zero (Division durch Null)
ID	12	Illegal direct (Unzulässige Anweisung im Direktmodus)
TM	13	Type mismatch (VariablenTyp stimmt nicht überein)
OS	14	Out of string space (Stringbereich zu klein)
LS	15	String too long (String zu lang)
ST	16	String formula too complex (Stringformel zu kompliziert)
CN	17	Can't continue (Fortsetzung nicht möglich)
LIF	18	Undefined user function (Undefinierte Benutzerfunktion)
	19	No RESUME (Kein RESUME vorhanden)
	20	RESUME without error (RESUME ohne Fehlerauffangroutine)
	21	Unprintable error (Keine Fehlermeldung vorhanden)
	22	Missing operand (Operand fehlt)
	23	Line buffer overflow (Überlauf Zeilenpuffer)
	24	FOR without NEXT (FOR ohne NEXT)
	29	WHILE without WEND (WHILE ohne WEND)
	29	WEND without WHILE (WEND ohne WHILE)
	50	Field overflow (Feldüberlauf)
	51	Internal error (Interner Fehler)
	52	Bad filenumber (Unzulässige Dateinummer)
	53	File not found (Dateiname nicht im Inhaltsverzeichnis)
	57	Disk I/O error (Fehler bei Ein-/Ausgabeoperation)
	58	File already exists (Datei bereits vorhanden)
	59	Disk full (Diskette voll)
	62	Input past end (Dateiende erreicht EOF)
	64	Bad filename (Unzulässiger Dateiname)
	66	Direct statement in file (Direkte Anweisung in der Datei)
	67	Too many files (Inhaltsverzeichnis voll oder mehr als 6 Dateien geöffnet)
	68	Operation not supported (Kommando oder Anweisung nicht implementiert)
	69	Not enough memory (Nicht genügend Speicher verfügbar)

KAPITEL 1 RFM (INDEXSEQUENTIELLES ZUGRIFFSSYSTEM) Seite

1. Benutzerschnittstelle	4-1-1
2. Operations-Code	4-1-2
3. Status-Code	4-1-3
4. Fehlercodes	.
5. Aufbau der Dateien	4-1-4
5.1 Indexsequentielle Dateien	.
5.2 Direkte Dateien	4-1-6
6. Dienstprogramme des RFM	4-1-7
7. RCOVER-Meldungen	4-1-8

KAPITEL 2 MEPLA

1. Inbetriebnahme	4-2-1
2. MEPLA-Programme	4-2-2
3. Hardwarevoraussetzungen	4-2-3
4. Fehlermeldungen	4-2-4

KAPITEL 3 LADBARER ZEICHENGENERATOR LZG

1. LZG Inbetriebnahme	4-3-1
2. Beschreibung der Funktionstasten für das TCGS	4-3-2
3. Hardwarevoraussetzungen	4-3-3

KAPITEL 4 FORMULARSPRACHE 2

1. Formularerstellungssystem	4-4-1
2. Datenerfassungssystem	4-4-3
3. Beschreibung der Funktionstasten	4-4-5
4. Aufbau der Dateien	4-4-6
5. Fehlermeldungen	4-4-7/8

1. BenutzerschnittstellePUBLIC-Anforderungen

RFMDLA
RFMDHA
RFMW
RFMWB
RFMSY

EXTERNAL-Definitionen

RFM
RFMIN

Parameterblock

1. (OCB) Operations-Code (1 Byte)
2. (SCP) Status-Code-Zeiger (1 Wort)
3. (ECP) Fehler-Code-Zeiger (1 Wort)
4. (FNP) Datei-Nummern-Zeiger (1 Wort)
5. (UBP) Benutzer-Puffer-Zeiger (1 Wort)
6. (IP) Identifikations-Zeiger (1 Wort)
7. (ILB) Identifikations-Länge (1 Byte)

Zugriff auf RFM

RFM.REL = an das Anwenderprogramm zu bindende RFM-Modul

CALL RFMIN (Initialisierung des RFM-Datenbereichs)

LXI B, parameterblockadr

CALL RFM

2. Operations-Codes

1. Lesen sequentiell (UBP)
2. Schreiben sequentiell (UBP)
3. Nicht belegt
4. Rückschreiben des letzten gelesenen Satzes (UBP)
5. Nicht belegt
6. Datei eröffnen (IP)
7. Datei(en) schließen
8. Nicht belegt
9. Lesen direkt (UBP, IP, ILB*)
10. Schreiben direkt (neu erzeugen) (UBP, IP)
11. Nicht belegt
12. Rückschreiben direkt (UBP, IP)
13. Löschen direkt (IP)
14. Wiederherstellen direkt (IP) *)

*) nur für indexsequentielle Dateien

() zusätzliche Parameter neben OCB, SCP,
ECP, FNP

3. Status-Codes

Hex Wert Bedeutung

- "01" Wiederholungswarnung
- "02" Überlaufwarnung (REORG nötig)
- "04" Bereits ausgeführt
- "08" Leer-Satz (nur bei Direkt-Dateien)

4. Fehlercodes

Hex. Wert	Bedeutung
"0A"	Datei nicht gefunden
"35"	Falscher Disketten-Typ (nicht INTEL)
"60"	RFM ist beschäftigt
"62"	Dateinummer nicht in Benutzung
"63"	Keine Direktzugriffs-Datei
"64"	Datei geöffnet gelassen
"65"	Ungültiger Dateiname
"66"	Überlauf-Fehler
"67"	Satz-Verdoppelung
"68"	Satz nicht gefunden
"69"	Satz gelöscht
"6A"	Ungültiger Identifikator
"6D"	Unzulässiger Operations-Code
"6E"	Dateiende
"6F"	Zu wenig Speicherplatz vorhanden, um Datei-Kontrollblock für die bezeichnete Datei anzulegen
99	Falsche Parameterangabe bei den RFM-Dienstprogrammen
??	Diskettenfehler

5. Aufbau der Dateien5.1 Indexsequentialle DateienDatei-Kopf-Sektor (128 Bytes)

Bytenr.	Inhalt
1	Dateityp 01 - Datei wurde angelegt 11 - Datei wurde eröffnet 21 - auf Datei wurde zugegriffen 31 - zugegriffene Datei blieb offen
2	Satzlänge
3	Erste Datenspur
4	Letzte Datenspur
5-6	Anzahl der geschriebenen Datensätze
7	Erste Indexspur
8	Letzte Überlaufspur
9	Schlüssellänge
10	Schlüsselposition innerhalb des Satzes

Indexspur-Blöcke (1 Block = 256 Bytes)

- . INDEX 0 (1. Block der Indexspur)

Bytenr.	Inhalt
1-2	Sektor/Spur des 1. freien Überlaufblockes
3-4	Sektor/Spur des höchsten INDEX 1-Blockes
5-6	Sektor/Spur des höchsten geschriebenen Datenblockes
7	Zeiger für Datei voll 00 = Datei nicht voll FF = Datei voll
8-...	Höchster Schlüssel der Datei (bzw. Dateiteil)
Schlüssellänge	

- . INDEX 1 (ab 2. Block der Indexspur)

Bytenr.	Inhalt
1-4	reserviert (4 Bytes)
5	aktuelle Schlüsselanzahl aus diesem INDEX 1-Block
6	max. Schlüsselanzahl aus diesem INDEX 1-Block
7	mindestens 10 Einträge mit: Sektor/Spur/höchster Schlüssel dieser Spur
.	
.	

Datenspur und Überlaufspurblöcke (1 Block = 256 Bytes)

Bytenr.	Inhalt
1-2	FF FF oder Verkettung (Sektor/Spur) mit dem Block, der in sequentieller Reihenfolge davor steht.
3-4	FF FF oder Verkettung (Sektor/Spur) mit dem Block, der in sequentieller Reihenfolge danach steht.
5	aktuelle Satzanzahl / Block
6	maximale Satzanzahl / Block
7-256	Datensatzbereich (250 Bytes)

5.2 Direkte DateienDatei-Kopf-Sektor (128 Bytes)

Bytenr.	Inhalt
1	Dateityp 00 - Datei wurde angelegt 10 - Datei wurde eröffnet 20 - auf Datei wurde zugegriffen 30 - zugegriffene Datei blieb offen
2	Satzlänge
3	Erste Datenspur
4	Letzte Datenspur
5-6	Höchste geschriebene Satznummer
7-8	Niedrigste Satznummer
9-10	Höchste Satznummer

Datenspur-Blöcke (1 Block = 256 Bytes)

Bytenr.	Inhalt
1-6	00 (6 Bytes)
7-256	Datensatzbereich (250 Bytes)

6. Dienstprogramme des RFM

Aufruf	Parameter
CREATE\$D, :lw:dat, :sl, as, esn Bsp: CREATE\$D, :F1:DIRECT,250,20,3	dat = Dateiname sl = Satzlänge (1-250) as = Anzahl der Datensätze esn = Erste Satznummer as = Anzahl der Datensätze als = Anzahl der Überlauftsätze sl = Schlüssellänge (1-24 Bytes) sp = Position des Schlüssels im Satz
CREATE\$I, :lw:dat, sl, as, als, sp Bsp: CREATE\$I, ISAM,50,10,5,15,0	qdat = Name der Quelldatei zdat = Name der Zieldatei (kann null je- doch nicht angelegt sein) as = (wahlweise) Anzahl der Daten- sätze der Zieldatei als = (wahlweise) Anzahl der Überlauf- sätze der Zieldatei
REORG\$C:lw:qdat,:lw:zdat[,as,aus] Bsp: REORG\$ISAM, :F1:ISAM1 REORG\$F1:dat,:F2:dat1,20,5	ein = Name der Eingabedatei aus = Name der Ausgabedatei (Bei in- dexsequentieller Ausgabedatei wird nach deren Schlüssel sortiert sl = Schlüssellänge (nur bei direkter Ausgabedatei) sp = Schlüsselposition (nur bei direkter Ausgabedatei)
SORT.RFM\$:lw:ein,:lw:aus[,sl,sp] Bsp: SORT.REFDIRECT,DIRECT.NEU,10,3 SORT.RFM\$:F1:DIRECT,:F2:ISAM	ein = Name der Eingabedatei
RCOVER\$[lw: Bsp: RCVER\$F1: RCOVER\$ISAM	RCOVER\$F1: RCOVER\$ISAM

7. RCOVER - Meldungen

- | | |
|---|---|
| RCOVER COMPLETED | - RCOVER BEENDET |
| RCOVER ABORTED | - RCOVER ABGE BROCHEN |
| CHECKING dat | - UBERPRUFUNG dat |
| FILE IS OK | - DATEI IST IN ORDNUNG |
| FILE IS CORRECTED | - DATEI WURDE BERICHTIGT |
| NO RECOVERY POSSIBLE | - KEINE WIEDERHERSTELLUNG MOGLICH |
| FILE WAS LEFT OPEN | - DATEI BLIEB OFFEN |
| OUTPUT FILE CREATED | - AUSGABE DATEI WURDE ANGELEGT |
| DATA TRACK(S) ADDED | - DATENSPUR(EN) WURDE(N) HINZUGEFUGT |
| NO RECORDS FOUND FOR RECOVERY | - KEINE SATZE WURDEN FÜR DIE WIEDERHERSTELLUNG GEFUNDEN |
| FILE RECOVERED | - DATEI WURDE WIEDERHERGESTELLT |
| FILE NOT RECOVERED | - DATEI WURDE NICHT WIEDERHERGESTELLT |
| VOLUME FULL INDICATOR SET IN NEW FILE | - IN DER NEUEN DATEI IST DER "DATEI VOLL"- ZEIGER GESETZT |
| PARAMETER ERROR | - FALSEHE PARAMETERANGABE |
| MULTIPART FILES NOT ALLOWED | - MEHRTEILIGE DATEIEN SIND UNZULÄSSIG |
| FILE REQUIRES RECOVERY | - WIEDERHERSTELLUNG DER DATEI IST ERFORDERLICH |
| OUTPUT FILE ACCESSED PREVIOUSLY | - AUF DIE AUSGABE DATEI WURDE BEREITS ZUGEGRIFFEN |
| OUTPUT FILE NOT COMPATIBLE WITH INPUT | - AUSGABE DATEI IST MIT DER EINGABE DATEI NICHT KOMPATIBEL |
| NOT ENOUGH INDEX-ENTRIES | - ZU WENIG INDEX-EINTRÄGE |
| TOO MANY INDEX-ENTRIES | - ZU VIELE INDEX-EINTRÄGE |
| INDEX ENTRY WRONG | - FALSCHER INDEX-EINTRAG |
| INVALID OVERFLOW LINKING | - UNZULÄSSIGE UBERLAUFVERKETTUNG |
| VOLUME FULL INDICATOR SET IN FILE WHICH IS NOT FULL | - "DATEI VOLL"-ZEIGER WURDE IN EINER NICHT VOLLEN DATEI GESETZT |
| RECORDS FOUND AFTER END OF DATA | - SATZE WURDEN NACH DATENENDE GEFUNDEN |
| TRACK = ??, SECTOR = ?? | - SPUR, SEKTOR IN WELCHER DER FEHLER AUFGETRETEN IST |

1. MEPLA - InbetriebnahmeMaster

- Laden von MEPLA
:Fn:MEPLA
- MEPLA - Initialisierung
 - Laufwerk-Angaben:
 - M - Laufwerk frei für MEPLA
 - L - lokales Laufwerk, dem Master vorbehalten
 - U - diese Laufwerk ist unbenutzt
 - A - dieses und alle weiteren Laufwerke frei für MEPLA
 - Kanalnummer: 1,2,...7
 - Aktivieren : + oder ohne Vorzeichen
 - Rücksetzen: - (minus)
 - Bsp: + 1,2,3,-4 (d.h. Kanal 1,2,3 aktiviert
 " 4 rückgesetzt)
 - Slave-Kanal-Status:
 - N = Kanal wurde nicht aktiviert
 - A = Kanal wurde aktiviert, Slave ist jedoch nicht eingeschaltet
 - V = Slave ist vorhanden und eingeschaltet
 - T = tätig (Slave greift während dieser Meldungsausgabe auf Master zu)
 - Automatische Initialisierung
 - Datei MINTDT mit EDIT zu erstellen
- Erneuter Anstoß der Konfigurierung
 - :Fn:MEPLA
 - :Fn:MEPKON

Slave

- Laden des BS1
- . Beim Einschalten automatisch
- . RS
- Anspringen des BS1
- . GFD00

Technische Daten

Betriebsweise: hx
Übertragungsprozedur: LSV1 (asynchron)
Übertragungsgeschwindigkeit: max. 19200 b/s

2. MEPLA - Programme

<u>Programme/Aufrufe</u>	<u>Bedeutung</u>
MEPLA MEPMM.xyz PINRT MEPLD RAMRST	<ul style="list-style-type: none"> - MEPLA Module xyz = BS1 - Versionsnummer (Bsp: BS1 Vers. 1.73 MMEPMM.<u>173</u>)
EPKON (oder MEPLA)	<ul style="list-style-type: none"> - Neukonfigurierung
EPEND	<ul style="list-style-type: none"> - Beenden des MEPLA-Betriebs
IBMRECS:Jn:name1,...	<ul style="list-style-type: none"> - Rücksetzung der "Protect"-Kennung bei IBM-Dateien
INTEL .FN:name	<ul style="list-style-type: none"> - Teilweises Retten von unter MEPLA abgestürzten sequentiellen INTEL-Ausgabedateien
SPOOL SI=:Fn:druckdat	<ul style="list-style-type: none"> - Ausgabe auf Drucker, zusätzlich zum MEPLA-Betrieb und lokalem Master-Prozeß (nur am Master)
ISPOOL	<ul style="list-style-type: none"> - Abbruch der SPOOL-Druckausgabe (nur am Master)
OCRFM	<ul style="list-style-type: none"> - Auf lokalen RFM (Sperre auf Dateiebene) schalten
ASRFM	<ul style="list-style-type: none"> - Rückschalten auf zentralen RFM (Sperre auf Satzebene)
MSUB.REL	<ul style="list-style-type: none"> - Assembler-Anschlußmodul an den MEPLA-RFM (anstelle RFM.REL)
ASM	<ul style="list-style-type: none"> - muß auf der Diskette vorliegen von der <u>am Master</u> RASM aufgerufen wird.
INK	<ul style="list-style-type: none"> - muß auf der Diskette vorliegen von der <u>am Master</u> LINK aufgerufen wird.
XREF	<ul style="list-style-type: none"> - muß auf der Diskette vorliegen von der <u>am Master</u> XREF aufgerufen wird.
DEN	<ul style="list-style-type: none"> - muß auf der Diskette vorliegen von der <u>am Master</u> DGEN aufgerufen wird.
RMA.T	<ul style="list-style-type: none"> - muß auf der Diskette vorliegen von der <u>am Master</u> FORMAT aufgerufen wird.
1	<ul style="list-style-type: none"> - muß auf der Diskette vorliegen von der <u>am Master</u> ASM aufgerufen wird.
IT	<ul style="list-style-type: none"> - muß auf der Diskette vorliegen von der <u>am Master</u> EDIT aufgerufen wird.
TY	<ul style="list-style-type: none"> - muß auf der Diskette vorliegen von der <u>am Slave</u> DELETE aufgerufen wird.
2	

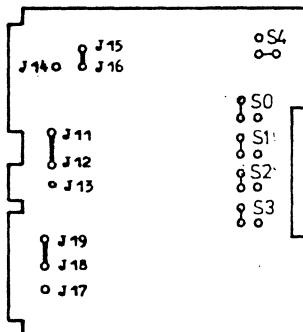
3. HardwarevoraussetzungenMaster

- Monitor: XMON/D Vers.6
- FBG-Cluster Interface
- Hauptspeicherkapazität > 48 kB

Einstellungen:

- FBG "Cluster Interface"
 - Brücken J11 - J12
J15 - J16
 - J18 - J19

S0 - S3

Falls vorhanden:

Einstellen wie abgebildet
(TTY Schnittstelle)
S4 Umschaltung
75 Band, ... Band
 $\textcircled{O} \rightarrow \Delta$ 75 Band

Einstellung der Ü-Geschwindigkeit
s.Kap. 16.1.8
Max. Ü-Geschw. ist abhängig von der
Länge des Kabels:
bis 100 m 19200 Bit/s
100-600 m 9600 Bit/s
Über 600 m muß eine Maxstrecke ver-
wendet werden.

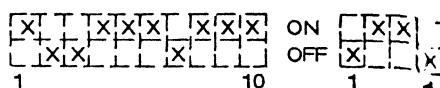
- Rest der FBG-Grundeinstellung

Slave

- Monitor : XMON/DS Vers.3
- Hauptspeicherkapazität: sinnvoll 48 kB

Einstellungen:

- FBG-Display Logic 1:

Rückwand:

NO PARITY	- NO	SPEED SELECT:
EVEN PARITY	- YES	je nach Geschw., die
TWO STOP BITS	- YES	auf der FBG "CLUSTER"
EXT ECHO	- YES	INTERFACE" eingestellt
AUTO RES	- ---*	wurde.
AUTO DSR	- ---*	

* Nur bei 6102

- Rest der FBG-s-Grundeinstellung

4. FehlermeldungenMaster - Fehlermeldungen

(Hex)

- 27 FILE ALREADY OPEN (Weiterer Druckversuch während der SPOOL-Funktion)
7F Fehlerhafter Monitor-PROM
81 Unzulässige Antwort vom Slave
82 Unstimmigkeit zwischen BS610 und MEPLA
84 Versuch eines unerlaubten Laufwerkzugriffs
A0 Fehler in der MEPLA-internen Prozedur
- A1 Gestörte Übertragung
Der Slave wird nur noch alle 2-3s gepollt, bis er wieder antwortet. Dann erfolgt das Polling wieder in den normalen Zeitabständen.
A2 Slave antwortet nicht
A4 Slave hat sich nach längerer Zeit gemeldet

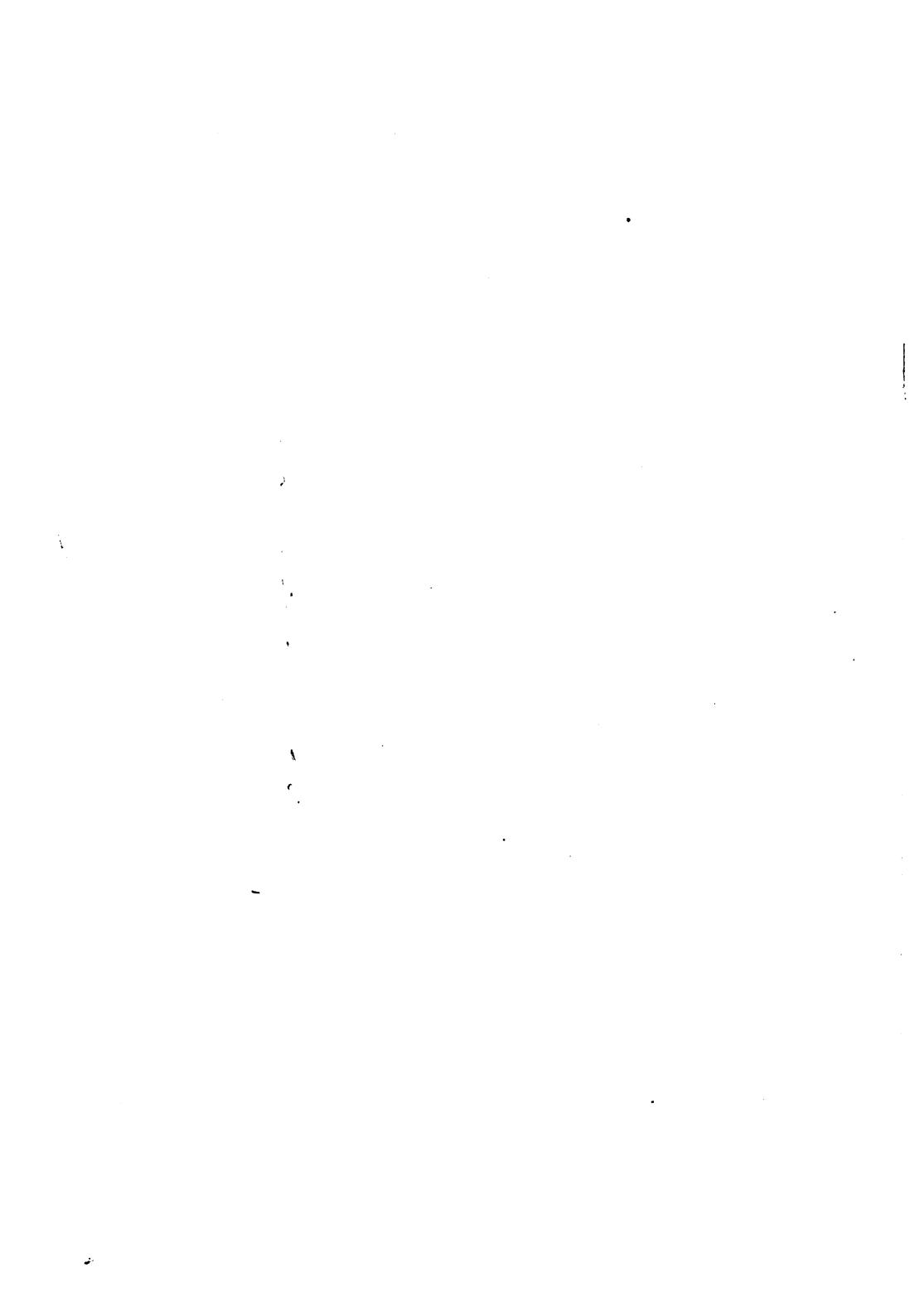
Außerdem sämtliche BS610-Fehlermeldungen

Slave-Fehlermeldungen

(Hex)

- 07 Kein Monitor XMON/DS (8 PROM's) vorhanden
7F Fehlerhafter Monitor-PROM
80 BS610 am Master nicht auffindbar
81 Unzulässige Antwort vom Master
82 Unstimmigkeit zwischen BS610 und MEPLA
84 Versuch eines unerlaubten Laufwerkzugriffs
A0 Fehler in der MEPLA-internen Prozedur
- A1 Übertragung nach dreimaliger Wiederholung erfolglos
A2 Keine Erwiderung vom Master
A3 Synchronisierung zwischen Master und Slave fehlerhaft
A5 Datenfehler im RAM bei der LSV-Prozedur

Außerdem sämtliche BS610-Fehlermeldungen mit Ausnahme der Nummern 01-08, die zunächst zwecks lokaler Behebung am Master erscheinen müssen.



1. LZG InbetriebnahmeZeichengeneriersystem (TCGS)

TCGS [SI=:lw:eingabe,] SO=:lw:ausgabe [,SL=:LP:]

Laden des Zeichengenerators (LCG)

LCG SI=:lw:ausgabe (=ausgabe vom TCGS)

Löschen des Zeichengeneratorspeichers (LCGCLR)

LCGCLR

Bsp: TCGS SI=GEN1,SO:F1:GEN2,SL=:LP:

LCG SI=:F1:GEN2

.

.

.

LCGCLR

LZG zusätzlich zum 54 KB-Arbeitsspeicher

- Assembler Anwenderprogramme die das TCGS bzw. den LCG aufrufen, dürfen erst ab Speicheradresse 2A00H gebunden werden.
- TCGS und LCG dürfen von BASIC-Programmen (mit BASIC.54) und COBOL-Programmen (mit RUN.54 über Assemblerschnittstelle) nicht gestartet werden. BASIC.48 bzw. RUN.48 sollte verwendet werden.

2. Beschreibung der Funktionstasten für das TCGS

Funktionstasten	Bedeutung
ER PAGE od REL +	Generierte Zeichen löschen
ER LINE od REL +	Löschen der 1. Zeilenhälfte und Positionieren auf Zeilenanfang
↖ (HOME)	Cursor auf Rasteranfang
LF	Sprung auf nächsten Zeilenanfang
←	Cursor 1 Position nach links
→	Cursor 1 Position nach rechts
↑	Cursor 1 Zeile höher
↓	Cursor 1 Zeile tiefer
Leerzeichen	Löschen des aktuellen Rasterpunktes
jedwelche Taste	Erzeugen eines Rasterpunktes
CR	Zeichen wird weggeschrieben
CTRL Pxy	Nächstes zu verarbeitendes Zeichen hat den Code xyH
CTRL Q	Laufendes Zeichen wird ausgedruckt
CTRL R	Abschluß

3. Hardware Voraussetzungen

- FBG LZG auf Einbauplatz A7
- FBG LZG muß mittels Anschlußkabel mit FBG DISPLAY LOGIC II verbunden sein. (Steckplatz für DFÜ Zeichen-PROM)
- Schalttereinstellung: FBG DISPLAY LOGIC II S9 auf "OFF"



1. FormularerstellungssystemAufruf

FORM CR

Modus (Meldung in der zweiten Status-Zeile)

C	Cursorposition	Zeile und Spalte
F	Formularebene	
I	Lade Formular	Name =
J	Setze/Bereinige Tab-stop	Eingabe von "+" oder "-"
K	Korrekturmodus	Funktion eingeben
O	Schreibe Formular	Name =
R	Formular einlesen	Eingabe Formularnummer
S	Formular abspeichern	Eingabe Formularnummer
T	Textfeld	Eingabe Darstellungsart
V	Variablenfeld	Eingabe Darstellungsart + Feldtyp
Z	Modusanwahl	Rückkehr zum Betriebs- system? Eingabe "J" oder "N"

Darstellungsart:

- 2 = halbhell
- 3 = blinkend
- 4 = invertiert
- 5 = unterstrichen
- 6 = unsichtbar
- 7 = normal

Feldtyp

Buch- stabe	Alpha- betisch	Alpha- numer.	Nume- risch	Links- bündig	Rechts- bündig	Muß- feld	Dup- feld	Vollst. Feld	Nullen- Auffüll zeichen	Vor- zeichen
A	x			x		x				
B	x				x					
C	x			x		x		x		
D	x					x				
E	x				x		x			
F	x								x	
H	x				x				x	
I	x				x	x		x	x	
P		x		x						
O		x		x		x				
R		x		x		x		x		
S		x			x				x	
T		x			x		x			
U		x			x			x	x	
V		x			x		x		x	
X	x				x		x		x	
a				x						
b				x						
c				x						
d				x				x		
e				x				x		
f				x				x		
h				x				x		
i				x				x		
j				x				x		
k				x				x		
l				x				x		
n				x				x		
p				x				x		
q				x				x		
r				x				x		
s				x				x		
t				x				x		
u				x				x		
v	x									
y	x									
z	x									

2. Datenerfassungssystem

Aufruf
FORM.D CR

Modusanwahl im DATENERFASSUNGSMODUS

MODE +	Bedeutung
A	Verzweigung in den Änderungsmodus
C	Dateiwahl schließen
E	Verzweigung in die Erfassung
F	Neu Formularsatz einlesen
H	Anzeige der freien Datensätze in der Datendatei
O	Öffnen einer Datendatei
P	Prüfmodus
R	Formularwechsel
S	Verzweigung in den Suchmodus
W	Wegschreiben der neu erfaßten Daten
Z	Rückkehr zum Betriebssystem

Modusanwahl im ÄNDERUNGSMODUS

MODE +	Bedeutung
E	Verzweigung in die Datenerfassung
G	Sequentielles Lesen
M	Lesen eines zur aktuellen Maske zugehörigen Datensatzes
P	Prüfmodus
R	Formularwechsel
S	Verzweigung in den Suchmodus
W	Rewrite Datensatz
DEL-Taste	Datensatz löschen

Modusanwahl im SUCHMODUS

MODE +	Bedeutung
A	Verzweigung in den Änderungsmodus
B	Positionierung des Lesezeigers
E	Verzweigung in den Datenerfassungsmodus
G	Sequentielles Lesen
M	Lesen einer zur aktuellen Maske zugehörigen Datensatzes
P	Prüfmodus
R	Formularwechsel
DEL-Taste	Datensatz löschen

Modusanwahl im PRÜFMODUS

MODE +	Bedeutung
E	Verzweigung in die Datenerfassung
G	Sequentielles Lesen
M	Lesen eines zur aktuellen Maske zugehörigen Datensatzes
R	Formularwechsel
S	Verzweigung in den Suchmodus
W	Zurückschreiben Datensatz
DEL-Taste	Datensatz löschen

3. Beschreibung der Funktionstasten

BI-Tastatur 2 Deutsche	BI-Tastatur 2 international	Bedeutung
SEND	SEND	Ausgabe auf Drucker eines Formulars
DEL 	DEL 	Aufheben der getätigten Eingaben Vorwärtssuchen nach einem Suchbegriff (+ ← Taste)
ANF	REQ	Rückwärtssuchen nach einem Suchbegriff (+ ← Taste)
CHAR →	CHAR	Bewegung ab Cursorposition den Rest des Feldes um eine Position nach rechts
CHAR ←	CHAR	Bewegung ab Cursorposition den Rest des Feldes um eine Position nach links
WORT →	WORD	Bewegung des nachfolgenden geschützten/ ungeschützten Feldes um eine Position nach rechts
WORT ←	WORD	Bewegung des nachfolgenden geschützten/ ungeschützten Feldes um eine Position nach links
F6	F6	Anzeigen der verbrauchten Datenbytes
AR 	CF	Setzen eines Fehlendesteuerzeichens
TAB →	TAB	Positionieren des Lesezeigers auf den ersten Datensatz Setzen den Cursor auf den nächsten Ta- bulatorstop in der aktuellen Zeile
TAB ←	TAB	Setzen den Cursor auf den vorhergehen- den Tabulatorstop in der aktuellen Zeile
↓↓ ↑↑	↓↓	Positionieren des Lesezeigers auf den letzten Datensatz
STX	STX	Duplizieren des Feldes ab Cursorposition
ETX	ETX	Duplizieren des Feldes in Abhängigkeit mit der "← TASTE" Ausgeben eines Minus-Zeichens auf der letzten Feldposition
NORMAL	NORMAL	Ausrichtung nach rechts und links mit Nullen auffüllen
F2	F2	Nur für Normalfelder im Prüfmodus Löschen eines Zeichens
F1	F1	Löschen eines Zeichens im Prüfmodus
F3	F3	Korrektur feldweise + CR-Taste im Prüfmodus



Aufbau der Dateien

Dateien müssen mit dem Dienstprogramm CREATE (siehe RFM) erstellt werden.

Länge: 250 Byte
Satznummer: 1

ersten beiden Sätze werden für die interne Verwaltung bestimmt.

100 angefangene Datensätze wird ein weiterer Verwaltungssatz reserviert.

100 Datensätze	=3 Verwaltungssätze
101-200 Datensätze	=4 Verwaltungssätze
201-300 Datensätze	=5 Verwaltungssätze
usw.	

jedem Datensatz (250 Byte) sind die ersten 10 Bytes für Verwaltungszwecke reserviert.

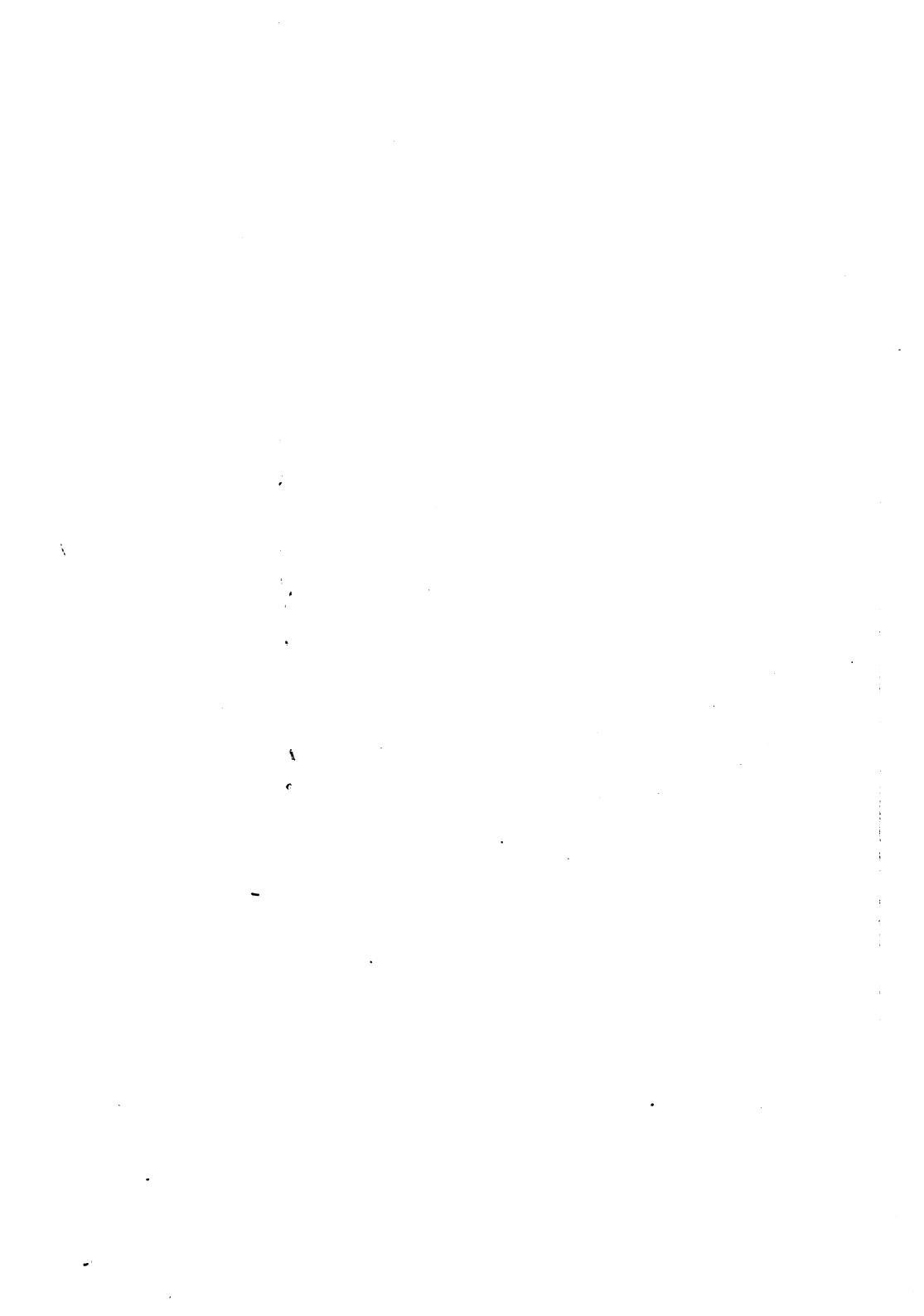
Aufbau der Verwaltungsbytes:

te 0	laufende Satznummer (hexadezimal)
te 1	(byte vertauscht)
te 3	linkes Halbbyte: 0 = Satz gültig 8 = Satz gelöscht rechtes Halbbyte: Maskennummer
te 4	frei (00H)
te 9	

5. Fehlermeldungen

Fehlercode (Hex-Word)	Bedeutung
08	Das angewählte Laufwerk ist nicht angeschlossen
0A	Die Datei wurde nicht gefunden (open)
3B	Die Datei ist zerstört
61	Fehler im DM Kontrollsatz
62	Datei-Nr. nicht im Gebrauch (ist nur möglich, wenn durch äußere Einflüsse das Programm manipuliert wurde)
63	Keine Direktzugriffsdatei (open)
64	Datei blieb offen; kein Zugriff möglich (RCOVER)
65	Unzulässiger Dateiname (open)
66	Datei voll (schreiben)
69	Gelöschte Datei (open)
6A	Ungültiger Identifikator
6D	Unzulässiger Operationscode
6E	Ein Zugriff auf einen Datensatz nach dem letzten Datensatz wurde versucht
6F	Dem Dateiverwaltungssystem steht nicht genügend Speicherplatz zur Verfügung
70	Ein Zugriff auf einen Datensatz vor dem ersten Datensatz der Datei wurde versucht
71	Datei leer (read)

Weiterhin können auch Fehlercodes des Betriebssystems für Diskettenoperationen ausgegeben werden.



KAPITEL 1	PROZEDUREN	Seite
1. MSV1 - Prozedur Unterstation		5-1-1
1.1 Allgemeines		.
1.2 Hardwarevoraussetzungen		5-1-2
1.3 Benutzerschnittstelle		5-1-3
1.4 Fehlermeldungen		5-1-9
1.5 Testhilfen		5-1-10
2. MSV2/ BSC1/2-Prozedur		5-1-11
2.1 Allgemeines MSV2		.
2.2 Allgemeines BSC1/2-Prozedur		5-1-12
2.3 Hardwarevoraussetzungen		5-1-13
2.4 Benutzerschnittstelle		5-1-14
2.5 Fehlermeldungen		5-1-22
2.6 Testhilfen MSV2		5-1-23
2.7 Testhilfen BSC1/2		5-1-24
KAPITEL 2	EMULATION	
1. Emulation 8150 (Dialog)		5-2-1
1.1 Inbetriebnahme		.
1.2 Beschreibung der Funktionstasten		5-2-2
1.3 Hardwarevoraussetzungen		5-2-4
2. Emulation 8151		5-2-5
2.1 Inbetriebnahme		.
2.2 Beschreibung der Funktionstasten		5-2-6
2.3 Hardwarevoraussetzungen		5-2-8
3. Emulation 8161		5-2-9
3.1 Inbetriebnahme		.
3.2 Beschreibung der Funktionstasten		5-2-10
3.3 Hardwarevoraussetzungen		5-2-13
4. Emulation 3270		5-2-14
4.1 Inbetriebnahme		.
4.2 Beschreibung der Funktionstasten		5-2-15
4.3 Hardwarevoraussetzungen		5-2-16
4.4 Fehlermeldungen		5-2-18

KAPITEL 3	DATEIÜBERTRAGUNGEN (FILETRANSFER)	Seite
1.	FT 6.610-6.610	5-3-1
1.1	Bedienung	.
1.2	Hardwarevoraussetzungen	5-3-5
1.3	Softwarevoraussetzungen	5-3-6
1.4	Fehlermeldungen	5-3-7
1.5	Testhilfen	5-3-8
2.	Filetransfer 6.610 - BS2000	5-3-9
2.1	Bedienung	.
2.2	Hardwarevoraussetzungen	5-3-11
2.3	Softwarevoraussetzungen	5-3-13
2.4	Fehlermeldungen	5-3-14
2.5	Testhilfen	5-3-16
3.	EMX 10	5-3-17
3.1	Bedienung	.
3.2	Hardwarevoraussetzungen	5-3-22
3.3	Softwarevoraussetzungen	5-3-24
3.4	Fehlermeldungen	.



1. MSV1-Prozedur-Unterstation

1.1 Allgemeines

- FTZ Zulassung

- Fernsprechwählnetz, Direktrufnetz

FTZ A 52 - 10 DEE 983

- DATEX-L-Netz

FTZ T 35 - 3 1/186/010/80

- Übertragungsverfahren

- Betriebsart: Abrufbetrieb, synchron, hx oder dx, Takt von der Datenübertragungseinrichtung
- Übertragungsgeschwindigkeit: max. 4800 Bit/s bei Fernsprech-Wählnetzen, Standleitungen und am DATEX-L-Netz.
- Übertragungscode: ASCII, normierter Modus, 7 Bit mit Paritätsbit ungerade.
- Datensicherung: Zeichenparität ungerade, Blockparität gerade, BCC mit Zeichenparität gesichert.

1.2 Hardwarevoraussetzungen

Manueller Betrieb wenn Sprechverkehr erwünscht ist, sonst Automatikbetrieb (gilt nicht für DATEX-L).

- FBG'en

FBG-Synchroninterface: Einbauplatz A4
Brücke I2-I3
Keinen internen Takt

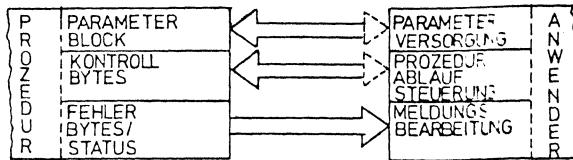
FBG-Display-Logik I:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>X</td><td></td><td>X</td><td>X</td><td></td><td>X</td><td>X</td> </tr> <tr> <td>X</td><td>X</td><td></td><td>X</td><td>X</td><td>X</td><td></td> </tr> <tr> <td style="text-align: center;">1</td><td></td><td></td><td></td><td></td><td></td><td style="text-align: center;">10</td> </tr> </table>	X		X	X		X	X	X	X		X	X	X		1						10	ON OFF
X		X	X		X	X																	
X	X		X	X	X																		
1						10																	
U39																							

FBG-Display-Logik II:	U61	<table border="1"> <tr> <td>X</td><td>X</td><td>X</td><td>X</td><td></td><td>X</td> </tr> <tr> <td>X</td><td>X</td><td></td><td>X</td><td>X</td><td>X</td> </tr> <tr> <td>1</td><td></td><td></td><td></td><td></td><td>10</td> </tr> </table>	X	X	X	X		X	X	X		X	X	X	1					10	ON OFF
X	X	X	X		X																
X	X		X	X	X																
1					10																

Auf allen anderen FBG'en sind die Standardschalter-einstellungen durchzuführen.

1.3 Benutzerschnittstelle

- Aufbau



- PARAMETER

Alle Felder sollten nur mit ihren symbolischen Namen angesprochen werden, um inkompatible MSV1 Änderungen zu vermeiden.

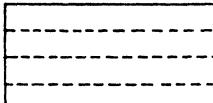
- Pufferstatusblock

Symb.	Adr.	Bit 7 6 5 4 3 2 1 0	Bedeutung
PSØID *			ID-Zeicher. 1
PSØKE			ID-Zeicher. 2
PSØKS			KONTROLL BYTE EMPF. PUFFER
PSØZR			KONTROLL BYTE SENDE PUFFER
PSØZE			ZEIGER EMPFANGSPUFFER
PSØAE *			ANFANGSADRESSE EMPF. PUFFER
PSØzs			ZEIGER SENDEPUFFER
PSØas *			ANFANGSADRESSE SENDEPUFFER
PSØle *			LÄNGE EMPFANGSPUFFER
PSØls *			LÄNGE SENDEPUFFER
PSØfw			FEHLERBYTE
PSØfz *			FUNKTIONSZEICHEN
PSØsz			EMPFANGENE STEUERZEICHEN (Folgende Steuerzeichen werden hier hinterlegt: ENQ, EOT, NAK, 10 für ACKØ/1, 3F für WABT, 3C für RVI)

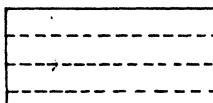
Die Zeiger dürfen vom Anwender nicht verändert werden.

* Vom Anwender ist vor dem Prozedurstart (CALL ANLØ) zu versorgen.

- Adressblock

Symb. Adr. Bit 7 6 5 4 3 2 1 0
 ADE * 

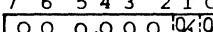
Bedeutung
 POLL-ADRESSE; SENDEAUFFORDERUNG
 (MAX. 4 ZEICHEN. NICHT BENÖTIGTE BYTES SIND MIT "FF" ZU BELEGEN)

ADAØ * 

SELECT ADRESSE; EMPF.
 AUFFORDERUNG
 (SONST WIE OBEN)

* VOM ANWENDER VOR DEM PROZEDURSTART ZU VERSORGEN

- Weitere Schnittstellenbytes

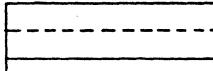
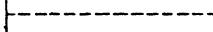
Symb. Adr Bit 7 6 5 4 3 2 1 0
 PSØEQ 

Bedeutung

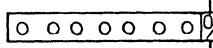
POLLING EMPFANGEN
 SELECTING EMPFANGEN

} 1)

- 1) BITS MÜSSEN VOM ANWENDER PROGRAMM RÜCKGESETZT WERDEN

NAKIN 
 NAKOU 

ZÄHLER EMPF. NAK's
 (ZÄHLT AB PROZEDUR START)
 ZÄHLER GESEND. NAK's
 (ZÄHLT AB PROZEDUR START)

TAKT 

TAKT = Ø = 500 MSEC
 ZEITTAKT 1 SEC
 TAKT = 1 = 500 MSEC

- Weitere Parameter

Symb.	Adr.	Bit 7	6	5	4	3	2	1	0	Bedeutung
SYNW									(Ø3)	Anzahl der SYN-ZEICHEN
SYWT2									(Ø2)	Anzahl SYN nach Ablauf von T2
EPADW									(Ø1)	Anzahl der PAD's nach Text oder Steuerzeichen
BETRA									(ØØ)	BEIRUJTISART
RESA									(1B)	ZEICHEN ESC UM ABBRUCH VON TASTATUR WÄHREND DES VERBINDUNGSAUFBAU ZU ERZWINGEN

KE STW	^{Δ)}	(Ø7)	Kopfeintrag Steuerwart
T1W		(1E)	T1 = 3 sec
		(ØØ)	= 30 x 100 m sec
T2W		(ØA)	T2 = 1 sec
		(ØØ)	
T3W		(14)	T3 = 2 sec
		(ØØ)	
T4W		(C8)	T4 = 20 sec
		(ØØ)	
		nicht benutzt.	
T6W		(Ø2)	Keine DFU-Zeit
		(ØØ)	Quittungsverzögerungszeit um WAIT zu vermeiden.
WAITW		(ØA)	Die effektive Verzögerungszeit errechnet sich aus:
		(ØØ)	T6W x WAIT und darf max. 25 sec. betragen.

Δ) Ab MSV1-Prozedur REV 3.00 R

() = Voreinstellung

KONTROLLBYTE KS ≡ ADR PSØKS

BIT 7654 3210
Puffer nicht lesbar
| Puffer leer
| | EOT nach ETX und positiver Quittung
0000 0000
11-1 -1--
| | |
| | | TTD statt Pufferinhalt senden
| | | Kein EOT nach positiver Quittung
| Puffer voll
Puffer lesbar

KONTROLLBYTE KE ≡ ADR PSØKE

BIT 7654 3210
Puffer nicht lesbar
| Puffer leer
| | Normale Quittierung
| | | Normale Quittierung
0000 0000
111- 1---
| | |
| | | Nächsten Satz mit RVI quittieren (Break)
| | | EOT statt Quitt. senden (Block-Interrupt)
| Puffer voll
Puffer lesbar

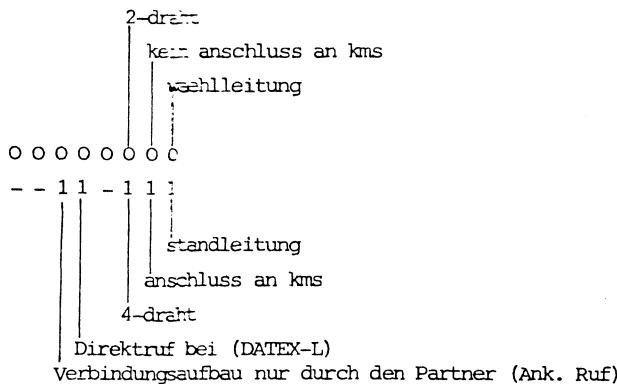
KOPFEINTRAG STEUERWORT ≡ KESTW (Ab MSV1-Prozedur REV 3.00R)

BIT 7654 3210
| | | Kein ID1 senden
| | | !Kein ID1 + ID2 senden
| | | | Kein FZ senden
0000 0000
---- -111
| | FZ senden
| | ID1 + ID2 senden
ID1 senden



- BETRIEBSART $\hat{=}$ ADR ~~0000~~ RA

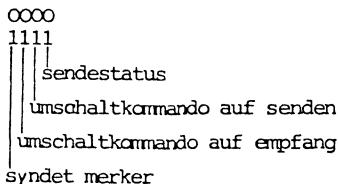
bit 7 6 5 4 3 2 1 0



- PROZEDURSTATUS $\hat{=}$ ADR PZW

bit 7654

empfangsstatus



bit 3210

hexa

- 0 = grundstellung, eot gesendet
- 1 = eot empfangen
- 2 = ade eng empfangen * polling **
- 3 = senden satz (block)
- 4 = bcc gesendet
- 5 = warten auf dle-folgezeichen
- 6 = folgeblock senden
- 7 = ada eng empfangen * selecting **
- 8 = stx empfangen
- a = etb empfangen
- b = etx empfangen
- c = richtiges bcc empfangen, quittung ack
- d = fehlerh. bcc empfangen, wiederholung, ttd empfangen
- e = richtiges bcc empfangen, quittung wabt
- f = e-puffer ueberlauf, timeout t3, wiederh.

• PROZEDURSTATUS ≡ ADR PZWH

bit 7654 3210

- kein wabt empfangen
- kein eot von disc gesendet
- kein etb empfangen
- kein fehler nach itb
- kein itb empfangen
- bcc-kontrolle nicht okay !!!
- kein timeout t1
- 7fh (pad)-zeichen nach text gesendet

0000 0000

1111 1111

- 7fh (pad)-zeichen nach text senden
- timeout t1
- bcc-kontrolle okay
- itb empfangen
- bcc-fehler nach itb
- etb empfangen
- eot von disc gesendet
- wabt empfangen

• PROZEDURSTATUS ≡ ADR PZWH1

bit 7654 3210 p z w,h 1

- aufford. phase
- eot nach etx gefordert
- keine blockwh n.f.lfnr möglich
- normale quittierung von pol., sel.
- keine rvi phase
- kein disc empfangen
- kein ttd gesendet

0000 0000

111- 1111

- ttd (stx, eng) gesendet
- disc (dle, eot) empfangen
- rvi (dle, <) empfangen
- pol., sel. mit disc quittieren
- blockwh n.f.lfnr. möglich
- kein eot nach etx gefordert
- textphase



1.4 Fehlermeldungen

Außer den BS610 Fehlermeldungen können noch Fehlermeldungen in der Prozedur vorkommen.

Diese Fehlermeldungen werden in der Zelle,

"PSOFW"

hinterlegt und können vom Anwender ausgewertet werden.

- PSOFW

FEHLERQUELLEN

error 01 = abbruch !!! durch sendepuffer überlauf,
keine ff-endm. eingetragen !
02 = empf-puffer, e-pool überlauf
03 = abbruch !!! durch zeitüberschreitung
t4=24sec bei warten auf blxx, eot, emq!
04 = eot nach falscher quittung trotz
n-facher wiederholung !
05 = nicht in msy1
06 = rvi empfangen !
07 = eot nach etb empfangen, datenmenge
unvollständig !
08 = nicht in msy1
09 = abbruch !!! durch empfang n. senden von
DISC (DLE-EOT)
0a = abbruch !!! keine Verbindung
0b = timeout t1 durch fehl. o. falsche quitt.
0c = nicht in msy1
0d = abbruch !!! durch eingabe von ESC oder
unter RESA def.zch in anlaufphase

1.5 Testhilfen

- Tracen

Der Verkehr auf der Leitung lässt sich durch Ändern der Adresse "BIZEI" (*) von "C9" auf "00" auf dem Bildschirm anzeigen.

" < " bedeutet Empfangsdaten

Das Zeichen " > " bedeutet Sendedaten

Das Tracen ist wie folgt möglich:

Laden des Programmes mit DEBUG.

Aufruf:

:fn:DEBUG3:fn:Anwenderprogramm

Anschließend kann mit dem Monitor die entsprechende Adresse geändert und das Programm gestartet werden

- Demo - Programm

Für den ersten Test der Prozedur steht ein Demonstrationsprogramm bereit. Mit diesem Demoprogramm können Daten von der Tastatur gesendet und empfangene Daten auf Bildschirm ausgegeben werden.

Aufruf:

:fn:DEBUG3:fn:demo.u32

:fn:DEBUG3:fn:MSV1D (ab REV 3.00R)

Anschließend müssen die Zellen:

BETR ≡ 700D mit der Betriebsart(siehe 1.3,Seite.....)

SAUF ≡ Y002 mit der Sendaufforderungsadresse

EAUF ≡ 7007 mit der Empfangsaufforderungsadresse

IDENT ≡ 7000 mit der Identifikation

ANZSY ≡ 700C mit der Anzahl der Syn-Zeichen

geladen werden.

Soll getract werden, so ist die Adresse BIZEI ≡ 5CC5 * von C9 auf 00 zu ändern. Anschließend ist das Programm mit dem Monitor zu starten.

* dresse abhängig von den LINK-Angaben.

2. MSV2/BSC1/2-Prozedur2.1 Allgemeines MSV2

- FTZ Zulassung
 - . Fernsprechwählnetz, Direktrufnetz
FTZ A 52 - 10 DEE 983
 - . DATEX-L
FTZ T 35 - 3 1/186/010/80
- Übertragungsverfahren
 - . Betriebsart: Stapelbetrieb, synchron hx oder dx, Takt von der Datenübertragungseinrichtung
 - . Übertragungsgeschwindigkeit: max. 4800 Bit/s bei Fernsprechwahlnetzen, Standleitungen und am DATEX-L-Netz
 - . Übertragungscode: ASCII, normierter Modus, 7 Bit mit Paritätsbit ungerade
 - . Datensicherung: Zeichenparität ungerade, Blockparität gerade, BCC mit Zeichenparität gesichert.

2.2 Allgemeines BSC1/2 - Prozedur

- FTZ Zulassung
 - . Fernsprechwählnetz, Direktrufnetz
FTZ A 52 - 10 DEE 983
 - . DATEX-L-Netz
FTZ T 35 - 3 1/186/010/80
- Übertragungsverfahren
 - . Betriebsart: Stapelbetrieb, synchron hx oder dx, Takt von der Datenübertragungseinrichtung
 - . Übertragungsgeschwindigkeit: max. 4800 Bit/s bei Fernsprech-Wählnetzen, Standleitungen und am DATEX-L-Netz
 - . Übertragungscode: EBCDI, normierter Modus, 8 Bit ohne Paritätsbit
 - . Datensicherung: Zyklische Blocksicherung entsprechend CRC 16.



2.3 Hardwarevoraussetzungen

- FBG'en

FBG-Synchroninterface: Einbauplatz A4
Brücke I2-I3
Keinen internen Takt

FBG-Display-Logik I:

U79

	X	X	
X			X
1			4

ON
OFF

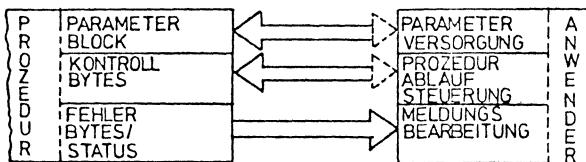
FBG-Display-Logik II:

U61	X	X	X	X	X	X	X	ON
	X	X		X	X	X		OFF
1							10	

Auf allen anderen FBG'en sind die Standardschaltereinstellungen durchzuführen.

2.4 Benutzerschnittstelle

- Aufbau



- PARAMETER

Alle Felder sollten nur mit ihren symbolischen Namen angesprochen werden, um inkompatible MSV1 Änderungen zu vermeiden.

- Pufferstatusblock

Symb.	Adr.	Bit 7 6 5 4 3 2 1 0	Bedeutung
PSØID *			ID-ZEICHEN 1
PSØKE			ID-ZEICHEN 2
PSØKS			KONTROLL BYTE EMPF. PUFFER
PSØZR			KONTROLL BYTE SENDE PUFFER
PSØZE			ZEIGER EMPFANGSPUFFER
PSØAE *			ANFANGSADRESSE EMPF. PUFFER
PSØZS			ZEIGER SENDEPUFFER
PSØAS *			ANFANGSADRESSE SENDEPUFFER
PSØLE *			LÄNGE EMPFANGSPUFFER
PSØLS *			LÄNGE SENDEPUFFER
PSØFW			FEHLERBYTE
PSØSZ			EMPFANGENE STEUERZEICHEN (Folgende Steuerzeichen werden hier hinterlegt: ENQ, EOT, NAK, 10 für ACKØ/1, 3F für WABT, 3C für RVI, 07 für BELL)

Die Zeiger dürfen vom Anwender nicht verändert werden.

* Vom Anwender ist vor dem Prozedurstart (CALL ANLØ) zu versorgen.

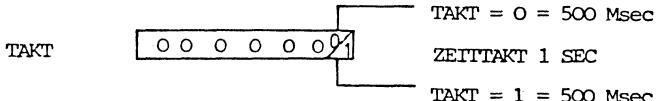
- Pufferstatusblock 1:

Symb. Adr.	Bit7	6	5	4	3	2	1	0	Bedeutung
PS1KE									Kontrollbyte Empf. Puffer
PSIKS									Kontrollbyte Sende Puffer
PS1ZR									
PS1ZE									Zeiger Empfangspuffer
PS1AE	*								Anfangsadr. Empfangspuffer
PS1ZS									Zeiger Sendepuffer
PS1AS	*								Anfangsadresse Sendepuffer
PS1LE	*								Länge Empfangspuffer
PS1LS	*								Länge Sendepuffer

* Vom Anwender vor dem Prozedurstart zu versorgen

- Weitere Schnittstellenbytes

Symb. Adr.	Bit7	6	5	4	3	2	1	0	Bedeutung
NAKIN									Zähler empfangene NAK's (zählt ab Prozedurstart)
NAKOU									Zähler gesendete NAK's (zählt ab Prozedurstart)



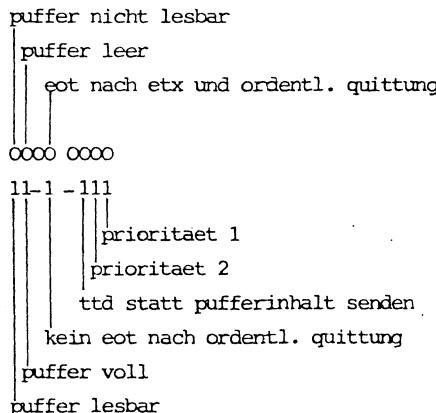
Weitere Parameter

Symb. Adr.	Bit 7 6 5 4 3 2 1 0	Bedeutung
WHW	(Ø3)	Anzahl Wiederholungen
WBTW	(28)	Anzahl der zulässigen WABT
WAITW	(ØA)	Anzahl T6-Zyklen als WABT Verzögerung
SYNW	(Ø7)	Anzahl der SYN-Zeichen
SYWTZ	(Ø2)	Anzahl SYN's nach Ablauf T2
EPADW	(Ø1)	Anzahl PAD's nach Text oder Steuerzeichen
BETRA	(ØØ)	Betriebsart
	(1E)	T1 = 3 sec =
T1W	(ØØ)	30 x 100 Msec
T2W	(ØA)	T2 = 1 sec
	(ØØ)	
T3W	(14)	T3 = 2 sec
	(ØØ)	
T4W	(C8)	T4 = 20 sec
	(ØØ)	
T1WA	(ØA)	T1A = 1 sec in Auford.
	(ØØ)	Phase bei Status Primary

() = Voreingestellt

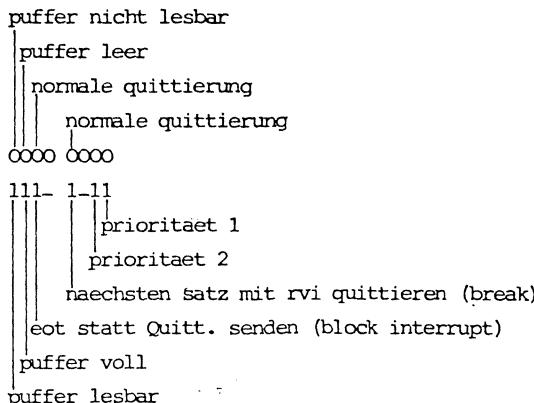
- KONTROLLBYTES KS $\hat{=}$ ADR. PS0KS, PS1KS
MSV2 und BSC1/2

Bit 7654 3210



- KONTROLLBYTES KE $\hat{=}$ ADR. PS0KE, PS1KE
MSV2

Bit 7654 3210



BSC1/2

bit 7654 3210

```

puffer nicht lesbar
  | text mit SOH beginnend
  | normale quittierung
  |   | normale quittierung
  |   | 0000 0000
 111- 1-11
  |   |   | prioritaet 1
  |   |   | prioritaet 2
  |   |   naechsten satz mit rvi quittieren (break)
  |   eot statt Quitt. senden (block interrupt)
  |   text mit stx beginnend
puffer voll, lesbar

```

. BETRIEBSART = ADR. BETRA

bit 7654 3210

```

nicht auf ankommenden ruf warten
  | kein direktruf bei DATEX-L
  | secondary (warten auf ruf)
  |   | 2-draht
  |   | waehlleitung
  |   | 0000 0000
  --11 11-1
  |   |   | standleitung
  |   |   | 4-draht
  |   |   primary (abgehender ruf)
  |   | Direktruf bei (DATEX-L)
  |   Verbindungsauftbau nur durch den Partner

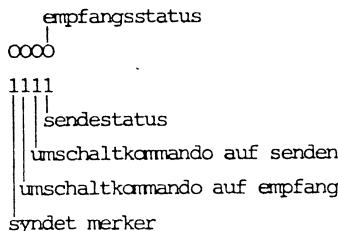
```



• PROZEDURSTATUS ≡ ADR. PZW

MSV2, BSC1/2

bit 7654



MSV2

- 0 = grundstellung
- 1 = id/eng gesendet
- 2 = warten auf 0/1 nach ack in auff.-phase
- 3 = senden block (satz)
- 4 = bcc gesendet
- 5 = warten auf 0/1 nach ack in text-phase
- 6 = folgeblock senden
- 7 = id/eng empfangen und empfangsbereit
- 8 = stx empfangen
- 9 = id empfangen, stx des 2.-n.blockes empf
- a = etb empfangen
- b = etx empfangen
- c = richtiges bcc empfangen
- d = fehlerh. bcc empf., parityfehler, wiederholung
- e = voruebergehend nicht empfangsbereit (WABT senden)
- f = e-puffer ueberlauf, timeout t3, wiederholung

Bits 2-0
in Hexa.

BSC1/2

- 0 = grundstellung
- 1 = id/eng gesendet
- 2 = warten auf 0/1 nach ack in auff.-phase
- 3 = senden block (satz)
- 4 = bcc gesendet
- 5 = warten auf 0/1 nach ack in text-phase
- 6 = folgeblock senden
- 7 = id/eng empfangen und empfangsbereit
- 8 = nicht benutzt
- 9 = stx empfangen, SOH empfangen
- a = etb empfangen
- b = etx empfangen
- c = richtiges bcc empfangen
- d = fehlerh. bcc empf., parityfehler, wiederholung
- e = voruebergehend nicht empfangsbereit (WACK senden)
- f = e-puffer ueberlauf, timeout t3, wiederholung

Bits 2-0
in Hexa.

. PROZEDURSTATUS ≡ ADR. PZWH

MSV2/BSC1/2

bit 7654 3210

slave status (textempf. -station)

aufford. phase

kein etb empfangen

kein fehler nach itb

kein itb empfangen

bcc-kontrolle nicht okay !!!

kein eot nach etx senden

7fh(pad)-zeichen nach text gesendet

0000 0000

1111 1111

7fh(pad)-zeichen nach text senden

eot nach etx senden

bcc-kontrolle okay

itb empfangen

bcc-fehler nach itb

etb empfangen

textphase

master status (textsendedstation)



- PROZEDURSTATUS ≡ ADR. PZWHL

MSV2

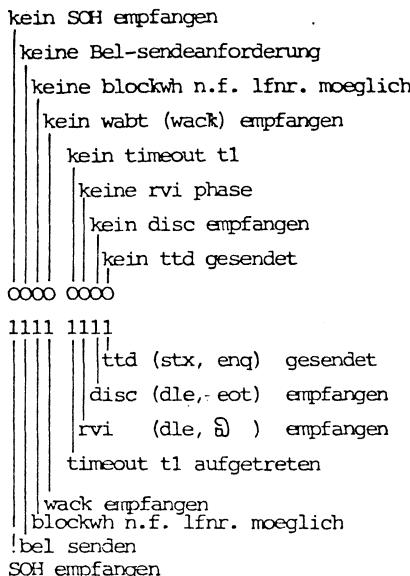
bit 7654 3210



- PROZEDURSTATUS ≡ ADR. PZWHL

BSC1/2

bit 7654 3210



2.5 Fehlermeldungen

Außer den BS 610 Fehlermeldungen können noch Fehlermeldungen in der Prozedur vorkommen.

Diese Fehlermeldungen werden in der Zelle

"PSOFW"

hinterlegt und können vom Anwender ausgewertet werden.

- PSOFW

FEHLERQUELLEN

- error 01 = abbruch !!! durch sendepuffer zu klein, fehlerhafter id-block, keine endmarke gefunden !
- 02 = empf-puffer zu klein !
- 03 = abbruch !!! durch zeitüberschreitung T4=24sec bei warten auf block, eot, enq !
- 04 = eot nach falscher quittung !
trotz n-facher wiederholung !
oder zulässige anzahl Nak's in aufford. phase überschritten
- 05 = nicht benutzt
- 06 = rvi empfangen !
- 07 = eot nach etb empfangen,
datenmenge unvollständig !
- 08 = warnung vor abbruch durch zeitüberschreitung
t4 = nach eot
(aufforderung zum senden !!!)
oder bei positiver quittung nach etb
(aufforderung zum senden nächster satz).
error 08 wird von prozedur rückgesetzt
- 09 = abbruch !!! durch empfang o. senden von DISC (DLE-EOT)
- 0a = abbruch !!! keine Verbindung
- 0b = timeout T1 durch fehl.o.falsche quitt.
- 0c = eot statt quittung empfangen .
(block interrupt)
- 0d = abbruch !!! durch eingabe von ESC in anlaufphase



2.6 Testhilfen MSV2

- Tracen

Der Verkehr auf der Leitung lässt sich durch Ändern der Adresse "BIZEI" = (*) von "C9" auf "00" auf dem Bildschirm anzeigen.

" < " bedeutet Empfangsdaten

Das Zeichen

" > " bedeutet Sendedaten

Das Tracen ist wie folgt möglich:
Laden des Programmes mit DEBUG.

Aufruf:

:fn:DEBUGZ:fn:Anwenderprogramm

Anschließend kann mit dem Monitor die entsprechende Adresse geändert werden

- Demo-Programm

Für den ersten Test der Prozedur steht ein Demonstrationsprogramm bereit. Mit diesem Demoprogramm können Daten von der Tastatur gesendet und empfangene Daten auf Bildschirm ausgegeben werden.

Aufruf:

:fn:DEBUGZ:fn:demo.24

:fn:DEBUGZ:fn:MSV2D (ab REV 3.00R)

Anschließend müssen die Zellen:

BETR ≈ 7021 mit der Betriebsart (2.4, s.5-1-18)	}
PARTID ≈ 7010 Partner Identifikation	
IDENT ≈ 7000 Eigene Identifikation	
ANZSY ≈ 7022 mit der Anzahl der Syn-Zeichen	

geladen werden

Soll getract werden, so ist die Adresse BIZEI ≈ 5F84 * von C9 mit 00 zu ändern. Anschließend ist das Programm mit dem Monitor zu starten.

*) Adresse abhängig von den LINK-Angaben.

2.7 Testhilfen BSC1/2

- Tracen

Der Verkehr auf der Leitung lässt sich durch Ändern der Adresse "BIZEI" = (*) von "09" auf 00 auf dem Bildschirm anzeigen.

" < " bedeutet Empfangsdaten

Das Zeichen

" > " bedeutet Sendedaten

Das Tracen ist wie folgt möglich:
Laden des Programmes mit DEBUG.

Aufruf:
:fn:DEBUG\$:fn:Anwenderprogramm

Anschließend kann mit dem Monitor die entsprechende Adresse geändert und das Programm gestartet werden

- Demo-Programm

Für den ersten Test der Prozedur steht ein Demonstrationsprogramm bereit. Mit diesem Demoprogramm können Daten von der Tastatur gesendet und empfangene Daten auf Bildschirm ausgegeben werden.

Aufruf:
:fn:DEBUG\$:fn:demo.b23
:fn:DEBUG\$:fn:BSC1/2D (ab REV 3.00R)

Anschließend müssen die Zellen:

BETR ≡ 9021 mit der Betriebsart (2.4, S.5-1-18)	}
PARTID ≡ 9010 Partner Identifikation	
IDENT ≡ 9000 Eigene Identifikation	
ANZSY ≡ 9022 mit der Anzahl der Syn-Zeichen	

geladen werden.

Soll getract werden so ist die Adresse BIZEI ≡ 60D4 von 09 auf 00 zu ändern. Anschließend ist das Programm mit dem Monitor zu starten.

*) Adresse abhängig von den LINK-Angaben.

1. EMULATION 8150 (DIALOG)**1.1 Dialog-Inbetriebnahme****Aufruf**

DIALOG CR

**EINGABEN**

W = Wählleitung oder 2 Draht-Standleitung

S = 4 Draht-Standleitung

J = mit LCG

N = ohne LCG

Technische DatenNetzart: 2-Draht Wählleitung
2-/4-Draht Standleitung

Betriebsweise: hx

Übertragungsprozedur: asynchron, ungesichert,
max. 1200 bit/s

Programmgröße: 5kB + 28kB Puffer

1.2 Beschreibung der Funktionstasten

Funktion	deutsche Standard Tastatur	internat. Standard Tastatur	andere Tasta-turen	Code	original 8150 Tastatur
Einfügen Zeichen	CHAR	CHAR	CTRL +P	10	EFG
Löschen Zeichen	CHAR oder DEL	CHAR oder DEL	CTRL +Q oder DEL	11	LZN
Löschen Zeile			CRTL +S	13	LZL
Zeilenbegrenzung	←	←	CR	OD	
Hardcopy			CTRL +T	14	DR
DÜE*	DÜ	XMT	CTRL +A	01	DÜ
Sonderfunktionen	ESC	ESC	ESC	18	-
Senden-Datei	SEND	SEND	CTRL +U od. SEND	15	-
Empfangs-Datei	ANF	REQ	CTRL +V od. REQ	16	-

Sonderfunktionen

Sonderfunktionen können durch Drücken der "ESC"-Taste eingeleitet werden.

- ESC - Verlassen der Schleife, "Warten auf Verbindung".
 - Verlassen der Schleife, "Warten auf erstes Empfangszeichen" nach vorausgegangenem Senden.
 - Abbruch des Dateitransfers.
- ESC C Löschen des Bildschirms
- ESC E Laden des EDT mit Absetzen von @ (bei internationaler Standard-Tastatur) bzw. § (bei deutscher Standard-Tastatur)
BLOCK ON und @ VDT 13,80
- ESC H @ bzw. § HALT-Kommando für den EDT
- ESC U Ein-/Ausschalten der Klein-/Großumschlüsselung
- ESC N Ein-/Ausschalten der NTA-Funktion
- ESC S Rückkehr in das BS 1
- LINE Abbau der Leitung

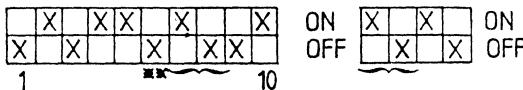
1.3 Hardwarevoraussetzungen

- Modem Anschluß am "ASYNCHRON Interface"
 - ZEICHEN PROM oder LCG (QAS 12)
 - Einstellungen:

- Rückwand Geschw. 1200 Bd

NO PARITY	NO
EVEN PARITY	YES
TWO STP BITS	YES
EXT. ECHO	YES

- Display Logic II



* abweichend von Grundstellung
** Bei Betrieb mit Siemens Modems N10 jedoch in Stellung ON, wenn M1 konstant im EIN-Zustand.

2. Emulation 81512.1 Emulation 8151-InbetriebnahmeEM 8151

- Die Emulation setzt auf die Prozedur MSV1 auf.
- Folgende Dateien müssen auf der FD vorhanden sein:
em 8151, block, em.lcg
- Hilfsprogramm für die Generierung:
:fi:param.1, vorher Blockdatei löschen bzw. für
:f0:exec ci = param.neu. Eingabe wie bei param.1.

Erstellung und Aufruf der Parameterdatei siehe Datei
"help 51.1"
- Start der Emulation: :fi:em 8151.1

Technische Daten

- Netz: 2-Draht-Wählleitung
2-/4-Draht-Standleitung
- Betriebsart: Abrufbetrieb, synchron, hx oder dx,
Takt von der Datenübertragungseinrichtung.
- Prozedur: MSV1
- Übertragungsgeschwindigkeit: max. 4800 Bit/s bei Fernsprech-Wählnetzen, Standleitungen und am DATEX-L-Netz.
- Übertragungscode: USASCII, normierter Modus, 7 Bit mit Paritätsbit ungerade.
- Datensicherung: Zeichenparität ungerade, Blockparität gerade, BCC mit Zeichenparität gesichert.

DFÜ-Testmöglichkeiten

- ON-LINE Test:
Für die Emulation 8151, 8161 steht im BS 2000 ein Terminaltest QFFE der Datensichtstation 8151, 8161 zur Verfügung. Der Test ist mit EXEC QFFE aufzurufen. Bei der Emulation 8151 muß die Funktion RNZ (Anzeige in der Statuszeile) dazu mit ESC+R ausgeschaltet werden.
- OFF-LINE Test:
Als Simulator der DVA kann der K19 verwendet werden. Wird am K19 der Schalter VA betätigt, muß die Prozedur neu aktiviert werden.
Abhilfe:
Nachbildung der 4-Draht-Standleitung, als Adapter zwischen Modemkabel und K19-Eingang.

2.2 Beschreibung der Funktionstasten

Funktion	Standard Tastatur 2 6.610	Standard Tastatur 1 6.610	Tastatur TRANSDA- TA 8151
Cursor nach rechts			
Cursor nach links			
Cursor an den Beginn der folgenden Zeile			
Cursor an dieselbe Stelle der folgenden Zeile			* nicht vorhanden
Cursor an den Beginn der vorhergehenden Zeile			
Cursor an den Bildschirmanfang			
Tabulator rechts			
Tabulator links			
Zeilenbegrenzung			
Einfügen von Zeichen			
Ausfügen von Zeichen			* nicht vorhanden
Zeichen löschen			
Zeile ab Cursor löschen			

Gesamte Zeile löschen	+	FIELD	* nicht vorhanden
Bildschirm ab Cursor löschen	+	DISPL ↓	+
Gesamten Bildschirm löschen	+	DISPL ↕	+
Nachrichtenabschluß		ETX	ETX
Daten übertragen von Zellenanfang bis ETX			
Daten übertragen von Cursor bis ETX	+	+	
Sendeauflorderung	+	+	DVA
Kurznachricht	+	+	FT1
Kurznechricht	+	+	FT2
RNZ ein-/ausschalten	+	+	RNZ
Rücksetzen 6.610			* nicht vorhanden
Rücksetzen Emulation		+	* nicht vorhanden
Verbindung trennen	+	+	VT
Drucken		PRINT	DR
Markieren	+	+	Lichtstift
Leitungsstatus	+	+	* nicht vorhanden

* Zusatzfunktion gegenüber TRANSDATA 8151

2.3 Hardwarevoraussetzungen

- Zulassungsnr.(Fernsprechwählnetz):FTZ A 52 - 10 DEE 983'
- Zulassungsnr.(DATEX-L): FTZ T 35 - 3 1/186/010/80
- LCG oder Zeichen-Prom (OAS 12)
- MODEM-Kabel GS 12
- Einstellungen
- FBG Synchron-Interface
Die FBG muß auf Steckplatz A4 stecken. Die Brücke I2-I3 muß gesteckt sein. Keinen internen Takt verdrahten.
- FBG Display-Logik I
Die folgenden Schalttereinstellungen sind nur erforderlich, wenn die Anzeigenlampen ENQ, ACK und NAK in der Software des Anwenderprogramms verwendet werden.
U39 S1 - ON, S2 - OFF und S9 - OFF
Rest = Grundeinstellung.
- FBG Display-Logik II
U61 S9 - OFF (Abbildung von DFÜ Steuerzeichen)
Rest = Grundeinstellung
- Auf allen anderen FBG'en sind die Standard-Schalterstellungen durchzuführen.
 - Erforderliche Hardware
6.610 mit OAS 7 und 48 kB Speicherausbau.
FBG Synchron-Interface mit GS 13 (Produkt Nr.61055).

3. Emulation 81613.1 Emulation 8161 - InbetriebnahmeEM 8161

- Die Emulation setzt auf die Prozedur MSV1 auf
- Folgende Dateien müssen auf der FD vorhanden sein:
em 8161.1, block, em.lcg
- Hilfsprogramm für die Generierung:
:fn:param.1, vorher Blockdatei löschen bzw. für
:f0:exec ci = param.neu. Eingabe wie bei param.1.

Erstellung und Aufruf der Parameterdatei siehe
Datei "help 61.v3"

Technische Daten

- Netz: 2-Draht-Wählleitung
2-/4-Draht-Standleitung
- Betriebsart: Abrufbetrieb, synchron, hx oder dx,
Takt von der Datenübertragungs-
einrichtung.
- Prozedur: MSV1
- Übertragungs-
geschwindigkeit: max. 4800 Bit/s bei Fernsprech-Wähl-
netzen, Standleitungen und am
DATEX-L-Netz.
- Übertragungscode: USASCII, normierter Modus, 7 Bit mit
Paritätsbit ungerade.
- Datensicherung: Zeichenparität ungerade, Blockparität
gerade, BCC mit Zeichenparität
gesichert.

DFÜ-Testmöglichkeiten

- ON-LINE Test
Für die Emulation 8161 steht im BS 2000 ein Terminaltest
QFFE der Datensichtstation 8161 zur Verfügung. Der Test
ist mit EXEC QFFE aufzurufen.
- OFF-LINE Test
Als Simulator der DVA kann der K19 verwendet werden.
Wird am K19 der Schalter VA betätigt, muß die Prozedur
neu aktiviert werden.
Abhilfe:
Nachbildung der 4-Draht-Standleitung, als Adapter
zwischen Modemkabel und K19-Eingang.

3.2 Beschreibung der Funktionstasten

Funktion	Standard-tastatur 1	Standard-tastatur 2	Tastatur 8161
Cursor Home			
Cursor rechts			
Cursor links			
Cursor an den Zeilenanfang			
Cursor nach oben			
Cursor nach unten			
Cursor an den Beginn der nächsten Zeile			
Cursor an den Beginn der vorhergehenden Zeile			
Tabulator links			
Tabulator rechts			

Funktion	Standard-tastatur 1	Standard-tastatur 2	Tastatur 8161
Zeichen löschen	+ +		
Zeile ab Cursor löschen		+	
Bildschirm ab Cursor löschen		+	
Bildschirm löschen		+	
Zeichen einfügen	+	+	
Zeichen ausfügen	-	-	
Zeile einfügen	+		
Daten übertragen			
Funktions-tasten	+ +
Kurztelegramm	+

Funktion	Standard-tastatur 1	Standard-tastatur 2	Tastatur 8161
Roll up			RU
Rücksetzen der Emulation			RS
Markieren			MAR
Endemarken			EM
Hardcopy			LA1
Verbindung trennen			VA
Wiederanlauf			—
Fehlerstatistik einschalten			—
Rücksetzen 6.610	RESET*	RESET*	—

3.3 Hardwarevoraussetzungen

- Zulassungsnr. (Fernsprechwählnetz):FTZ A 52-10 DEE 983
- Zulassungsnr. (DATEX-L):FTZ T 35-3 1/186/010/80
- LCG oder Zeichen-Prom (OAS 12)
- MODEM-Kabel GS 12
- Einstellungen
- FBG Synchron-Interface
Die FBG muß auf Steckplatz A4 stecken. Die Brücke I2-I3 muß gesteckt sein. Keinen internen Takt verdrahten.
- FBG Display Logik I
Die folgenden Schaltereinstellungen sind nur erforderlich, wenn die Anzeigenlampen ENQ, ACK und NAK in der Software des Anwendungsprogramms verwendet werden.
U39 S1-ON, S2-OFF und S9-OFF
Rest = Grundeinstellung
- FBG Display Logik II
U61 S9-OFF (Abbildung von DFÜ Steuerzeichen).
Rest = Grundeinstellung
- Auf allen anderen FBG'en sind die Standard Schaltereinstellungen durchzuführen.
- Erforderliche Hardware
6.610 mit OAS 7 und 48 kB Speicherausbau.
FBG Synchron-Interface mit GS 13 (Produkt Nr. 61055)

4. Emulation 32704.1 Emulation 3270 - Inbetriebnahme

- Die Emulationssoftware wird im Master beim Einlegen der Diskette automatisch geladen.
Das Laden der Slaves geschieht mittels Software-Modul "MEPLA"
- Hilfsprogramm für Konfigurierung

~~DELETE~~ confil

Eingabe G 4006 CR

- Technische Daten
- 6.610 \longleftrightarrow IBM-Zentralrechner

Betriebsart: hx (bei 4-Draht-Standleitung ist das MODEM auf Duplex zu stellen).

Netzart: 2 oder 4 Draht Standleitung

Übertragungs-
geschwindigkeit: 4800 b/s

Gleichlaufverf.: synchron (BSC-Prozedur)

Übertragungscode: EBCDIC

- 6.610 - 6102 (Slave)

Betriebsart: hx

Übertragungs-
geschwindigkeit: max. 19200 b/s

Gleichlaufverf.: asynchron, LSV1

DFÜ-Testmöglichkeit

Bei Auftreten von Prozedurproblemen besteht die Möglichkeit ein Leitungsprotokoll zu erstellen.
Starten der Ausgabe: G 4003

4.2 Beschreibung der Funktionstasten

CTRL @	00	LOCK RESET
CTRL A	01	PA-TASTE
CTRL B	02	BROADCAST
CTRL C	03	COPY-MODE
CTRL D	04	DUP
	05	FELD LOESCHEN
	06	EINGABE LOESCHEN
	07	TABULATOR LINKS
	08	CURSOR LINKS
	09	TABULATOR RECHTS
CTRL J	0A	FIELD-MARK
	0B	CURSOR DOWN
CTRL L	0C	TABULATOR LINKS AUF GESCH., FELD
	0D	NEW LINE
CTRL N	0E	HOME AUF GESCHÜTZTES FELD (SELECT)
	0F	ZEILE EINFÜGEN
	10	ZEICHEN EINFÜGEN
	11	ZEICHEN LÖSCHEN
CTRL R	12	TABULATOR RECHTS AUF GESCH. FELD
	13	ZEILE LÖSCHEN
CTRL T	14	TEST-ABFRAGE
SEND	15	SEND
CTRL V	16	PF-TASTE
	17	ILLEGAL CHAR
	18	CURSOR RECHTS
	19	BILD SCHIRM LÖSCHEN (CLEAR)
AR	1A	STEUERUNG OFFLINE DATENERFASSUNG
	1B	ILLEGAL CHAR
	1C	CURSOR UP
	1D	CURSOR HOME
	1E	ILLEGAL CHAR
	1F	ILLEGAL CHAR

4.3 Hardwarevoraussetzungen

- Zulassungsnr. FTZ DEE 983
- MODEM-Kabel GS 11
- Einstellungen (6.610 - Master)
 - FBG Synchron-Interface
Die FBG muß auf Steckplatz A4 stecken.
Die Brücke I2 und I3 muß gesteckt sein.
Keinen internen Takt verdrahten.
 - FBG Display Logic I
U39 S1-ON, S2-OFF, S9-OFF
Rest = Grundeinstellung
 - FBG Cluster-Interface
Die FBG ist auf Steckplatz A3 zu stecken.
Einstellung auf TTY-Pegel
Übertragungsgeschwindigkeit wahlweise zwischen
75 und 19200 B/s. Für alle 7 Kanäle muß je
eine Verbindung zur Clock eingelötet sein.
- Rückwand

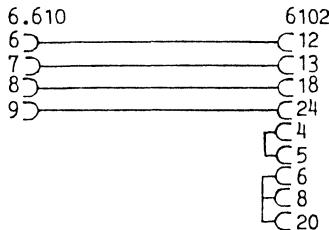
NO PARITY	- NO
EVEN PARITY	- YES
TWO STOP BITS	- YES
EXT. ECHO	- YES

- Einstellungen (6102-Slave)
 - Display Logic I und II siehe Master

NO PARITY	- NO
EVEN PARITY	- YES
TWO STOP BITS	- YES
EXT. ECHO	- YES
AUTO RES	- --
AUTO DSR	- --

SPEED SELECT: 0...7
je nach Ü-Geschw., die auf der FBG Cluster Interface
des Masters eingestellt wurde.

- Kabelbelegung (TTY-Pegel):



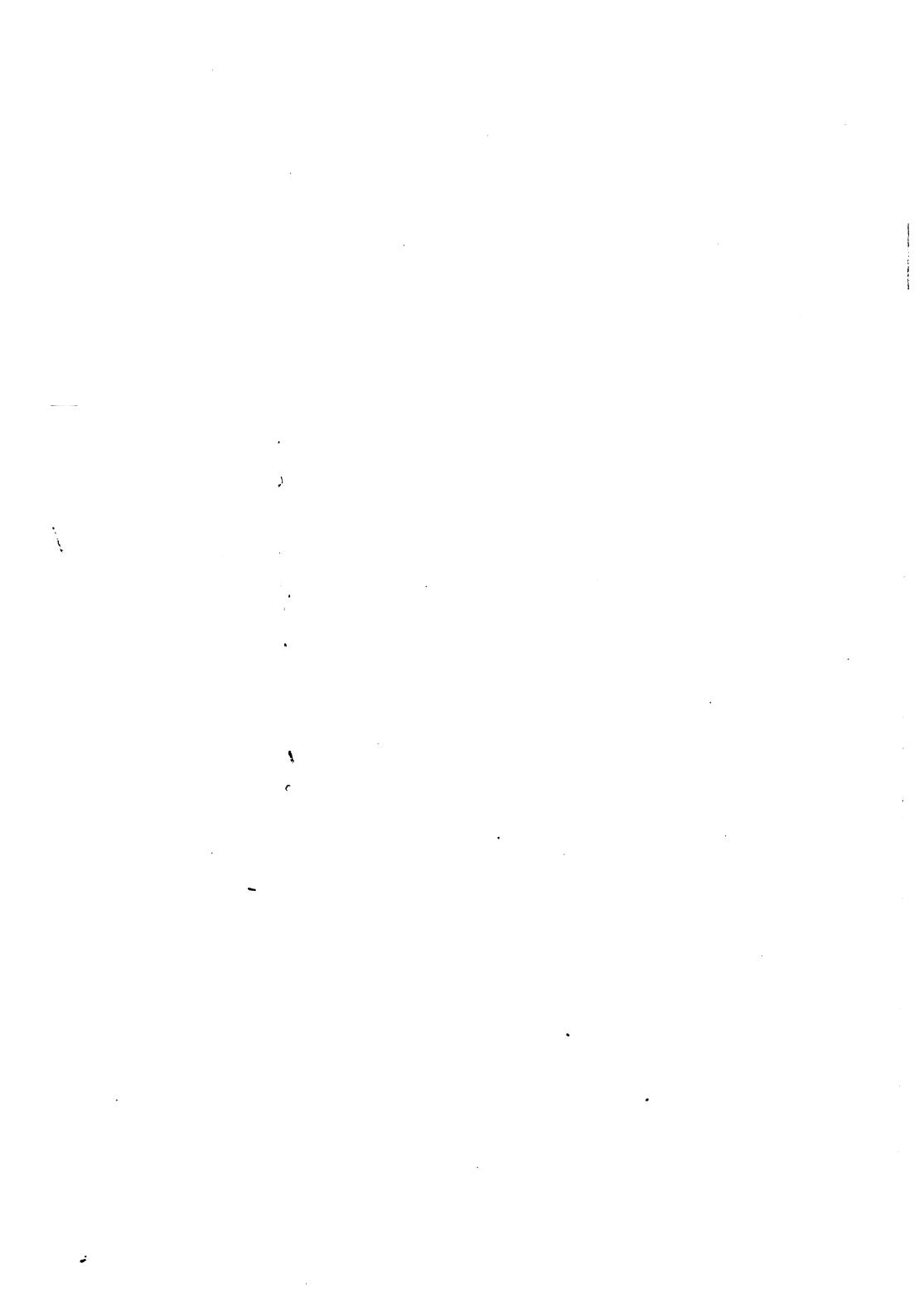
- Erforderliche Hardware

Für den einwandfreien Betrieb Emulation 3270 sind folgende Hardwarevoraussetzungen notwendig:

- Im Bildschirm Computer 6.610 (Master):
 - Mehrfach-Anschaltung 61060 in GS 18
 - MON D in GS 6
- In der Datensichtstation 6102 (Slave):
 - Grundeinheit 6102 in OS 17
 - MON DS in GS 3

4.4 Fehlermeldungen

Meldung	Ursache
"NO CONFIL"	Keine Systemkonfiguration vorhanden, s. Anhang A2 Softwarekonfiguration
"MODEM DOWN"	Modem ist nicht eingeschaltet oder nicht angeschlossen
"ILLEGAL COPY"	Kein Drucker angeschlossen
"ILLEGAL COPY!"	Drucker arbeitet zur Zeit
"ILLEGAL COPY"	Drucker nicht betriebsbereit
"ILLEGAL ADDRESS FOR BROADCAST"	Falsche Adresse bei Broadcast- Nachricht
"COMMUNICATION ERROR, SLAVE TERMINAL n"	Wird an der Steuereinheit ausge- geben, falls bei einem Arbeitsplatz z.B. der Netzstecker gezogen ist.



1. Filetransfer 6.610 - 6.610

1.1 Bedienung

- Laden

Das Filetransferprogramm ist mit dem Aufruf:
:fn:FILE ao:fn:dateiname
:fn:FT610.48/54:fn:dateiname (ab REV 3.00)
zu laden.
n = beliebige Laufwerksnummer
dateiname = wahlfreier Name der Loggingdatei



Achtung:

Die Dateien CODE und block müssen sich auf der gleichen Floppy befinden, auf der die Loggingdatei eröffnet wird.

Nach dem Laden werden folgende Meldungen bzw. Anzeigen abgebildet:

- 1.) "PROZEDUR MSV2 - AUSGABESTAND)"
- 2.) Abbildung der Statuszeile (1.Zeile am Bildschirm)
- 3.) Nur bei Wählverbindungen
"BITTE VERBINDUNG AUFBAUEN" (primary) bzw.
"BITTE WARTEN" (secondary)
- 4.) Nach dem Drücken der Datentaste bzw. bei Standleitungen erscheint in der Statuszeile:
"MSV"
Es kann nun mit dem Filetransfer gearbeitet werden.

- Dialog

Außerhalb der Dateitransferphase können Informationen über Bildschirm ausgetauscht werden.
Ablauf:

1. Eingeben der Nachricht : (max.3 Zeilen)
2. Absenden der Nachricht durch Drücken der "DU" bzw. "XTM" Taste
3. Auf der aktuellen Cursorposition wird das Zeichen "\ " eingetragen und die Nachricht übertragen.

- Dateittransfer

- Senden von Dateien
 - 1.) Taste "SEND" drücken
 - 2.) Am Bildschirm wird "SEN" angezeigt.
 - 3.) Weiter mit Punkt Parameterangaben
- Anfordern von Dateien
 - 1.) Taste "ANF" bzw. "REQ" drücken
 - 2.) Am Bildschirm wird "REQ" angezeigt.
 - 3.) Weiter mit Punkt Parameterangaben.

Parameterangaben

Nach der Ausgabe von "SEN" bzw. "REQ" am Bildschirm sind noch folgende Angaben zu machen:

SEN { Fn } DATEI1, { FN } DATEI2, A
 REQ { In } In H Spassw.name

 |
 Ausgabe am Bildschirm

In,Fn = Dateiart und Laufwerk
 Datei 1 = Dateiname auf der eigenen 6.610
 Datei 2 = Dateiname der Partner 6.610
 H = Hexadezimaldatei (der Inhalt muß
 an keinen Code gebunden sein)
 passw. = Passwort (max. 6 Zeichen)
 name = Anwendernname (max. 3 Zeichen)

Überstrichene Parameter können entfallen, es gilt dann folgende Vorbesetzung

Dateiart und Laufwerk = F0
 ASCII-Datei (bei Angabe Fn) EBCDI-Datei (bei Angabe In)
 ohne Passwort und Anwendernamen.

Unabhängig ob "SENDEN" oder "ANFORDERN" ist immer zuerst die Datei der eigenen 6.610 und dann die Datei der Partnerstation anzugeben.

- Funktionstasten

Abbruch des DÜ-Vorgangs bzw. Rückkehr in das Betriebssystem wenn Anzeige UAN in Statuszeile.



+



Betätigung nur in der Dialogphase möglich. Filetransfer beenden und Rückkehr in das Betriebssystem.



Blankkomprimierung einschalten.
Nicht bei Hexa-Dateien

3

- Anzeigen in der Statuszeile

"MSV" → Die MSV2 Prozedur ist betriebsbereit

"UEF" → Abbruch und Wiederanlauf wegen MSV2-Fehler

"UAN" → Es wird von dieser Station gerade gesendet

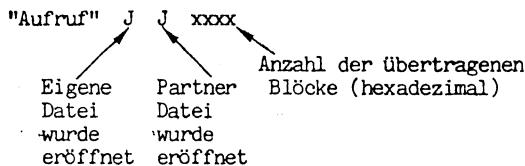
"ABW" → Der Partner hat getrennt

"CON" → Contention Fall: Es wird von beiden Stationen gleichzeitig ein Sendewunsch abgegeben.

"BC" → Blankkomprimierung wurde eingeschaltet.

- Betriebsmeldungen

- . Meldung während des Transfers



Erscheint anstelle des "J" ein "N" bedeutet dies, daß die Datei nicht eröffnet werden kann.

- . Meldung nach Beendigung des Transfers.

Aktive Station

"Aufruf", J J → xxxx (Senden)

oder

"Aufruf", J J ← xxxx (Empfang)

xxxx = Anzahl der Blöcke (Darstellung Hexadezimal)

Passive Station

z.B.:

E A B dateiname name J → xxxx (Empfang)

oder

S A 'B dateiname name J ← xxxx (Senden)

1) 2) 3) 4) 5) 6) 7)

1) Aufforderung zum Senden bzw. Empfangen

2) Dateiart A=ASCII, H=Hex

3) keine Bedeutung

4) Name der zu eröffnenden Datei

5) Kurzzeichen des Anfordernden

6) Datei wurde eröffnet

7.) Anzahl der DU-Blöcke

1 DU-Block entspricht bei:

Standard-Intel Dateien 4 Sätzen a 256 Byte.

ECMA TC54(IBM) Dateien 8 Sätzen a 128 Byte.

Hexadezimal Dateien 4 Sätzen a 256 Byte.

1.2 Hardwarevoraussetzungen

- FTZ Zulassung.
Der Anschluß ist an folgende Netze genehmigt:
Fernsprechwählnetz, Direktrufnetz mit

FTZ A 52 - 10 DEE 983

DATEX-L-Netz

FTZ T 35 - 3 1/186/010/80

- Hardware
 - . 6.610 mit OAS 7, 48kB und Monitor ab GS 6
 - . FBG Synchroninterface mit GS 13
 - . Modemanschlußkabel mit GS 12
 - . Datenübertragungseinrichtung
- Einstellungen
 - . FBG-Synchroninterface
FBG auf Einbauplatz A4
Brücke I2-I3 muß gesteckt sein
keinen internen Takt.
 - . FBG Display Logic I

U39	<table border="1"> <tr><td>X</td><td></td><td>X</td><td>X</td><td></td><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td><td></td><td></td><td>X</td><td>X</td><td>X</td></tr> <tr><td>1</td><td></td><td></td><td></td><td>10</td><td></td><td></td></tr> </table>	X		X	X		X	X	X	X			X	X	X	1				10			ON	<table border="1"> <tr><td></td><td>X</td><td>X</td><td></td></tr> <tr><td>1</td><td></td><td></td><td>4</td></tr> </table>		X	X		1			4	U79
X		X	X		X	X																											
X	X			X	X	X																											
1				10																													
	X	X																															
1			4																														
		OFF																															

- . Auf allen anderen FBG'en sind die Standardschaltereinstellungen durchzuführen.
- . Datenübertragungseinrichtung
S1, S2 gesteuert von DEE
hx für Wahl oder 2-Draht Standleitung
dx für 4-Drahtleitungen
Takt von der Datenübertragungseinrichtung

Manueller Betrieb wenn Sprechverkehr erwünscht ist, sonst Automatikbetrieb.
(Gilt nicht für DATEX-L)

Anmerkung:

Die Einstellungen an den Datenübertragungseinrichtungen sind von der Deutschen Bundespost auszuführen.

- Anschluß
 - . Standverbindung 2/4 Draht max. 4800 B/s
 - . Wählverbindung 2 Draht max. 4800 B/s
 - . Wählverbindung 4 Draht max. 4800 B/s(DATEX-L)
 - . Datenübertragungseinrichtung entsprechend der Netzart.

1.3 Softwarevoraussetzungen

- Zusatzdateien
 - . Datei "CODE"
 - . Datei "block"
- Einstellung der Zusatzdateien
 - . Datei "CODE"

Diese Datei wird mit Programm FCODE erzeugt.
Aufruf: FCODE ↴ A0 = CODE

In dieser Datei kann ein Passwort hinterlegt werden. Wird kein Passwort benötigt, so muß die Datei trotzdem erzeugt werden. In diesem Fall ist auf die Frage nach dem Passwort nur CR einzugeben.

- . Datei "block"

Diese Datei wird mit Hilfe eines EXEC-Files erzeugt.
Aufruf: execuci = param.2 neu

Die Floppy Disk muß dazu in Laufwerk 0 eingelegt werden.

Das Programm ist selbsterklärend und fordert im Dialog die Angaben an.

1.4 Fehlermeldungen

Außer den BS610 - Fehlermeldungen können noch folgende Meldungen vorkommen:

- ERROR A1 = Der Filetransfer wurde asynchron
- ERROR A2 = Passwortfehler
- ERROR A3 = Abbruch durch den Anwender
- ERROR A4 = Eine als ASCII-Datei aufgerufene Datei enthält Zeichen die nicht im ASCII-Zeichenvorrat enthalten.
- ERROR A5 = Satzlänge bei IBM-Dateien 128

- ERROR 64,63 = Partner vermutlich unklar
- ERROR 68 = wie ERROR 64 oder Floppy im Partnersystem defekt

- ERROR 6A = V24-Fehler (M1-signalgestört oder M2, M5 nicht innerhalb von 2 Sekunden nach einer Empfangs-Sendeumschaltung in der richtigen Polarität).

- ERROR 6C = Leitung ist so gestört, daß Zeichenverfälschungen auftreten, die von der Prozedur nicht mehr abgefangen werden können.

Tritt vor der Fehlermeldung ein Zeichen:
" > " auf, so ist dies ein Hinweis auf die Partnerstation

MSV2-Fehler werden im Filetransfer angezeigt mit:
ERROR 6x (x=Größe von 1-F soweit in MSV2 verwendet).

1.5 Testhilfen

1.) NAK - Anzeigen

Nach Betätigen der Tastenfolge
"ESC + S"

Wird die Anzahl der empfangenen bzw. gesendeten
NAK's angezeigt.

Nochmaliges Drücken schaltet die Anzeige wieder
ab.

2.) Tracen

Der Verkehr auf der Leitung lässt sich durch
Ändern der Adresse BIZEI (89D7) von
"C9" auf "00"
auf dem Bildschirm anzeigen.

Das Zeichen: " < " bedeutet Empfangsdaten
" > " bedeutet Sendedaten.

Das Tracen ist wie folgt möglich:

Laden des Programms mit DEBUG.

:fn:DEBUG\$:fn:FILE AO=:fn:dateiname

:fn:DEBUG\$:fn:FT610.48/54 AO=:dateiname (ab REV 3.00)

Anschließend kann mit dem Monitor die entsprechende
Adresse geändert und das Programm gestartet werden.

2. Filetransfer 6.610 - BS2000

2.1 Bedienung

- Laden

:fn:FILE AI = BS2000
:fn:FILE.48/54 AI=BS2000 (ab REV 3.00)

n = Laufwerksnummer

Achtung: Die Dateien BS2000
MSP80.P2
block
FILE
kenn
müssen auf der gleichen Floppy Disk sein.

Nach dem Laden werden folgende Meldungen bzw. Anzeigen abgebildet.

- 1) "PROZEDUR MSV1 AUSGABESTAND"
- 2) "Abbildung der Statuszeile
(1 Zeile am Bildschirm)"
- 3) Nur bei Wählleitung
"BITTE VERBINDUNG AUFBAUEN"
- 4) Nach dem Drücken der Datentaste bzw. bei Standleitungen erscheint in der Statuszeile:
"LTG"

Der Anwender wird nun zur Eingabe des Logon-kommandos entsprechend den 2000 Konventionen aufgefordert.

- Betrieb

Nach erfolgreicher Durchführung des Logonkommandos werden dem Anwender die möglichen Funktionen angezeigt. Zum Aufrufen der Funktionen stehen folgende Tasten zur Verfügung:

Taste

Funktion



Senden Dateien von 6.610 zum BS2000



Anfordern Dateien vom BS2000 zur 6.610



Ausgeben eines Inhaltsverzeichnisses (FSTAT)



Löschen einer Datei im BS2000



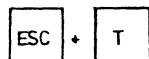
Dialogbetrieb. Es wird ein einfaches Zeilenterminal ohne Formatsteuerung simuliert.



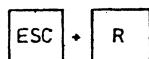
Rücknahme von Funktionen



Abbruch der Verbindung (ohne LOGOFF) (Nur nach Betätigen einer Funktions-taste ausgenommen der Funktion D).



Verbindungsabbau mit LOGOFF
(nur wenn keine Funktion ausgewählt)



Beendigung des Dialogbetriebes.



Absenden der Eingaben im Dialog-
betrieb
(Es darf kein ETX gegeben werden)

Die erforderlichen Angaben zur Ausführung der einzelnen Funktionen werden vom Programm in Dialog erfragt.

2.2 Hardwarevoraussetzungen

- FTZ Zulassung

Der Anschluß ist an folgende Netze genehmigt.
Fernsprechwählnetz, Direktrufnetz mit

"FTZ A 52 - 10 DEE 983

DATEX-L-Netz

"FTZ T 35 -3 1/186/010/80

- Hardware
 - . 6.610 mit OAS 7, 48 kB und Monitor ab GS 6
 - . FBG Synchroninterface mit GS 13
 - . Modemanschlußkabel mit GS 12
 - . Datenübertragungseinrichtung
 - . DFÜ-EPROM
- Einstellungen
 - . FBG auf Einbauplatz A4
Brücke I2-I3 muß gesteckt sein
keinen internen Takt.
 - . FBG - Display Logik I

	X	X	X	X	X	X	
1	X	X		X	X	X	10

ON
OFF

	X	X	
1			4

U39

U79

- Auf allen anderen FBG'en sind die Standardschaltereinstellungen durchzuführen.
- Datenübertragungseinrichtung

S1, S2 gesteuert von DEE
hx für Wahl oder 2-Draht Standleitung
dx für 4-Drahtleitungen
Takt von der Datenübertragungseinrichtung

Manueller Betrieb wenn Sprechverkehr erwünscht ist, sonst Automatikbetrieb. (Gilt nicht für DATEX-L)

Anmerkung:

Die Einstellungen an den Datenübertragungseinrichtungen sind von der Deutschen Bundespost auszuführen.

- 7.000
Auf der 7.000 Seite muß die entsprechende Hardware für den Anschluß einer Datensichtstation 8151 vorhanden sein.

- Anschluß

- Standverbindung 2/4 Draht max. 4800 B/s
- Wählverbindung 2 Draht max. 4800 B/s
- Wählverbindung 4 Draht max. 4800 B/s
(DATEX-L)
- Datenübertragungseinrichtung entsprechend der Netzart.

2.3 Softwarevoraussetzungen

- Zusatzdateien auf 6.610 Seite

- . Datei " BS2000 "
- . Datei " MSP80.P2 "
- . Datei " block "
- . Datei " kenn "

Diese Dateien müssen auf der gleichen Floppy Disk vorhanden sein auf der sich auch die Programmdatei "FILE" befindet. (FILE.48/54 ab REV 3.00)



- Erstellung der Zusatzdateien

Vom Anwender muß nur die Datei "block" erstellt, bzw. darf vom Anwender verändert werden.

- . Datei "block"

Diese Datei wird mit Hilfe eines EXEC-Files erzeugt.

Aufruf: exec ci = param 3. neu

Die Floppy Disk muß dazu im Laufwerk Ø eingelegt werden.

Das Programm ist selbsterklärend und fordert im Dialog die Angaben an.

Ab REV 3.00 ist die Datei "block" mit dem Programm MSV2P zu erzeugen.

Es kann jedes Laufwerk benutzt werden.

- Zusatzsoftware auf der 7.000 Seite

- . BS2000: Betriebssystem BS2000 ab Vers. 5.0

Editor EDT Vers. C11

oder

Editor EDT Vers. 1.12

2.4 Fehlermeldungen

- Senden

Meldung	Ursachen
ABBRUCH WEGEN ERÖFFNUNGSFEHLER	Dateityp falsch Dateiname falsch Datei nicht vorhanden Laufwerk unklar
DATEI SCHON VORHANDEN	Dateiname dem BS 2000 bekannt, anderen Namen wählen
ABBRUCH WEGEN FLOPPYFEHLER. ES WIRD BIS ZUR FEHLERHAFTEN STELLE BEIM PARTNER ABGESPEICHERT	Zugriffsfehler beim Dateilesen. Der Filetransfer fährt wie im Falle Dateiende (alles Übertra- gen) fort.
E4 : ES WURDE CR EINGEFÜGT	Bei Intel-Dateien: Der Datensatz ist länger als 256 Byte (bevor ein CR auftritt) Bei ECMA IC 54 Dateien. Eine J-Datei wurde mit I aufgerufen. Nach jedem Satz wird CR eingefügt.
E5 : IM DATENSATZ SIND VERBOTENE ZEICHEN ENTHALTEN	Die Datei kann in dieser Form nicht übertragen werden. Die Übertragung wird vorzeitig beendet.(Abbruch wegen Floppyfehler !)

- Empfangen

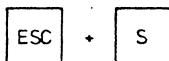
Meldung	Ursache
ABBRUCH WEGEN ERÖFFNUNGSFEHLER	Dateityp falsch Dateiname falsch Datei nicht vorhanden Laufwerk unklar
FEHLER BEIM LESEN DER DATEI oder DATEI IST LEER	Datei im BS 2000 nicht vorhanden, bzw. leer
- Löschen	
Meldung	Ursache
DATEI IST NICHT IM KATALOG	falsche Namenseingabe oder Datei schon gelöscht
DATEI NICHT VORHANDEN	wie vorher
FEHLER BEIM LÖSCHEN DER DATEI	Passwortfehler. Der Anwender muß ein Passwort nennen.
- Sonstige	
Fehler A1, A2	MSP80.P2 ist zerstört.

Außer den vorgenannten Meldungen werden noch
Meldungen des BS1000/BS2000 und des BS610
angezeigt.

2.5 Testhilfen

NAK-Anzeigen

Nach Betätigen der Tastenfolge:



wird die Anzahl der empfangenen bzw. gesendeten
NAK's angezeigt.

Nochmaliges Drücken der Tasten schaltet die
Anzeige wieder ab.

Anmerkung: Die Funktion ist nur
möglich, wenn keine
Funktion ausgewählt ist.

3. EMX 10

3.1 Bedienung

- Laden

Das Programm ist wie folgt zu laden:
:fn:emx10_S0=:fn:dateiname

n = beliebige Laufwerksnummer
dateiname = Standard Output-Datei
Überstrichene Angabe kann entfallen.



- Escape - Funktionen:

ESC I = Programm initialisieren, d.h. in den Ausgangszustand bringen

ESC E = Ändern der Empfangszuweisung

ESC S = Ändern der Sendezuweisung

ESC T = Telex-Betrieb ein- und ausschalten

ESC Q = Fehlermeldung in der Systemzeile löschen

ESC H = Programm beenden und Rückkehr ins BS 610

ESC ESC = Escape-Code senden

- Sendezuweisung: Senden von:
 - S = :CI: ≈ Tastatur (Standard)
 - S = :Fx:Dateiname ≈ der angegebenen Datei im Laufwerk Nr. x
 - S = :Tx:Dateiname ≈ der angegebenen Textdatei im Laufwerk Nr. x
 - S = /SI ≈ der als Standard-Output zugewiesenen Datei

 - Empfangszuweisung: Empfang durch:
 - E = :CO: ≈ Bildschirm (Standard)
 - E = :Fx:Dateiname ≈ angegebene Datei im Laufwerk Nr. x
 - E = :Tx:Dateiname ≈ angegebene Textdatei im Laufwerk Nr. x
 - E = /SO ≈ Standard-Output-Datei

Jede Eingabe muß mit CR abgeschlossen werden.
CR an erster Stelle bewirkt Rückkehr zur
Standardzuweisung. Mit der DEL Taste kann die
Eingabe korrigiert werden. DEL an erster Stelle
macht ESC+S bzw. ESC+E rückgängig.

- Steuerzeichen

Wirkung von empfangenen bzw. eingegebenen Steuerzeichen am Bildschirm.

Zeichen	Hex. Code	Taste:	Wirkung:
CR	0DH		Cursor an Zeilenanfang
LF	0AH		Cursor springt in die nächste Zeile
BEL	07H		akustisches Signal
EM	19H	+	Löschen des Bildschirms außer Systemzeile, Cursor am Beginn der 2. Zeile
GS	1DH		Cursor am Beginn der 2. Zeile (1 Zeile=Systemzeile)
FF	0CH		Roll up

Alle Steuerzeichen, auch NIL, werden als Blanks dargestellt.

- Weitergabe von Steuerzeichen auf Floppy-Disk und Drucker

CR, LF und FF werden direkt weitergegeben. BEL wird ignoriert, alle übrigen Steuerzeichen werden als Blanks weitergegeben.

- Senden von Steuerzeichen

Alle Steuerzeichen werden eins zu eins an die EMX 1010 weitgesendet.

- Bedienungshinweise für Betrieb mit EMX 1010

• Lokalbetrieb

Betriebseinleitung	⊕ <uhrzeit>
Betriebsfunktion	f: l+
Bestätigung	ok
Eingaben	<u>Text</u>
Ende des Lokalbetriebes	+++
Betriebsfunktion	f: ± - keine weitere Funktion
Betriebsende	<uhrzeit> ende

• Kommandobetrieb

Betriebseinleitung	⊕ <uhrzeit>
Betriebsfunktion	f: k+
Eingabe eines Kommandos	\

kmd:msg_ <na-ident> _cncl+

Löschen einer Nachricht

msg_ <na-ident> _dump_ <an> +

Aussenden einer Kopie

msg_ <na-ident> _cncl_ <an> +

Löschen einer Adresse

msg_ <na-ident> _rte_ <an> <neu-an> +

Ändern von Empfängeradressen

Bestätigung	ok
Betriebsfunktion	f:± - keine weitere Funktion
Betriebsende	<uhrzeit> ende

• Speicherbetrieb

Betriebseinleitung	<u>Ø</u>	<uhrzeit>
Betriebsfunktion	f: <u>s+</u>	
Adressierung	an: <u>Rufnummer</u> <u>Rufname</u> <u>Verteiler+</u>	und/oder und/oder
Absenderangabe	von: <u>Name, Telefon</u>	
Nachrichtenkopf	< firma > < na-ident > < datum > < uhrzeit >	
Texteingabe	<u>Text</u>	
Nachrichtenende	<u>+++</u>	
Sendeanweisung	anw: <u>sh+</u> <u>se+</u> <u>snt+</u> <u>ln+</u>	höchste Priorität einfache Priorität senden zur Niedertarifzeit löschen Nachricht
Bestätigung	ok	
Betriebsfunktion	f:+	-keine weitere Funktion
Betriebsende	< uhrzeit > ende	

• Abkürzungen:

< firma >	: == Anlagenidentifikation, Zeichenfolge bis zu 20 Zeichen
< datum >	Datum in Jahr, Monat, Tag
< uhrzeit >	: == Anlagenzeit in Stunden, Minuten
< na-ident >	: == Nachrichtenidentifikation bestehend aus: < nst > / < laufnr >
< nst >	: == Nummer der eigenen Nebenstelle (zweistellig)
< laufnr >	: == Nachrichteneingangslaufnummer (vierstellige Dezimalzahl)
< nr >	: == Leitungsnummer (zweistellig)
< an >	: == Adressangabe: Rufname (maximal 8 stellige Zeichenfolge beginnend mit einem Buchstaben) oder Rufnummer (maximal 16 stellige Ziffernfolge) oder Nebenstellennummer (zweistellige Dezimalzahl)
< neu-an >	

Die unterstrichenen Angaben sind die an der 6.610 zu tätigen Eingaben für den Betrieb mit der EMX 1010.

3.2 Hardwarevoraussetzungen

- 6.610
 - . Grundeinheit mit OAS 7, 48 kB-Speicherausbau
internes Floppy-Disk Laufwerk 256 kB und
Monitor D ab Rev. GS 6
 - . Bei Modemanschluß: Modemkabel (S26391-F17-
GS 12)
 - . Bei Direktanschluß: Direktanschlußkabel (Kabel)
muß vom EMX 1010 Seite bereit gestellt werden.
 - . DFÜ-PROM (kein LCG)
 - . Drucker

- Einstellungen

- . Display Logik I

<input checked="" type="checkbox"/>							
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>				
1		U39		10	ON	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

ON
OFF

<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>		
1	U79	4

- . CPU

U54: Mit den Schaltern 3-8 muß die
richtige Geschwindigkeit für den
jeweiligen Drucker eingestellt
sein.

- . Auf allen anderen FBG'en sind die Standard-
schaltereinstellungen durchzuführen.

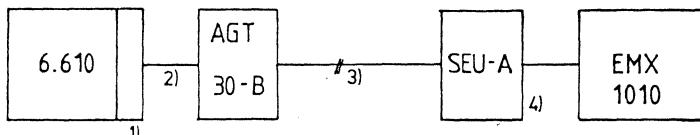
- . Rückwand

SPEED SELECT :	300 Bd
NO PARITY :	NO
EVEN PARITY :	YES
TWO STOP BITS :	YES
EXT ECHO :	NO

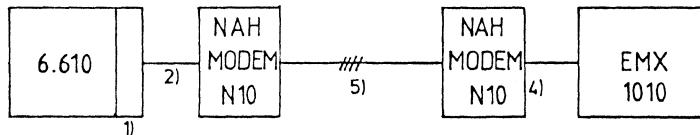
- Anschluß

Anschluß über Datenübertragungseinrichtungen

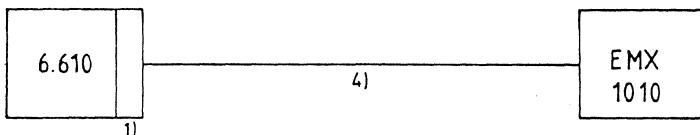
1)



2)



3) Direktanschuß



1) = Asynchron Interface

2) = V24 Anschlußkabel (S26391-F17-~~GS~~12)

3) = Private 2 Draht-Standleitung

4) = Anschlußkabel für EMX 1010 (Kabel wird von EMX 1010 Seite bereitgestellt)

5) = private 4 Draht-Standleitung

3.3 Softwarevoraussetzungen

- 6.610
 - . Programm EMX10
- EMX 1010
 - . Betriebssystem der EMX 1010

3.4 Fehlermeldungen

Außer den EMX 1010 Fehlermeldungen (siehe EMX 1010
Beschreibung Best.-Nr. A22238-K30-A100-*19)
werden nur Meldungen des BS 610 benutzt.



KAPITEL 1	BRÜCKEN UND SCHALTER EINSTELLUNG	Seite
1.	Externe Einstellungen	6-1-1
2.	FBG Floppy-Disk-Drive	6-1-3
3.	FBG Display Logic I	6-1-4
4.	FBG Display Logic II	6-1-5
5.	FBG CPU	6-1-6
6.	FBG RAM	6-1-7
7.	FBG PROM	6-1-8
8.	FBG CLUSTER INTERFACE	6-1-9
9.	FBG SYNCHRONOUS INTERFACE	6-1-10
10.	Abweichungen von der Schaltergrundeinst.	6-1-11/12
10.1	Drucker	.
10.2	TDS	.

KAPITEL 2 EINBAUPLÄTZE

1.	Einbauplätze der Flachbaugruppen (FBG)	6-2-1
2.	Einbauplätze der steckbaren Bausteine	6-2-2

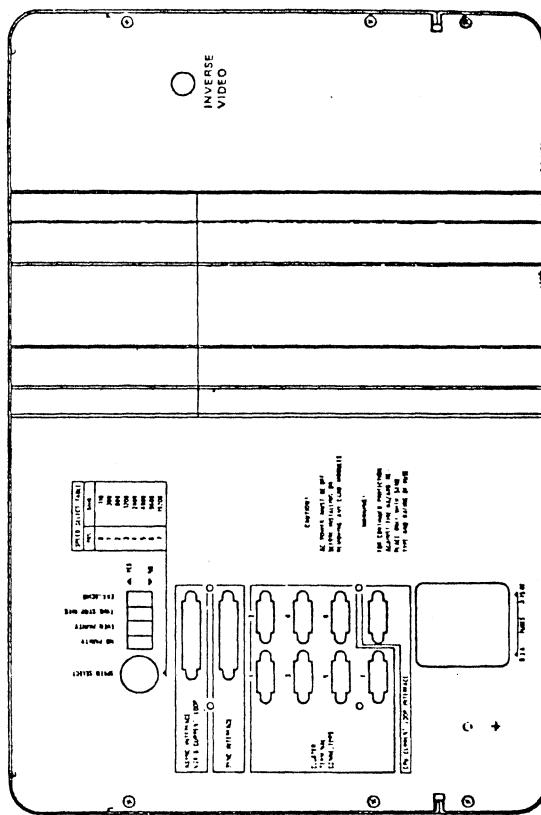
KAPITEL 3 STECKER UND KABELBELEGUNGEN

1.	Steckerbelegungen	6-3-1
1.1	ASYNC. INTERFACE	.
1.2	SYNC. INTERFACE	.
1.3	CPU CURRENT LOOP	6-3-2
1.4	CLUSTER TERMINALS CONNECTORS	.
2.	Kabelbelegungen	6-3-3
2.1	Modemkabel	.
2.2	Kabel für MEPLA,	.
2.3	Druckerkabel	6-3-4

EXTERNE EINSTELLUNGEN

1. Externe Einstellungen

Die sechs Einstellmöglichkeiten auf der Rückseite des Terminals sind dafür da, um die Anzeigen des Terminals mit dem System, in welchem es arbeitet, in Einklang zu bringen.



EXTERNE EINSTELLUNGEN

SPEED SELECT bestimmt die Übertragungsrate des asynchronen Interface in Baud. Die Tabelle gibt die einer Schalterstellung entsprechende Geschwindigkeit wieder.

NO PARITY; Schalter muß auf NO gestellt werden, wenn die Empfangsdaten auf Parity geprüft und die Sendedaten mit einem Parity versehen werden sollen.

EVEN PARITY wechselt zwischen gerader und ungerader Parität. Der Schalter ist nur aktiv, wenn der Schalter NO PARITY in Stellung NO ist.

TWO STEP BITS; Schalter muß so eingestellt werden, daß er mit der Anzahl der im System benutzten STOP BITS übereinstimmt.

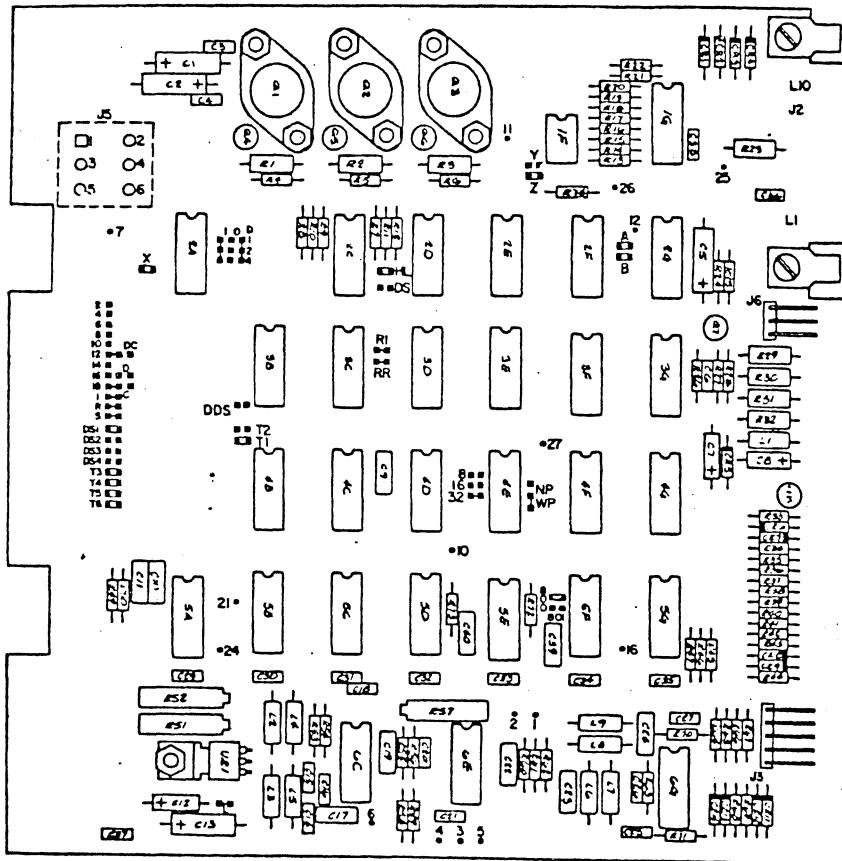
EXT. ECHO; Schalter muß bei Halbduplexbetrieb in Stellung NO geschaltet werden.

INVERSE VIDEO kann durch Betätigen des entsprechenden Schalters auf der FBG Display-Logic II erreicht werden. Der Schalter läßt sich durch ein Loch in der rechten Rückwand erreichen.

HINWEIS:

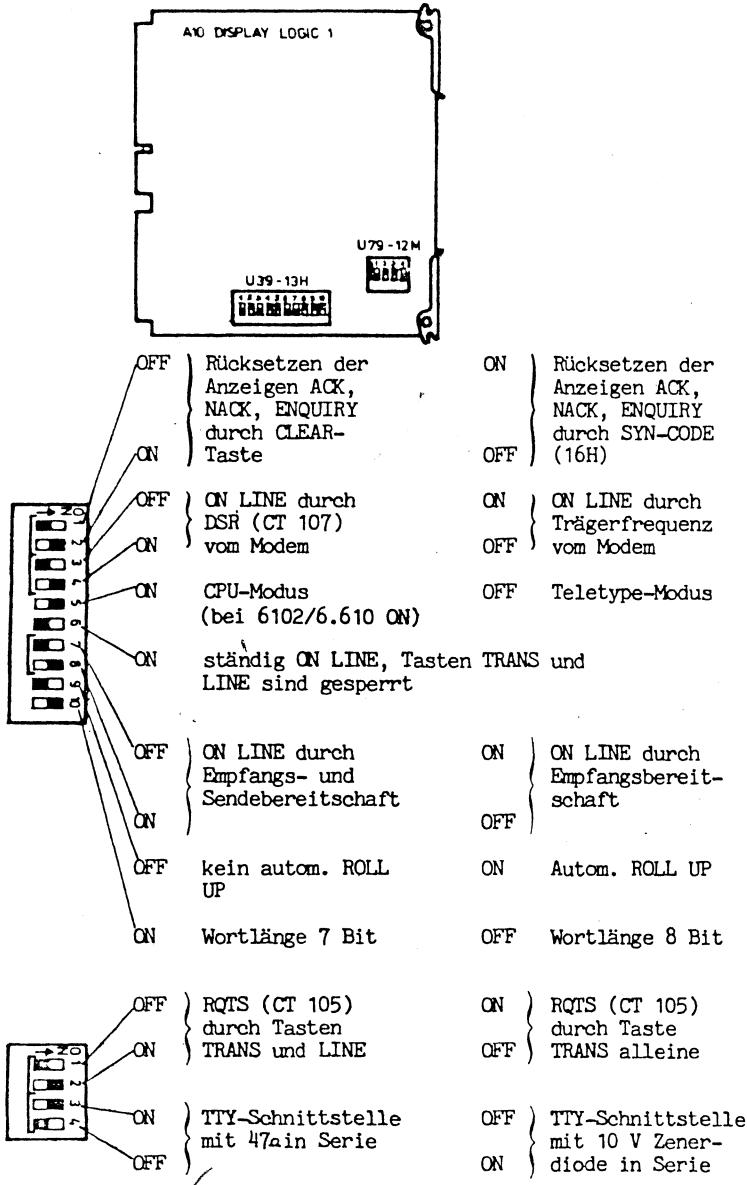
Das Gerät ist immer auszuschalten,
bevor Einstellungen vorgenommen werden.

2. FEG Floppy-Disk-Drive



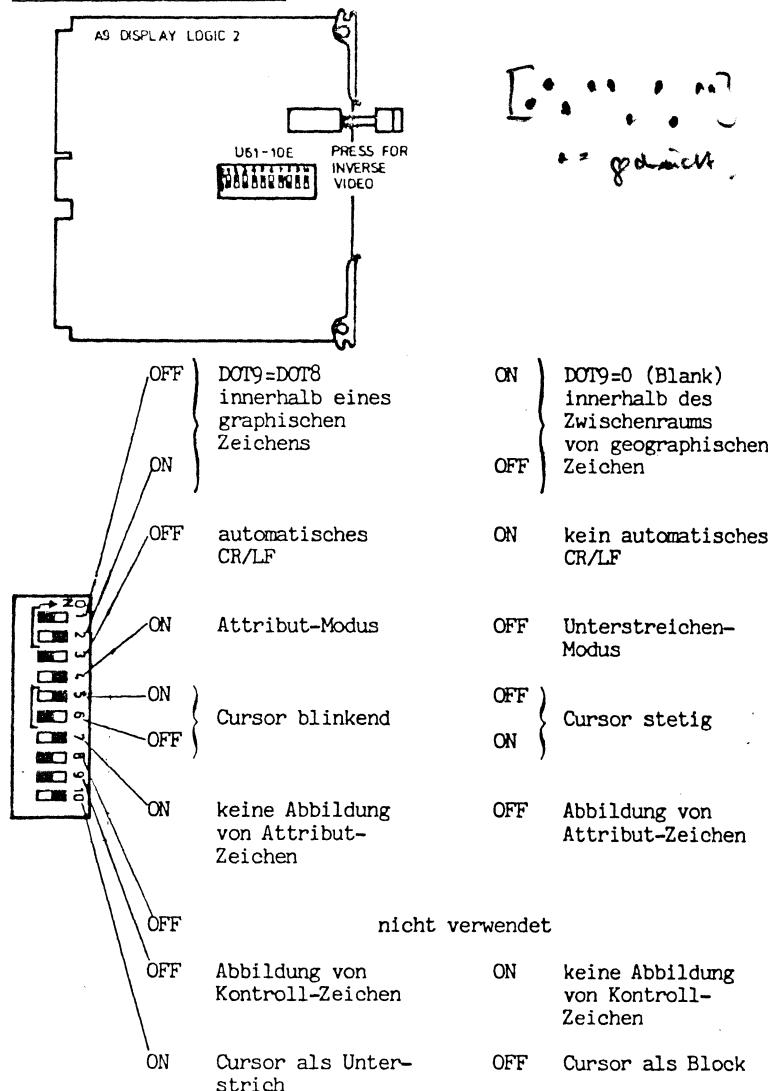
FBG DISPLAY LOGIC I

3. FBG Display Logic I



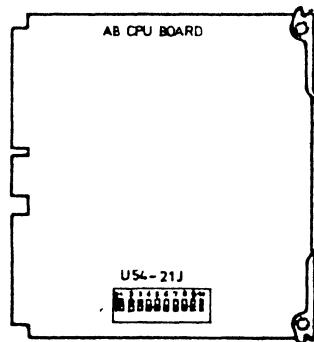
FBG DISPLAY LOGIC II

4. FBG Display Logic II

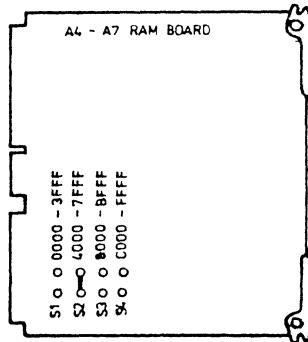


FBG CPU

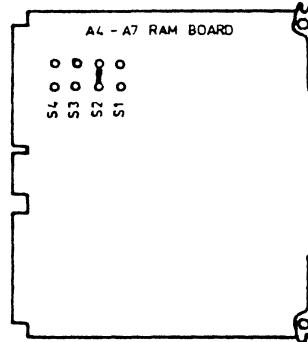
5. FBG CPU



ON	Wortlänge 7 Bit	OFF	Wortlänge 8 Bit
ON	Parityprüfung	OFF	keine Parityprüfung
ON	9600 Bit/s		
ON	4800 Bit/s		
ON	2400 Bit/s		
ON	1200 Bit/s		
ON	600 Bit/s		
ON	300 Bit/s		
ON	75 Bit/s	ON	110 Bit/s
ON		OFF	

6. FBG RAM

FBG RAM 16 kB (TAN 960 315)



FBG RAM 16, 32, 48, 54 kB

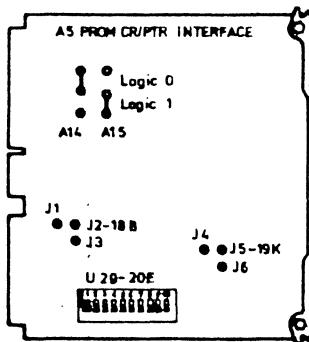
Adress-Schalter	Startadresse	Startadresse**
S1	0000*	2800
S2	4000	6800
S3	8000	A800
S4	0000	E800

* Wird von der CPU benutzt.

** nur 54 kB

FBG PROM

7. FBG PROM



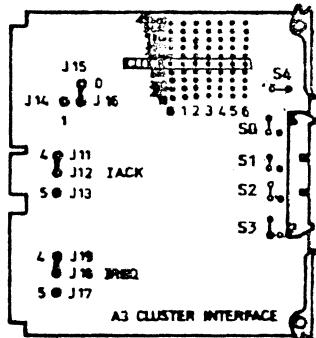
HINWEIS:

Zwischen I1, I2 und I3 bzw. zwischen I4, I5 und I6 dürfen sich keine Brücken befinden. Alle Schalter von U29 müssen in "OFF"-Position sein.

Speicherbereich	A14	A15
0000 - 3FFF *	0	0
4000 - 7FFF	1	0
8000 - BFFF	0	1
C000 - FFFF	1	1

* Wird von CPU benutzt.

8. FBG Cluster Interface



Auf der FBG Cluster Interface muß für jedes einzelne der 7 Interfaces die Übertragungsgeschwindigkeit eingestellt werden. Die Einstellung erfolgt jeweils durch eine Wrapverbindung zwischen dem Stift "Clock" und dem Stift mit der gewünschten Übertragungsgeschwindigkeit.

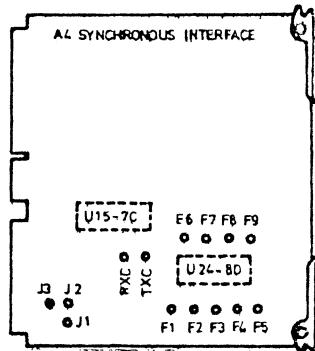
Folgende Verbindungen sind notwendig

- Brücken: I11 - I12,
- I15 - I16 und I18 - I19
- S0S3 Umschaltung TTY - V24 Pegel
(TTY-Pegel, Einstellung wie abgebildet)
- S4 Umschaltung 75 - 100 Baud
(75-Baud, Einstellung wie abgebildet)

Einbauplatz A3.

FBG SYNCHRONOUS INTERFACE

9. FBG Synchronous Interface



für Asynchron-Betrieb mit der FBG SYN. INTERFACE
müssen die Stifte RXC und TXC mit einem der Stiften
F1 bis F9 verbunden werden (Takteinstellung).

F1 = 19200 Bit/s
F2 = 9600 "
F3 = 4800 "
F4 = 2400 "
F5 = 1200 "
F6 = 600 "
F7 = 300 "
F8 = 150 "
F9 = 75 "

ABWEICHUNGEN VON DER SCHALTER-GRUNDEINSTELLUNG

10. Abweichungen von der Schalter-Grundeinstellung

10.1 Drucker

6322: FBG CPU
* Geschw. 600 Bd

X	X					X			X
		X	X	X	X		X	X	

1

ON
OFF

6324: FBG CPU
* Geschw. 1200 Bd

X	X				X			X	
		X	X	X		X	X	X	

1

ON
OFF

6327: FBG CPU
* Geschw. 9600 Bd

X	X	X						X	
			X	X	X	X	X	X	

1

ON
OFF

8106: FBG Display Logic I
* ständig ON LINE

	X		X	X	X		X	X	X
1		X				X		X	

1

ON
OFF

Rückwand Geschwindigkeit 9600 Bd

NO PARITY NO
EVEN PARITY YES
TWO STOP BITS YES
EXT.ECHO NO

10.2 TDS

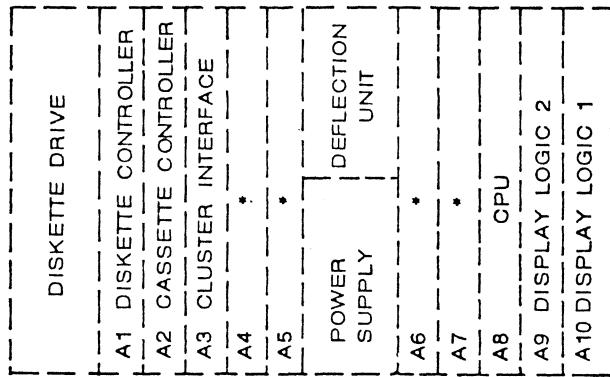
FBG Display Logic I
* Automatischer
ROLL UP

	X		X	X		X	X	X	X
1		X			X	X			

ON
OFF

EINBAUPLÄTZE DER FLACHBAUGRUPPEN

1. Einbauplätze der Flachbaugruppen

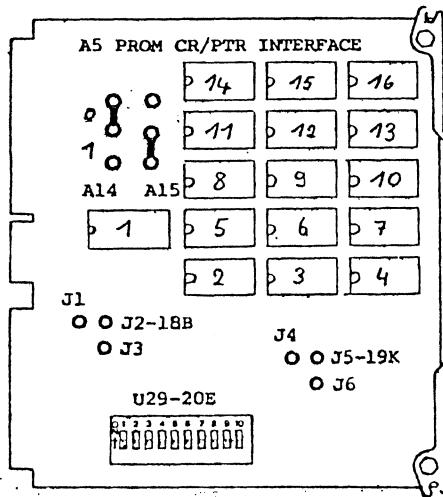
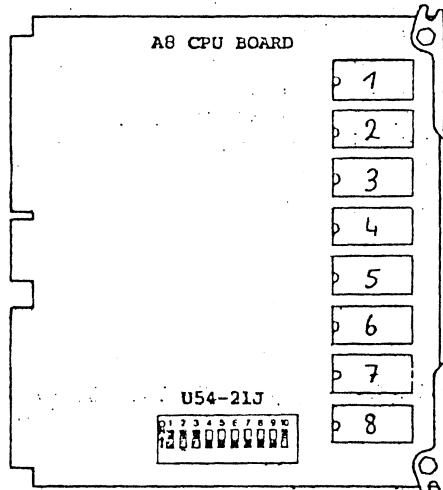


* A7 RAM- od. LCG BOARD

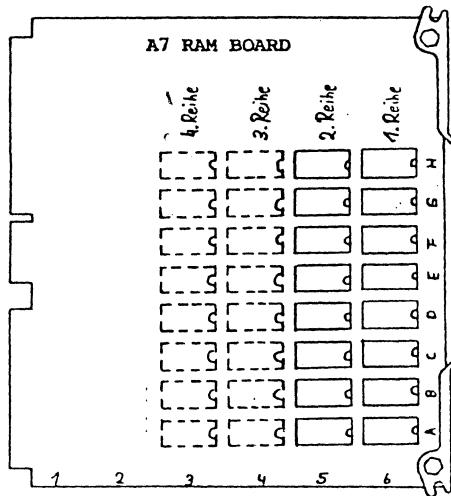
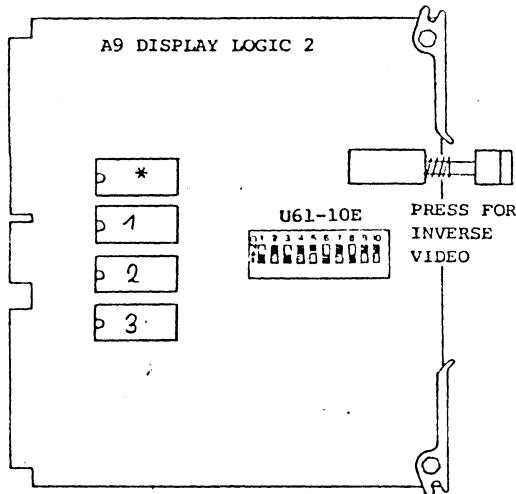
A4 - A7 RAM- od. PROM BOARDS

EINBAUPLÄTZE DER STECKBAREN BAUSTEINE

2. Einbauplätze der steckbaren Bausteine



EINBAUPLÄTZE DER STECKBAREN BAUSTEINE



1. Steckerbelegungen1.1 Async. Interface
V24 & Current Loop

PIN	
1	E1 CT 101
2	D1 CT 103
3	D2 CT 104
4	S2 CT 105
5	M2 CT 106
6	M1 CT 107
7	E2 CT 102
8	M5 CT 109
20	S1 CT 108
22	M3 CT 125
23	S4 CT 111
10	+ 12 V FL
11	
12	CLDICI
13	CLDICO
18	CLDOCI
24	CLDOCO (-12 V FL)
25	
21	+ 12V
9	- 12V

CURRENT LOOP
DISPLAY LOGIC I

1.2 Sync. Interface

PIN	
1	E1 CT 101
2	D1 CT 103
3	D2 CT 104
4	S2 CT 105
5	M2 CT 106
6	M1 CT 107
7	E2 CT 102
8	M5 CT 109
15	T2 CT 114
17	T4 CT 115
20	S1 CT 108
22	M3 CT 125
23	S4 CT 111

STECKERBELEGUNGEN

1.3 CPU CURRENT LOOP (PRINTER INTERFACE)

PIN
6 PRNTHI
7 PRNLO
8 KEYHI
9 KEYLO

1.4 CLUSTER TERMINAL CONNECTORS

Stecker 1...4

PIN
1 TRANSMIT DATA (TRD)
2 GND V24
3 RECEIVE DATA (RD) }
4 - 12V
5 + 12V
6 TRD HI } V28 Signale
7 TRD LO }
8 RD HI }
9 RD LO } CURRENT LOOP
 Signale

Stecker 5...7

PIN
6 TRD HI
7 TRD LO
8 RD HI
9 RD LO

KABELBELEGUNGEN

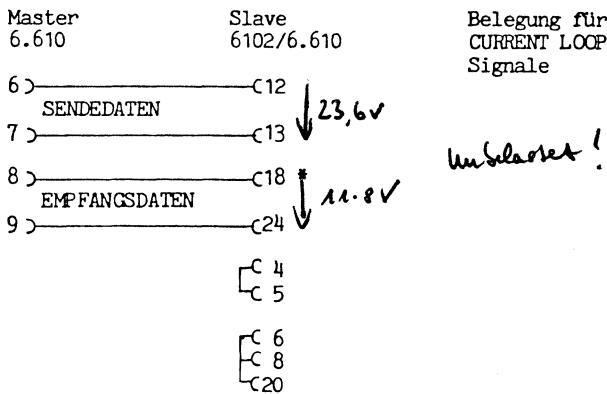
2. Kabelbelegungen

2.1 Modemkabel

PIN

1	E1	CT 101	C 1	* eingebaut ab GS 12
2	D1	CT 103	C 2	
3	D2	CT 104	C 3	
4	S2	CT 105	C 4	
5	M2	CT 106	C 5	
6	M1	CT 107	C 6	
7	E2	CT 102	C 7	
8	M5	CT 109	C 8	
15	T2	CT 114	C 15	
17	T4	CT 115	C 17	
20	S1	CT 108	C 20	
22	M3	CT 125	C 22	*
23	S4	CT 111	C 23	

2.2 Kabel für MEPLA



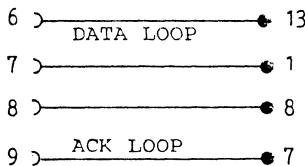
* 6102 bis OAS 1 PIN 23
6.610 bis OAS 7 PIN 23
dann Kabel in GS 10

KABELBELEGUNSEN

2.3 Druckerkabel

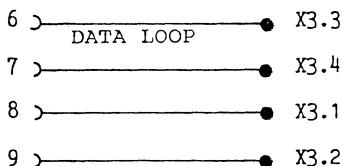
Drucker 6324

6.610 Drucker
9 pol.Cannon J1 auf FBG HCULR



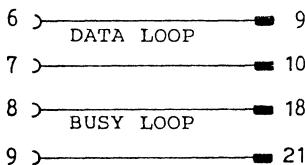
Drucker 6322

6.610 Drucker
9 pol.Cannon SST8/A101



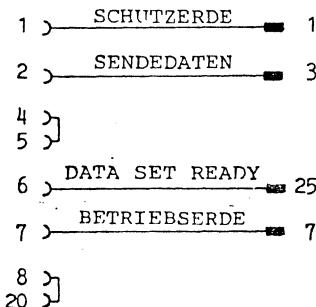
Drucker 6327

6.610 Drucker
9 pol.Cannon 25 pol.Cannon



Drucker 6325

6.610 Drucker
25 pol.Cannon 25 pol.Cannon





KAPITEL 1	CODE-TABELLEN	Seite
1. ISO 7-Bit-Code (ASCII)		7-1-1
1.1 Codetabelle		.
1.2 Steuerzeichen		7-1-2
2. EBCDIC		7-1-3
3. ISO 7-Bit - EBCDIC Konvertierung		7-1-4
4. EBCDIC - ISO 7-Bit Konvertierung		7-1-5
5. Tastaturbelegung		
5.1 Standard Tastatur		7-1-6
5.2 RIZ Tastatur		7-1-13
6. Bildschirmattribute		7-1-17
7. CCITT V24 - Schnittstellenübersicht		7-1-18

KAPITEL 2 DISKETTENFORMATE UND - VERWALTUNG

1. INTEL 256 kB	7-2-1
2. ECMA TC-15 (IBM) 256 kB	7-2-3/4

KAPITEL 3 SPEICHERAUFBAU

1. Speicherbelegung 48 kB	7-3-1
2. Speicherbelegung 54 kB	7-3-2
3. Wichtige Speicheradressen (2000H-27FFH)	7-3-3

1. ISO 7-Bit Code (ASCII)

1.1 Codetabelle (nach SN 77315)

X'ab' b=	a=	0	1	2	3	4	5	6	7
0	NUL (DLE)	TC ₁ (SOH)	SP	0	@ ¹⁾ §	P	'	p	
1	TC ₁ (SOH)	DC ₁	!	1	A	Q	a	q	
2	TC ₂ (STX)	DC ₂	"	2	B	R	b	r	
3	TC ₃ (ETX)	DC ₃	#	3	C	S	c	s	
4	TC ₄ (EOT)	DC ₄	¤ ¹⁾ §	4	D	T	d	t	
5	TC ₅ (ENQ)	TC ₈ (NAK)	%	5	E	U	e	u	
6	TC ₆ (ACK)	TC ₉ (SYN)	&	6	F	V	f	v	
7	BEL	TC ₁₀ (ETB)	,	7	G	W	g	w	
8	FEO (BS)	CAN	(8	H	X	h	x	
9	FE ₁ (HT)	EM)	9	I	Y	i	y	
A	FE ₂ (LF)	SUB	*	:	J	Z	j	z	
B	FE ₃ (VT)	ESC	+	;	K	Œ ¹⁾ Ä	k	{ ¹⁾ ä}	
C	FE ₄ (FF)	IS ₄ (FS)	,	<	L	Œ ¹⁾ Ö	l	¹⁾ ö	
D	FE ₅ (CR)	IS ₃ (GS)	-	=	M	Œ ¹⁾ Ü	m	{ ¹⁾ ü}	
E	SO	IS ₂ (RS)	.	>	N	^	n	- ¹⁾ ß	
F	SI	IS ₁ (US)	/	?	O	-	o	DEL	

1) obere Hälfte international
untere Hälfte deutsch

USASCII

1.2 Steuerzeichen

USASCII CCITT-Nr.5		
ACK0	1030	DLE0
ACK1	1031	DLE1
BEL	07	
DLE	10	
DLE EOT	1004	
ENQ	05	
EOT	04	
ESC	1B	
ETB	17	
ETX	03	
HT	09	
I	49	I
ITB	1F	US
NAK	15	
NN	nach Wahl meist 0B	
RVI	103C	DLE <
SOH	01	
STX	02	
SYN	16	
TTD	0205	STX ENQ
WABT	103F	DLE ?
WACK	103B	DLE;

2. EBCDIC (nach SN 77315)

X'ab'

a=	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
b=	NUL	TC7 (DLE)	HIA TUS		SP	&	-				{	}	\	0		
1	TC1 (SOH)	DC1				/	a	j	-		A	J			1	
2	TC2 (STX)	DC2		TC9 (SYN)			b	k	s		B	K	S	2		
3	TC3 (ETX)	DC3					c	l	t		C	L	T	3		
4							d	m	u		D	M	U	4		
5	FE1 (HT)	FE2 (LF)					e	n	v		E	N	V	5		
6		FE0	TC10 (BS) (ETB)				f	o	w		F	O	W	6		
7	DEL		ESC	TC4 (EOT)			g	p	x		G	P	X	7		
8		CAN					h	q	y		H	Q	Y	8		
9		EM					'	i	r	z	I	R	Z	9		
A			[]	I	:										
B	FE3 (VT)			.	X ¹⁾	,	#									
C	FE4 (FF)	IS4 (FS)		DC4	<	*	%	@								
D	FE5 (CR)	IS3 (GS)	TC5 (ENQ)	TC8 (NAK)	()	-	'									
E	SO	TS2 (RS)	TC6 (ACK)		+	;	>	=								
F	SI	IS1 (US)	BEL	SUB	!	^	?	"								

1) X oder §

ISO 7-BIT - EBCDIC KONVERTIERUNG

3. ISO 7-Bit-EBCDIC Konvertierung (nach SN 77315)

X'ab'

a=	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	10	40	F0	7C	D7	79	97	20	30	41	58	76	9F	B8	DC
1	01	11	4F	F1	C1	D8	81	98	21	31	42	59	77	A0	B9	DD
2	02	12	7F	F2	C2	D9	82	99	22	1A	43	62	78	A.A	BA	DE
3	03	13	7B	F3	C3	E2	83	A2	23	33	44	63	80	A.B	BB	DF
4	37	3C	5B	F4	C4	E3	84	A3	24	34	45	64	8A	AC	BC	EA
5	2D	3D	6C	F5	C5	E4	85	A4	15	35	46	65	8B	AD	BD	EB
6	2E	32	50	F6	C6	E5	86	A5	06	36	47	66	8C	AE	BE	EC
7	2F	26	7D	F7	C7	E6	87	A6	17	08	48	67	8D	AF	BF	ED
8	16	18	4D	F8	C8	E7	88	A7	28	38	49	68	8E	B0	CA	EE
9	05	19	5D	F9	C9	E8	89	A8	29	39	51	69	8F	B1	CB	EF
A	25	3F	5C	7A	D1	E9	91	A9	2A	3A	52	70	90	B2	CC	FA
B	0B	27	4E	5E	D2	4A	92	C0	2B	3B	53	71	9A	B3	CD	FB
C	0C	1C	6B	4C	D3	E0	93	6A	2C	04	54	72	9B	B4	CE	FC
D	0D	1D	60	7E	D4	5A	94	D0	09	14	55	73	9C	B5	CF	FD
E	0E	1E	4B	6E	D5	5F	95	A1	0A	3E	56	74	9D	B6	DA	FE
F	0F	1F	61	6F	D6	6D	96	07	1B	E1	57	75	9E	B7	DB	FF

erst nach Nominierung eines
ISO 8-Bit-Code sinnvoll

EBCDIC - ISO 7-BIT KONVERTIERUNG

4. EBCDIC - ISO 7-Bit Konvertierung

X'ab'

a=	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
b=	00	10			20	26	2D							7B	7D	5C	30
0	01	11					2F		61	6A	7E		41	4A		31	
1	02	12		16					62	6B	73		42	4B	53	32	
2	03	13							63	6C	74		43	4C	54	33	
3									64	6D	75		44	4D	55	34	
4	09		0A						65	6E	76		45	4E	56	35	
5		08	17						66	6F	77		46	4F	57	36	
6	7F		1B	04					67	70	78	—	47	50	58	37	
7		18							68	71	79		48	51	59	38	
8		19							60	69	72	7A		49	52	5A	39
9					5B	5D	7C	3A									
A					2E	24	2C	23									
B	0B																
C	0C	1C		14	3C	2A	25	40									
D	0D	1D	05	15	28	29	5F	27									
E	0E	1E	06		2B	3B	3E	3D									
F	0F	1F	07	1A	21	5E	3F	22									

alle nicht nach ISO umsetzbaren EBCDIC Werte werden
nach SUB (1A)₁₆ umgesetzt.

5.1 Standard Tastatur

- Belegung Standard International

Symbol	ISO-7-Bit Code	Hex. Code	Bedeutung
	000 0000	00	
XMT	000 0001	01	Datenaustausch mit Rechner
STX	000 0010	02	Steuerz. zur Übertr. eines Blockes
ETX	000 0011	03	Steuerz. am Ende eines Blockes
—	000 0100	04	Zeile löschen Cursor auf Zeilenanf.
—	000 0101	05	Ab Cursor Zeile löschen
—	000 0110	06	Ab Cursor Bildschirm löschen
← TAB	000 0111	07	Positionieren auf nächsten TAB (links)
←	000 1000	08	Cursor nach links eine Stelle
TAB →	000 1001	09	Positionieren auf nächsten TAB (rechts)
↓ (LF)	000 1010	0A	Cursor zur nächsten Zeile selbe Stelle
↑↑↑	000 1011	0B	Cursor eine Zeile tiefer
↑↑↑	000 1100	0C	Bildschirminhalt 1 Zeile nach oben
←	000 1101	0D	Cursor zur 1.St.der aktuellen Zeile
WORD-U-F1	000 1110	0E	Wort einfüg. Zeile verschieben nach rechts
↓ (NORM)	000 1111	0F	Zeile einfügen-verschieb.nach unten
CHAR(F2)	001 0000	10	Zeichen einfügen verschieben nach rechts
CHAR(F3)	001 0001	11	Zeichen einfügen verschieben nach links
WORD(F4)	001 0010	12	Wort einfügen Zeile verschieben nach links
↑	001 0011	13	Zeile löschen-verschieben nach oben
—	001 0100	14	Bildschirminhalt ausdrucken
SEND	001 0101	15	Daten zum Rechner übertragen
REQ (F5)	001 0110	16	Daten werden vom Rechner angeordert
↓↓↓	001 0111	17	Bildschirminhalt 1 Zeile nach unten
→	001 1000	18	Cursor 1 Zeichen nach rechts
—	001 1001	19	Bildschirm löschen

Symbol	ISO-7-Bit Code	Hex. Code	Bedeutung
CF (F6)	001 1010	1A	Sprung zur Anwenderroutine
ESC ↑	001 1011	1B	Umschaltung
↓	001 1100	1C	Cursor 1 Zeile höher
→	001 1101	1D	Cursor auf Bildschirmanfang
MODE (F7)	001 1110	1E	Nächstes Zeichen ist ein Modus-Auswahlzeichen
CE	001 1111	1F	Löschen letzten Wert
—	010 0000	20	Blank
!	010 0001	21	
"	010 0010	22	
+	010 0011	23	
*	010 0100	24	
%	010 0101	25	
&	010 0110	26	
'	010 0111	27	Apostroph
(010 1000	28	Sonderzeichen
)	010 1001	29	
*	010 1010	2A	
+	010 1011	2B	
,	010 1100	2C	Komma
-	010 1101	2D	Minus
.	010 1110	2E	Punkt
/	010 1111	2F	schräger Strich
0	011 0000	30	
1	011 0001	31	
2	011 0010	32	
3	011 0011	33	Ziffern
4	011 0100	34	
5	011 0101	35	
6	011 0110	36	

Symbol	ISO-7-Bit Code	Hex. Code	Bedeutung
7	011 0111	37	
8	011 1000	38	
9	011 1001	39	
:	011 1010	3A	
;	011 1011	3B	
<	011 1100	3C	
=	011 1101	3D	
>	011 1110	3E	
?	011 1111	3F	
Ⓐ	100 0000	40	
A	100 0001	41	
B	100 0010	42	
C	100 0011	43	
D	100 0100	44	
E	100 0101	45	
F	100 0110	46	
G	100 0111	47	
H	100 1000	48	
I	100 1001	49	
J	100 1010	4A	
K	100 1011	4B	
L	100 1100	4C	
M	100 1101	4D	
N	100 1110	4E	
O	100 1111	4F	
P	101 0000	50	
Q	101 0001	51	
R	101 0010	52	

Symbol	ISO-7-Bit Code	Hex. Code	Bedeutung
S	101 0011	53	
T	101 0100	54	
U	101 0101	55	
V	101 0110	56	
W	101 0111	57	
X	101 1000	58	
Y	101 1001	59	
Z	101 1010	5A	
[101 1011	5B	
\	101 1100	5C	Eckige Klammer auf schräger Strich
]	101 1101	5D	Eckige Klammer zu
^	101 1110	5E	Logisches und
-	101 1111	5F	Unterstrich
.	110 0000	60	Gravis
a	110 0001	61	
b	110 0010	62	
c	110 0011	63	
d	110 0100	64	
e	110 0101	65	
f	110 0110	66	
g	110 0111	67	
h	110 1000	68	
i	110 1001	69	
j	110 1010	6A	
k	110 1011	6B	
l	110 1100	6C	
m	110 1101	6D	
n	110 1110	6E	

Alphabetische Kleinbuchstaben

Symbol	ISO-7-Bit Code	Hex. Code	Bedeutung
o	110 1111	6F	
p	111 0000	70	
q	111 0001	71	
r	111 0010	72	
s	111 0011	73	
t	111 0100	74	
u	111 0101	75	
v	111 0110	76	
w	111 0111	77	
x	111 1000	78	
y	111 1001	79	
z	111 1010	7A	
{	111 1011	7B	geschweifte Klammer auf
	111 1100	7C	senkrechter Strich
}	111 1101	7D	geschweifte Klammer zu
—	111 1110	7E	Überstreichung
DEL	111 1111	7F	Zeichen löschen

TASTATURBELEGGUNG

STANDARD TASTATUR

- Abweichungen der Belegung Standard DEUTSCH gegenüber International

Symbol	ISO-7-Bit Code	Hex. Code	Bedeutung
DÜ	000 0001	01	
§	100 0000	40	
Ä	101 1011	5B	
Ö	101 1100	5C	
Ü	101 1101	5D	
ä	111 1011	7B	
ö	111 1100	7C	
ü	111 1101	7D	
ß	111 1110	7E	
ANF	001 0110	16	
/	010 0111	27	

- Abweichungen der Belegung Standard DÄNISCH gegenüber International

Symbol	ISO-7-Bit Code	Hex. Code	Bedeutung
Å	101 1011	5B	
Ø	101 1100	5C	
Ä	101 1101	5D	
Ü	101 1110	5E	
æ	111 1011	7B	
ø	111 1100	7C	
å	111 1101	7D	
ü	111 1110	7E	

- Abweichungen der Belegung Standard SCHWEDISCH gegenüber International

£	010 0011	23	
Ä	101 1011	5B	Umlaut
Ö	101 1100	5C	Umlaut
Å	101 1101	5D	Angström
ä	111 1011	7B	Umlaut
ö	111 1100	7C	Umlaut
å	111 1101	7D	Angström
ü	111 1110	7E	Umlaut

- Abweichungen der Belegung Standard BELGIESCH/
FRANZÖSISCH AZERTY und QWERTY gegenüber International

Symbol	ISO-7-Bit Code	Hex. Code	Bedeutung
à	100 0000	40	Sonderzeichen
o	101 1011	5B	Grad
ç	101 1100	5C	Cedille
BS	101 1101	5D	Back Space
^	101 1110	5E	Circonflex
-	101 1111	5F	Unterstrich
,	110 0000	60	Gravis
é	111 1011	7B	e accent aigue
ù	111 1100	7C	u accent grave
è	111 1101	7D	e accent grave
-	111 1110	7E	Tilde

- Abweichungen der Belegung Standard NORWEGISCH
gegenüber International

Symbol	ISO-7-Bit Code	Hex. Code	Bedeutung
£	010 0011	23	Sonderzeichen
Æ	101 1011	5B	
Ø	101 1100	5C	
Å	101 1101	5D	
^	101 1110	5E	
-	101 1111	5F	
œ	111 1011	7B	
ø	111 1100	7C	
å	111 1101	7D	
-	111 1110	7E	

5.2 RIZ Tastatur

Symbol	Hex-Code	Bedeutung
	00	
Schalter B	01	Ende Transaktion Bediener B
STX	02	Steuerz.z.Übertr. eines Blockes
ETX	03	Steuerz.am Ende eines Blockes
	04	Zeile löschen - Cursor auf Anfang
	05	Ab Cursor, Zeile löschen
	06	Ab Cursor, Bildschirm löschen
	07	Positionieren auf nächst.TAB (links)
	08	Cursor n. links, eine Stelle
	09	Positionieren auf nächst.TAB (rechts)
	0A	Cursor z.nächst. Zeile, selbe Stelle
	0B	Cursor eine Zeile tiefer
	0C	Bildschirminhalt eine Zeile nach oben
	0D	Cursor zur 1.Stelle d.aktuellen Zeile
UNT.STR. # *	0E	Attribute-Betriebsart ein/# Funktion
NORMAL X *	0F	Attribute-Betriebsart aus X Funktion
STO	10	Storno
KT	11	Konto-Nummer
UNI	12	Auslösetaste für UNI-Code
OFF	13	Vorwahltaste für OFF-LINE-Buchungen
	14	Bildschirminhalt ausdrucken
Schalter K	15	Funkt.wird über Schlüsselschal.freigeg.
MSL	16	Auslösen der Funk.Magnetstreifenleser
	17	Bildschirminhalt 1 Zeile nach unten
	18	Cursor ein Zeichen nach rechts
	19	Gesamten Bildschirm löschen
Schalter A	1A	Ende Transaktion Bediener A
ESC	1B	Umschaltung
	1C	Cursor eine Zeile höher
	1D	Cursor auf Bildschirmanfang

Symbol	Hex. Code	Bedeutung
F	1E	Vorwahltaste für Bedienerführung
CE	1F	Löschen letzter Wert
_	20	Blank
!	21	
"	22	
#	23	
\$	24	
%	25	
&	26	
'	27	(Apostroph)
(28	Sonderzeichen
)	29	
*	2A	
+	2B	
,	2C	(Komma)
-	2D	
.	2E	
/	2F	(schräger Strich)
0	30	
1	31	
2	32	
3	33	
4	34	Ziffern
5	35	
6	36	
7	37	
8	38	
9	39	
:	3A	
;	3B	Sonderzeichen
<	3C	
=	3D	

Symbol	Hex-Code	Bedeutung
>	3E	
?	3F	
§	40	
A	41	
B	42	
C	43	
D	44	
E	45	
F	46	
G	47	
H	48	
I	49	
J	4A	
K	4B	
L	4C	
M	4D	
N	4E	Alphabetische Großbuchstaben
O	4F	
P	50	
Q	51	
R	52	
S	53	
T	54	
U	55	
V	56	
W	57	
X	58	
Y	59	
Z	5A	
Ä	5B	Umlaut (Großbuchstabe)
Ö	5C	Umlaut (Großbuchstabe)
Ü	5D	Umlaut (Großbuchstabe)
^	5E	Logisches und

Symbol	Hex-Code	Bedeutung
—	5F	
,	60	Unterstrich Gravis
a F1*	61	
b F2*	62	
c F3*	63	
d F4*	64	
e F5*	65	
f F6	66	
g F7*	67	
h F8*	68	
i SUM*	69	
j TK*	6A	
k NB*	6B	
l BEL*	6C	
m	6D	Alphabetische Kleinbuchstaben
n	6E	
o	6F	
p	70	
q	71	
r	72	
s	73	
t	74	
u	75	
v	76	
w	77	
x	78	
y	79	
z	7A	
ä	7B	
ö	7C	Alphabetische Kleinbuchstaben
ü	7D	
ß	7E	
BSA	7F	Buchsaldo

6. Bildschirmattribute

Bildschirm im ATTRIBUT-Modus

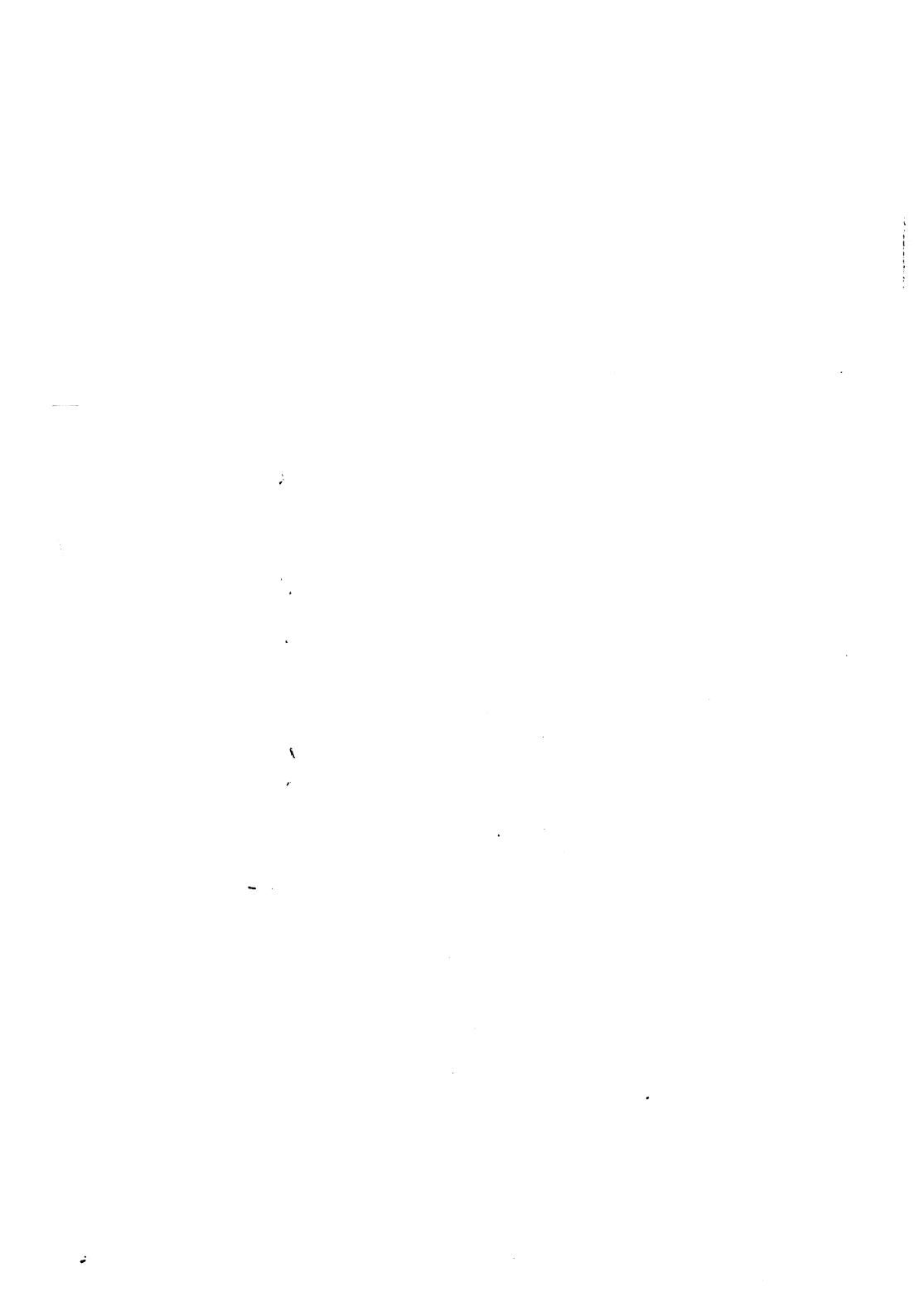
Betriebszustand	Tastenbetätigung	Tasten-Code	BIN-CODE*	HEX-CODE
Invertiert	UNDERLINE SHIFT + A NORMAL	0E 41 0F	bit 654 100	CØ
Blinken	UNDERLINE irgendeine Ziffer z.B.0 NORMAL	0E 30 0F	011	BØ
Geringere Intensität	UNDERLINE SPACE NORMAL	0E 20 0F	010	AØ
Unter- streichung	UNDERLINE SHIFT + Q NORMAL	0E 51 0F	101	DØ
Unsichtbar	UNDERLINE a NORMAL	0E 61 0F	110	EØ
Normal	UNDERLINE q NORMAL	0E 71	111	FØ

* Angabe im Assembler: 1xxxØØØØ

CCITT V24 - SCHNITTSTELLENÜBERSICHT

7. CCITT V24 - Schnittstellenübersicht

Schnittstellenbezeichnung bei einer nach Nummer 150 oder 600/1000 MHz betriebenen Anlage	Daten- rate	Schnittstellenbeschreibung		Taste steuert und steuert Analogzeichen oder Digital- Zeichen	Frequenz- signale	Frequenz- signale		Taste steuert und steuert Analogzeichen oder Digital- Zeichen	
		Standard	speziell			von	nach		
1. 1. 101	1	1. 1. 102	1	1. 1. 103	1	1. 1. 104	1	1. 1. 105	1
1. 1. 106	2	1. 1. 107	2	1. 1. 108	2	1. 1. 109	2	1. 1. 110	2
1. 1. 111	5	1. 1. 112	5	1. 1. 113	5	1. 1. 114	5	1. 1. 115	5
1. 1. 116	6	1. 1. 117	6	1. 1. 118	6	1. 1. 119	6	1. 1. 120	6
1. 1. 121	7	1. 1. 122	7	1. 1. 123	7	1. 1. 124	7	1. 1. 125	7
1. 1. 126	8	1. 1. 127	8	1. 1. 128	8	1. 1. 129	8	1. 1. 130	8
1. 1. 131	9	1. 1. 132	9	1. 1. 133	9	1. 1. 134	9	1. 1. 135	9
1. 1. 136	10	1. 1. 137	10	1. 1. 138	10	1. 1. 139	10	1. 1. 140	10
1. 1. 141	11	1. 1. 142	11	1. 1. 143	11	1. 1. 144	11	1. 1. 145	11
1. 1. 146	12	1. 1. 147	12	1. 1. 148	12	1. 1. 149	12	1. 1. 150	12
1. 1. 151	13	1. 1. 152	13	1. 1. 153	13	1. 1. 154	13	1. 1. 155	13
1. 1. 156	14	1. 1. 157	14	1. 1. 158	14	1. 1. 159	14	1. 1. 160	14
1. 1. 163	15	1. 1. 164	15	1. 1. 165	15	1. 1. 166	15	1. 1. 167	15
1. 1. 168	16	1. 1. 169	16	1. 1. 170	16	1. 1. 171	16	1. 1. 172	16
1. 1. 173	17	1. 1. 174	17	1. 1. 175	17	1. 1. 176	17	1. 1. 177	17
1. 1. 178	18	1. 1. 179	18	1. 1. 180	18	1. 1. 181	18	1. 1. 182	18
1. 1. 185	19	1. 1. 186	19	1. 1. 187	19	1. 1. 188	19	1. 1. 189	19
1. 1. 190	20	1. 1. 191	20	1. 1. 192	20	1. 1. 193	20	1. 1. 194	20
1. 1. 195	21	1. 1. 196	21	1. 1. 197	21	1. 1. 198	21	1. 1. 199	21
1. 1. 200	22	1. 1. 201	22	1. 1. 202	22	1. 1. 203	22	1. 1. 204	22
1. 1. 205	23	1. 1. 206	23	1. 1. 207	23	1. 1. 208	23	1. 1. 209	23
1. 1. 210	24	1. 1. 211	24	1. 1. 212	24	1. 1. 213	24	1. 1. 214	24
1. 1. 215	25	1. 1. 216	25	1. 1. 217	25	1. 1. 218	25	1. 1. 219	25
1. 1. 220	26	1. 1. 221	26	1. 1. 222	26	1. 1. 223	26	1. 1. 224	26
1. 1. 225	27	1. 1. 226	27	1. 1. 227	27	1. 1. 228	27	1. 1. 229	27
1. 1. 230	28	1. 1. 231	28	1. 1. 232	28	1. 1. 233	28	1. 1. 234	28
1. 1. 235	29	1. 1. 236	29	1. 1. 237	29	1. 1. 238	29	1. 1. 239	29
1. 1. 240	30	1. 1. 241	30	1. 1. 242	30	1. 1. 243	30	1. 1. 244	30
1. 1. 245	31	1. 1. 246	31	1. 1. 247	31	1. 1. 248	31	1. 1. 249	31
1. 1. 250	32	1. 1. 251	32	1. 1. 252	32	1. 1. 253	32	1. 1. 254	32
1. 1. 255	33	1. 1. 256	33	1. 1. 257	33	1. 1. 258	33	1. 1. 259	33
1. 1. 260	34	1. 1. 261	34	1. 1. 262	34	1. 1. 263	34	1. 1. 264	34
1. 1. 265	35	1. 1. 266	35	1. 1. 267	35	1. 1. 268	35	1. 1. 269	35
1. 1. 270	36	1. 1. 271	36	1. 1. 272	36	1. 1. 273	36	1. 1. 274	36
1. 1. 275	37	1. 1. 276	37	1. 1. 277	37	1. 1. 278	37	1. 1. 279	37
1. 1. 280	38	1. 1. 281	38	1. 1. 282	38	1. 1. 283	38	1. 1. 284	38
1. 1. 285	39	1. 1. 286	39	1. 1. 287	39	1. 1. 288	39	1. 1. 289	39
1. 1. 290	40	1. 1. 291	40	1. 1. 292	40	1. 1. 293	40	1. 1. 294	40
1. 1. 295	41	1. 1. 296	41	1. 1. 297	41	1. 1. 298	41	1. 1. 299	41
1. 1. 300	42	1. 1. 301	42	1. 1. 302	42	1. 1. 303	42	1. 1. 304	42
1. 1. 305	43	1. 1. 306	43	1. 1. 307	43	1. 1. 308	43	1. 1. 309	43
1. 1. 310	44	1. 1. 311	44	1. 1. 312	44	1. 1. 313	44	1. 1. 314	44
1. 1. 315	45	1. 1. 316	45	1. 1. 317	45	1. 1. 318	45	1. 1. 319	45
1. 1. 320	46	1. 1. 321	46	1. 1. 322	46	1. 1. 323	46	1. 1. 324	46
1. 1. 325	47	1. 1. 326	47	1. 1. 327	47	1. 1. 328	47	1. 1. 329	47
1. 1. 330	48	1. 1. 331	48	1. 1. 332	48	1. 1. 333	48	1. 1. 334	48
1. 1. 335	49	1. 1. 336	49	1. 1. 337	49	1. 1. 338	49	1. 1. 339	49
1. 1. 340	50	1. 1. 341	50	1. 1. 342	50	1. 1. 343	50	1. 1. 344	50
1. 1. 345	51	1. 1. 346	51	1. 1. 347	51	1. 1. 348	51	1. 1. 349	51
1. 1. 350	52	1. 1. 351	52	1. 1. 352	52	1. 1. 353	52	1. 1. 354	52
1. 1. 355	53	1. 1. 356	53	1. 1. 357	53	1. 1. 358	53	1. 1. 359	53
1. 1. 360	54	1. 1. 361	54	1. 1. 362	54	1. 1. 363	54	1. 1. 364	54
1. 1. 365	55	1. 1. 366	55	1. 1. 367	55	1. 1. 368	55	1. 1. 369	55
1. 1. 370	56	1. 1. 371	56	1. 1. 372	56	1. 1. 373	56	1. 1. 374	56
1. 1. 375	57	1. 1. 376	57	1. 1. 377	57	1. 1. 378	57	1. 1. 379	57
1. 1. 380	58	1. 1. 381	58	1. 1. 382	58	1. 1. 383	58	1. 1. 384	58
1. 1. 385	59	1. 1. 386	59	1. 1. 387	59	1. 1. 388	59	1. 1. 389	59
1. 1. 390	60	1. 1. 391	60	1. 1. 392	60	1. 1. 393	60	1. 1. 394	60
1. 1. 395	61	1. 1. 396	61	1. 1. 397	61	1. 1. 398	61	1. 1. 399	61
1. 1. 400	62	1. 1. 401	62	1. 1. 402	62	1. 1. 403	62	1. 1. 404	62
1. 1. 405	63	1. 1. 406	63	1. 1. 407	63	1. 1. 408	63	1. 1. 409	63
1. 1. 410	64	1. 1. 411	64	1. 1. 412	64	1. 1. 413	64	1. 1. 414	64
1. 1. 415	65	1. 1. 416	65	1. 1. 417	65	1. 1. 418	65	1. 1. 419	65
1. 1. 420	66	1. 1. 421	66	1. 1. 422	66	1. 1. 423	66	1. 1. 424	66
1. 1. 425	67	1. 1. 426	67	1. 1. 427	67	1. 1. 428	67	1. 1. 429	67
1. 1. 430	68	1. 1. 431	68	1. 1. 432	68	1. 1. 433	68	1. 1. 434	68
1. 1. 435	69	1. 1. 436	69	1. 1. 437	69	1. 1. 438	69	1. 1. 439	69
1. 1. 440	70	1. 1. 441	70	1. 1. 442	70	1. 1. 443	70	1. 1. 444	70
1. 1. 445	71	1. 1. 446	71	1. 1. 447	71	1. 1. 448	71	1. 1. 449	71
1. 1. 450	72	1. 1. 451	72	1. 1. 452	72	1. 1. 453	72	1. 1. 454	72
1. 1. 455	73	1. 1. 456	73	1. 1. 457	73	1. 1. 458	73	1. 1. 459	73
1. 1. 460	74	1. 1. 461	74	1. 1. 462	74	1. 1. 463	74	1. 1. 464	74
1. 1. 465	75	1. 1. 466	75	1. 1. 467	75	1. 1. 468	75	1. 1. 469	75
1. 1. 470	76	1. 1. 471	76	1. 1. 472	76	1. 1. 473	76	1. 1. 474	76
1. 1. 475	77	1. 1. 476	77	1. 1. 477	77	1. 1. 478	77	1. 1. 479	77
1. 1. 480	78	1. 1. 481	78	1. 1. 482	78	1. 1. 483	78	1. 1. 484	78
1. 1. 485	79	1. 1. 486	79	1. 1. 487	79	1. 1. 488	79	1. 1. 489	79
1. 1. 490	80	1. 1. 491	80	1. 1. 492	80	1. 1. 493	80	1. 1. 494	80
1. 1. 495	81	1. 1. 496	81	1. 1. 497	81	1. 1. 498	81	1. 1. 499	81
1. 1. 500	82	1. 1. 501	82	1. 1. 502	82	1. 1. 503	82	1. 1. 504	82
1. 1. 505	83	1. 1. 506	83	1. 1. 507	83	1. 1. 508	83	1. 1. 509	83
1. 1. 510	84	1. 1. 511	84	1. 1. 512	84	1. 1. 513	84	1. 1. 514	84
1. 1. 515	85	1. 1. 516	85	1. 1. 517	85	1. 1. 518	85	1. 1. 519	85
1. 1. 520	86	1. 1. 521	86	1. 1. 522	86	1. 1. 523	86	1. 1. 524	86
1. 1. 525	87	1. 1. 526	87	1. 1. 527	87	1. 1. 528	87	1. 1. 529	87
1. 1. 530	88	1. 1. 531	88	1. 1. 532	88	1. 1. 533	88	1. 1. 534	88
1. 1. 535	89	1. 1. 536	89	1. 1. 537	89	1. 1. 538	89	1. 1. 539	89
1. 1. 540	90	1. 1. 541	90	1. 1. 542	90	1. 1. 543	90	1. 1. 544	90
1. 1. 545	91	1. 1. 546	91	1. 1. 547	91	1. 1. 548	91	1. 1. 549	91
1. 1. 550	92	1. 1. 551	92	1. 1. 552	92	1. 1. 553	92	1. 1. 554	92
1. 1. 555	93	1. 1. 556	93	1. 1. 557	93	1. 1. 558	93	1. 1. 559	93
1. 1. 560	94	1. 1. 561	94	1. 1. 562	94	1. 1. 563	94	1. 1. 564	94
1. 1. 565	95	1. 1. 566	95	1. 1. 567	95	1. 1. 568	95	1. 1. 569	95
1. 1. 570	96	1. 1. 571	96	1. 1. 572	96	1. 1. 573	96	1. 1. 574	96
1. 1. 575	97	1. 1. 576	97	1. 1. 577	97	1. 1. 578	97	1. 1. 579	97
1. 1. 580	98	1. 1. 581	98	1. 1. 582	98	1. 1. 583	98	1. 1. 584	98
1. 1. 585	99	1. 1. 586	99	1. 1. 587	99	1. 1. 588	99	1. 1. 589	99
1. 1. 590	100	1. 1. 591	100	1. 1. 592	100	1. 1. 593	100	1. 1. 594	100



1. INTEL 256 kB

Diskettenaufbau (Spur/Sektor):

00/01 bis 00/24 ISIS.ERR
00/25 bis 00/26 ISIS.LAB
01/01 bis 01/26 ISIS.DIR
02/01 bis 02/03 ISIS.MAP

02/04 bis 76/25 Datenbereich

ISIS.ERR

00/01 Datei-Kettungsblock
00/02 bis 00/24 BS10-Fehlermeldungen

ISIS.LAB

00/25 Datei-Kettungsblock
00/26 Disketten-Kennsatz

ISIS.DIR

01/01 Datei-Kettungsblock
01/02 bis 01/26 Disketten-Inhaltsverzeichnis

Jeder Sektor des Disketten-Inhaltsverzeichnis besteht aus 8 Einträgen zu je 16 Bytes Länge mit folgenden möglichen Formaten:

0	1	7	10	11	12	14	16
PRE	NAME	EXT	FLAG	NB	NBLK	DSDIR	

PRE : 00 - Datei vorhanden
FF - Datei gelöscht
7F - Freiplatz

NAME : Dateiname (6 Bytes)

EXT : Namenserweiterung (3 Bytes)

FLAG : 80H - F Bsp: Datei mit Attribut X,W,I
40H - X FLAG = 45H (40H + 04H + 01H)
08H - A
04H - W
02H - S
01H - I

NB : Datenbytes/letzter Block

NBLK : Anzahl Datenblöcke

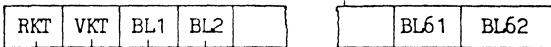
DSDIR : Sektor/Spur des 1. Dateikettungsblockes

ISIS.MAP

02/01 Datei-Kettungsblock
02/02 bis 02/03 Disketten-Belegungsverzeichnis

Jedes Bit entspricht einem Sektor der Diskette
Bit = Ø Sektor nicht belegt
Bit = 1 Sektor belegt

Dateikettungsblock-Aufbau



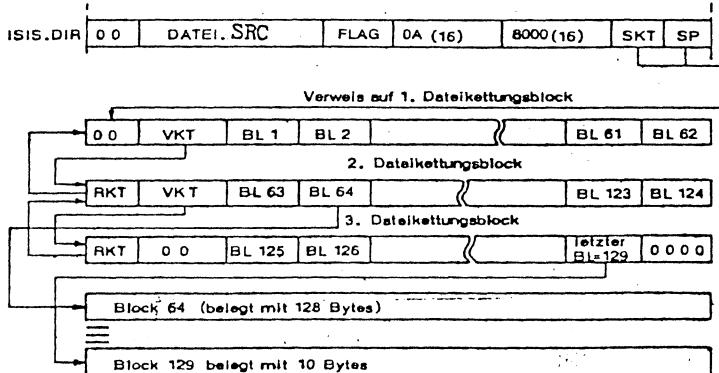
RKT - Sektor/Spur des vorhergehenden Kettungsblockes
falls vorhanden, sonst ØØ

VKT - Sektor/Spur des nächsten Kettungsblockes
falls vorhanden, sonst ØØ

BL1 - Sektor/Spur der Datenblöcke die zu diesem
Kettungsblock gehören

BL62

Beispiel: Datei "DATEI.SRC" bestehend aus 129 Datenblöcken



2. ECM TC - 15 (IBM) 256 KBDiskettenaufbau (Spur/Sektor)

00/01 bis 00/04	Reserviert
00/05	ERMAP
00/06	Reserviert
00/07	DATENTRÄGER-KENNSATZ
00/08 bis 00/26	DATEIKENNSATZ
01/01 bis 74/26	Datenbereich
75/01 bis 75/26	Ersatzspur 1
76/01 bis 76/26	Ersatzspur 2

ERMAP (00/05)

Stelle: 1 - 5 : ERMAP
 7 - 8 : Leer oder Nummer der ersten defekten Spur (dezimal)
 11 - 12 : Leer oder Nummer der zweiten defekten Spur (dezimal)

DATENTRÄGER-KENNSATZ (00/07)

Stelle: 1 - 4 : VOL1
 5 - 10 : Datenträger-Kennsatzinformation
 11 : Zugriffsschutzfeld)*
 38 - 51 : Eigentümerinformation)*
 76 : Phys. Sektorlänge der Spuren
 1-76)*
 leer - 128 Bytes
 1 - 256 Bytes
 2 - 512 Bytes
 77 - 78 : Phys. Sektorfolge)*

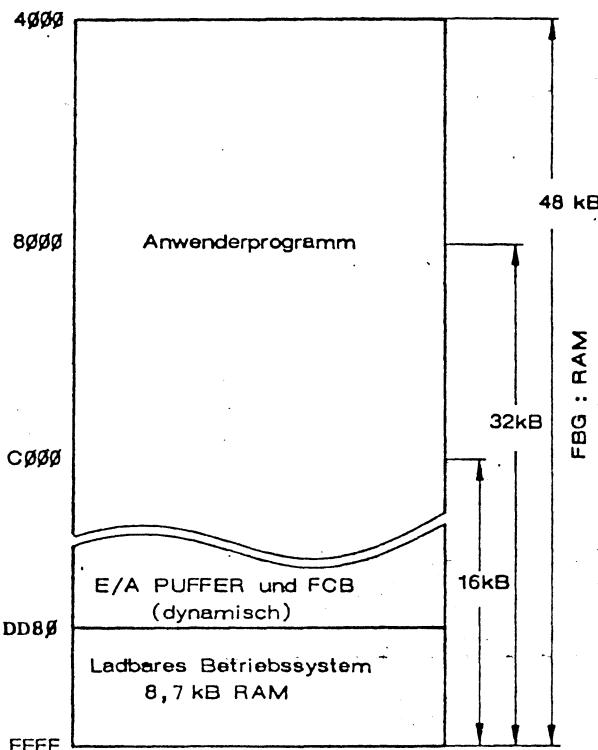
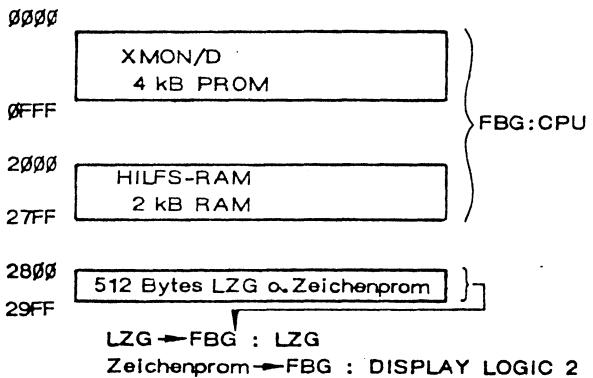
DATEIKENNSATZ (00/08 bis 00/26)

Stelle: 1 - 4 : HDR1
 6 - 13 : Dateiname
 29 - 33 : BOE
 29 - 30 Spurnummer
 32 - 33 Sektornummer
 34 : Phys. Satzlänge)*
 Vergl. Stelle 76 des Datenträger-Kennsatzes
 35 - 39 : EOE
 35 - 36 Spurnummer
 38 - 39 Sektornummer
 42 : Zugriffsschutzfeld)*
 43 : Schreibschutz)*
 Leerfeld - ungeschützt
 P - geschützt
 48 - 53 : Schreibdatum
 JJMMDD
 54 - 57 : Satzlänge
 67 - 72 : Freigabedatum)*
 JJMMDD
 75 - 79 : EOD
 75 - 76 Spurnummer
 78 - 79 Sektornummer

)* vorläufig nicht benutzt

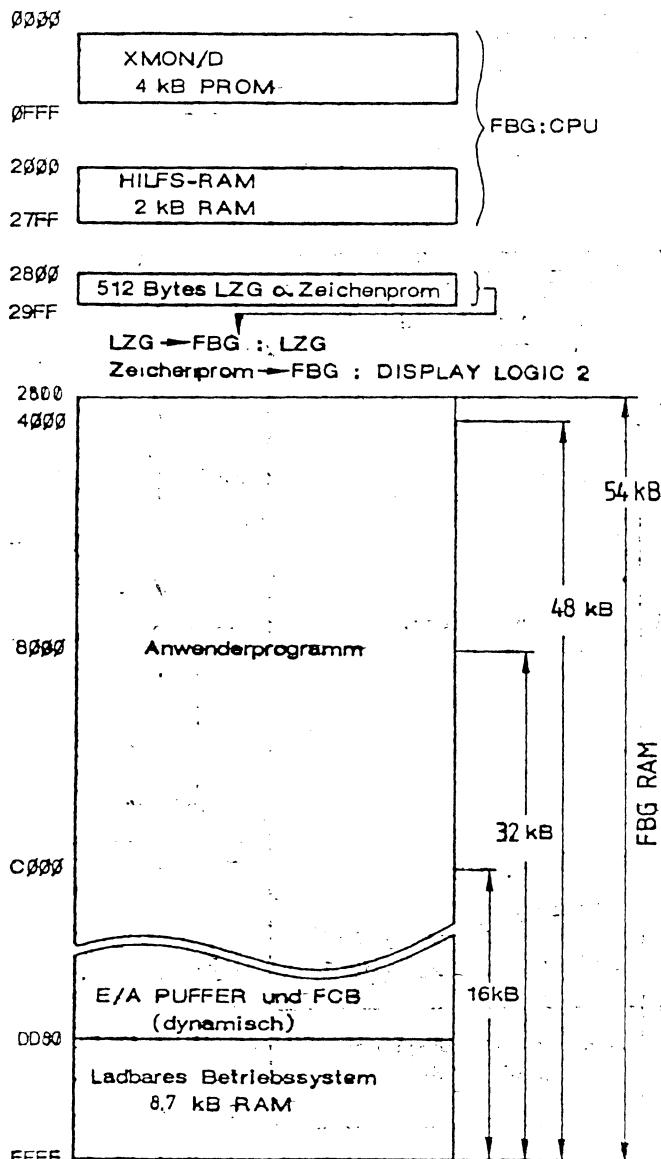
SPEICHERBELEGUNG 48 kB

1. Speicherbelegung 48 kB



SPEICHERBELEGUNG 54 kB

2. Speicherbelegung 54 kB



WICHTIGE SPEICHERADRESSEN

3. Wichtige Speicheradressen (2000H-27FFH)

SYSTEM - KELLERSPEICHER		
2000H - 25BEH	STACK	Kellerspeicher-Bereich
UART - SENDEPUFFER		
25C0H	XMIQNT	Zeichenzähler
25C1H	XMIPNT	Pufferzeiger
25C2H - 25DFH	XMIBUF	Sendepuffer
UART - EMPFANGSPUFFER		
25E0H	RCVCNT	Zeichenzähler
25E1H	RCVPNT	Pufferzeiger
25E2H - 25FFH	RCVBUF	Empfangspuffer
SYSTEMPUFFER II		
2600H - 267FH	BUF2	Systempuffer für Diskettenoperationen
SYSTEMPUFFER I		
2680H - 26FFFH	BUF1	Systempuffer für Diskettenoperationen
SYSTEM - ARBEITSBEREICH I		
2730H	UNIT	FD-Gerätenummer (von wo SYSTEM geladen wurde)
2739H - 273AH	MAP1	Startadresse des 1.Sektors der ISIS.MAP
273BH - 273CH	MAP2	Startadresse des 2.Sektors der ISIS.MAP
DISKETTEN - PUFFER		
277AH - 277BH	XMINT	Adresse eines eigenen UART-Treivers
277CH - 277DH	EBCDIC	Adresse der Konvertierungstabelle ASCII → EBCDIC
277EH - 277FH	ASCII	Adresse der Konvertierungstabelle EBCDIC → ASCII
TASTATUR - PUFFER		
2780H	KEYCNT	Zeichenzähler
2781H	KEYPNT	Pufferzeiger
2782H - 279FH	KEYBUF	Tastaturnpuffer
ANWENDERSTACK - ZEIGER		
279EH - 279FH	USTACP	Anwenderstackzeiger

WICHTIGE SPEICHERADRESSEN (2000H-27FFH)

UNTERBRECHUNGS - KONTROLLEBLOCK

27AOH - 27A1H	RST0	Power up
27A2H - 27A3H	RST1	Zeitgeber
27A4H - 27A5H	RST2	Synchron Interface *
27A6H - 27A7H	RST3	Master-Display *1)
27A8H - 27A9H	RST4	Cluster-Interface *
27AAH - 27ABH	RST5	frei *
27ACH - 27ADH	RST6	Disketten-Controller
27AEH - 27AFH	RST7	MB-Controller

* können vom Anwender benutzt werden

1) nur Tastatur

ZEITBLOCK

27B0H - 27B1H	TIMADR	Unterbrechungsadresse (TIMER)
27B2H	EXSW	EXEC-Schalter
27B3H - 27B4H	PTRRADR	Druckertreiberadresse
27B5H	YEAR	Jahr (Hex) nicht vom
27B6H	MNT	Monat (Hex) Monitor
27B7H	DOFMCTR	Tag (Hex) behandelt
27B8H	DAYCNT	Tageszähler (01 aufwärts)
27B9H	HRONT	Stundenzähler (24 abwärts)
27BAH	MINCTR	Minutenzähler (60 abwärts)
27BBH	SECCTR	Sekundenzähler(60 abwärts)
27BCH	MSCTR	20ms-Zähler (50 abwärts)
27BDH - 27BEH	BINCLK	Zeitintervalzähler (TIMER)
27BFH	DTIMER	Diskettentimerzelle (Timer abgelaufen Fehler05)

Notation

Wörter in Großbuchstaben	= Schlüsselwörter (müssen angegeben werden)
Wörter in Kleinbuchstaben	= stehen stellvertretend für Angaben die der Benutzer eintragen muß
[:::]	= Eine der angebotenen Optionen <u>kann</u> ausgewählt werden
{:::}	= Eine der angebotenen Optionen <u>muß</u> ausgewählt werden
...	= Ausdruck kann beliebig oft wiederholt werden

Verwendete Abkürzungen:

SP oder Zwischenraum	= Leertaste
CR	= Carriage Return Taste
LF	= Line Feed Taste
lw	= Laufwerk
adr	= Adresse
reg	= Registerbezeichnung
param	= Parameter
l	= Länge