

C17B



**Prozeßleitsystem audatec
Technische Dokumentation
Applikationsrechner**

Anwendungsvorschrift B



Inhaltsverzeichnis

0.	Allgemeines	6
1.	Systembeschreibung	10
1.1.	Komponenten des Betriebssystems	10
1.2.	Anlauforganisation	10
1.3.	Programmabarbeitung	13
1.4.	Stackverwaltung	14
1.5.	Registerrettung	15
1.6.	Verarbeitungsprogramme	15
1.7.	Programmablaufsteuerung	17
1.8.	Ebenenverwaltung	17
1.9.	Zugriff auf periphere Gerate (Drucker, PD)	18
1.9.1.	Kommando zur E/A- Steuerung - EAST	19
1.9.2.	Kommando Warten auf E/A - WEA	25
1.9.3.	Vorrangbelegung	26
1.10.	Datenorganisation/Datentypen	27
1.11.	DUE-Organisation	30
1.11.1.	Prozessabbild	30
1.11.2.	Direkter Zugriff ueber Kommando	32
1.11.2.1.	Grundregime der Kopplung	33
1.11.2.2.	Datenbereitstellungsregime	35
1.11.2.3.	Schreibbedingungen bei KOM-Zugriff	40
1.11.3.	Beispiele zur DUE	41
1.11.4.	Aussagen zur DUE-Leistungsfahigkeit	46
1.12.	RST-Routinen	48
1.13.	Kommandos, Hilfsprogramme, Unterprogramme	48
1.13.1.	Unterprogramme fuer BS/Tastatur	48
1.13.1.1.	BS-Unterprogramme	49
1.13.1.2.	Tastatur UP	55
1.13.2.	Arithmetik- und Konvertierungsprogramme	61
1.13.2.1.	GK-Format	61
1.13.2.2.	Konvertierungsprogramm fuer weitere Formate	61
1.13.2.3.	Konvertierung Integer 2, 4 Byte <=> GK	62
1.13.3.	Uhrzeit/Datum	63
1.13.4.	Kommandos/Systemrufe	63
1.13.5.	Unrechnung zwischen GK und PG	68
1.13.6.	Unterprogramme zur Drucker- und Floppy-Disk-Ansteuerung	69
1.13.6.1.	Ausgabe zum Seriendrucker	69
1.13.6.2.	Floppy-Disk-Programme	70
1.14.	Einbindung von Interrupt-Service-Routinen durch den Anwender	77
1.15.	Zusammenstellung von Systemsteuerzellen/Adressen	79
1.16.	Kenngroessen des Betriebssystems	82
2.	Ueberwachung und Fehlermeldung	83
2.1.	Allgemeines Anzeigekonzept	83
2.2.	Fehleranzeige bei Rechnerhalt	83
2.3.	Beschreibung der Anlaufmerker MVA/PUA/MUSD1	86
2.4.	Systemfehlermeldungen	92
2.4.1.	Ueberblick der Systemfehlermeldungen	92
2.4.2.	Systemfehlermeldungen bei laufendem System - Darstellung auf Bildschirm	92
2.4.3.	Erlaeuterungen der Systemfehlermeldungen	95
2.4.4.	Nutzung des internen Fehlersystems durch den Anwender	106

3.	Standardfunktionen des Systems	113
3.1.	Funktionen - Ueberblick	113
3.2.	Grundmenue	113
3.3.	Systemstrukturierung	114
3.3.1.	Funktion 0 bis 2	115
3.3.2.	Systemdaten	115
3.3.3.	Anmeldeleiste	116
3.3.4.	Steuerleiste VAP	116
3.3.5.	Strukturierung des Prozessabbildes (PRAB)	118
3.3.5.1.	Neustart	118
3.3.5.2.	Verwaltung	118
3.3.5.3.	Steuerfelder	120
3.3.6.	Strukturierung des AR-Abbildes	121
3.3.7.	Strukturierung weiterer Steuerlisten	121
3.4.	Systembedienung	122
3.4.1.	Zugriff auf PRAB und AR-Abbild	122
3.4.1.1.	Anzeige PRAB/AR-Abbild	122
3.4.1.2.	Zugriff auf Prozessabbild und AR-Abbild	124
3.4.2.	Dateneingabe in BSE	124
3.4.3.	Anzeige der Eingabe in Hexa	127
3.4.3.1.	Aufbau der Steuerlisten zur Anzeige	127
3.4.3.2.	Anzeige der Daten	132
3.4.3.3.	Eingabe von Daten	132
3.4.3.4.	Kommando	132
3.4.4.	Fehlerzustand, Fehlerpuffer und Status des AR	132
3.4.5.	Bedienung Seriendrucker	133
3.4.5.1.	Systemfunktion	133
3.4.5.2.	Hardcopy	133
3.4.6.	Bedienung der Floppy-Disk-Einheit	135
3.4.6.1.	Hilfsmenue	H 136
3.4.6.2.	Bibliothek auflisten	B 137
3.4.6.3.	Diskette initialisieren	I 137
3.4.6.4.	Systemdiskette duplizieren	Y 138
3.4.6.5.	Alle Dateien duplizieren	D 138
3.4.6.6.	Fehler-/Statusabfrage	X 138
3.4.6.7.	Laufwerk Freigabe	F 138
3.4.6.8.	Datei kopieren	K 138
3.4.6.9.	Datei streichen	C 138
3.4.6.10.	Datei umbenennen	R 139
3.4.6.11.	Datei lesen	L 139
3.4.6.12.	Datei schreiben	S 139
3.4.6.13.	Datei weiter schreiben	W 140
3.4.6.14.	Ausgewählte Fehler-/Statusmeldungen	140
3.4.7.	EPRON-Programmierung und Pruefsummenbildung	141
3.4.8.	Uhrzeit/Datum	143
3.5.	Monitor	144
4.	Erstellung und Einbindung von Anwenderprogrammen	150
4.1.	Programmerstellung	150
4.2.	Einordnung in die zeitliche Abarbeitung	150
4.3.	Systemstrukturierung	150
4.4.	Programmtest	150
4.5.	Programmeinbindung	150
4.6.	Nutzung des Grundbereiches auf EPRON durch den Anwender	151
4.7.	Zugriff zum PRAB und AR-Abbild	151
4.8.	UP mit Ebenenwechsel (parallel zur Programmebene)	151

4.9.	Einbindung von ISR durch den Anwender	151
4.10.	Aufbau der Pruefsumentabelle auf EPROM 00	152
4.11.	Aenderung des Strukturier-EPROM's	153
4.12.	Zuweisung der Funktionstasten durch den Anwender	153
4.13.	Hinweise fuer die Gestaltung von Anlauf und Verwaltung	153
5.	Gleitkommeprogrammpaket	161
5.1.	Allgemeine Hinweise	161
5.1.1.	Bestandteile des Gleitkommepaketes	161
5.1.2.	Zahlendarstellung	162
5.1.3.	Parameteruebergabe	164
5.1.4.	Fehlermeldungen	164
5.1.5.	Adressen Arithmetik	165
5.1.6.	Hinweise zu Aufwandsangaben	165
5.2.	Programmbeschreibung Arithmetikgrundmodul	166
5.2.1.	Gleitkommaaddition	ADD 166
5.2.2.	Gleitkommaabtraktion	SUB 166
5.2.3.	Gleitkomma multiplikation	MULT 167
5.2.4.	Gleitkomma division	DIV 167
5.2.5.	Quadratfunktion	SQU 168
5.2.6.	Kehrwert	RVS 168
5.2.7.	Betragsfunktion	ABS 169
5.2.8.	Konvertierung 16-Bit-Zweierkomplementdarstellung im Gleitkommaformat	FLOAT 169
5.2.9.	Konvertierung Gleitkomma darstellung in 16-Bit-Zweierkomplementdarstellung	TRUNC/ROUND 170
5.3.	Programmbeschreibung Arithmetikerweiterungsmodul	171
5.3.1.	Polynomberechnung	POLY 171
5.3.2.	Quadratwurzel	SQRT 172
5.3.3.	Exponentialfunktion	EXP 172
5.3.4.	Natuerlicher Logarithmus	LN 173
5.3.5.	Cosinusfunktion	COS 173
5.3.6.	Sinusfunktion	SIN 173
5.3.7.	Tangensfunktion	TAN 174
5.3.8.	Arcustangensfunktion	ARCTAN 174
5.3.9.	Dekadischer Logarithmus	LG 174
5.3.10.	Potensfunktion	POT 175
5.3.11.	Gleitkommavergleich	COMP 175
5.4.	Programmbeschreibung E/A-Konvertierungsmodul	176
5.4.1.	Eingabekonvertierung	GET 176
5.4.2.	Ausgabekonvertierung	PUT 177
6.	Editor zum BASIC-Compiler	183
6.1.	Einfuehrung	183
6.2.	Textformat	183
6.3.	Bedienung des Editors	183
6.3.1.	Start des Editors	183
6.3.2.	Arbeiten in Menue des Editors	184
6.3.3.	MENU	184
6.3.4.	ADDR	184
6.3.5.	NEW	185
6.3.6.	QUIT	185
6.3.7.	KEY	185
6.3.8.	KEYLIST	185
6.4.	Compileraufruf COMP	186

6.5.	Quelltextbearbeitung EDIT, TOP, BOTTOM	186
6.5.1.	Quelltextdarstellung	187
6.5.2.	Cursorbewegungen	187
6.5.3.	Cursorpositionierung	187
6.5.4.	Einfuegen/Streichen	187
6.5.5.	Copymode	187
6.5.6.	CAPLOCK	188
6.5.7.	Funktionstasten	188
6.5.8.	Verlassen des Edit-Modus	188
6.5.9.	RESET (CTRL)-Modus	188
6.6.	Zugriff auf periphere Gerate	191
6.6.1.	LIST	191
6.6.2.	SAVE	191
6.6.3.	LOAD	191
6.6.4.	PD	191
6.7.	Uebersicht Editor-Fehlermeldungen	192
7.	BASIC-Compiler, Sprachbeschreibung	193
7.1.	Einleitung	193
7.2.	Aufbau einer Programmzeile	194
7.3.	Namen	194
7.4.	Variablen	194
7.5.	Konstanten	195
7.6.	Arithmetische Ausdruecke	196
7.7.	Anweisungen	199
7.8.	Inline-Assembler	208
7.8.1.	Allgemeines	208
7.8.2.	Aktivierung des Assemblers	208
7.8.3.	Sprachumfang	209
7.8.4.	Verfuegbarkeit der Prozessorregister	209
7.9.	Systemkommando-Compiler	210
7.9.1.	Allgemeine Bemerkungen	210
7.9.2.	Steuerung des Echtzeitbetriebssystems	210
7.9.3.	Steuerung der Systemuhr	211
7.9.4.	Steuerung des FE-Status	212
7.9.5.	Zugriff auf Prozess- und Applikationsrechner- abbild	213
7.9.6.	Sonstiges	214
7.9.7.	Fehlerbehandlung bei Systemrufen	215
7.10.	Fehlermeldungen	216
8.	Schematasystem des Applikationsrechners mit Eigenstrukturierung	217
8.1.	Erstellung der statischen Bilder	217
8.1.0.	Zweck des Bildsystems	217
8.1.1.	Speicheraufteilung	217
8.1.2.	Systematik der Speicherung	219
8.1.3.	Dateiverwaltung	219
8.1.4.	Bedienung	219
8.1.5.	Festlegungen zum Zeichengenerator	223
8.1.6.	Sprachbeschreibung fuer statische Bild- informationen	225
8.1.7.	Hinweise zur Strukturierung der statischen Bilder	226

8.2.	Erstellung der dynamischen Bildelemente		227
8.2.1.	Wirkungsweise und Funktion		227
8.2.2.	Speicherfestlegungen und Vereinbarungen		227
8.2.3.	Dialogfuehrung		232
8.2.4.	Bildmodule		235
8.2.4.0.	Allgemeine Festlegungen und Moduluebersicht		235
8.2.4.1.	Einblendung von Bildmodulen	BMOD	240
8.2.4.2.	Einblendung einer Textzeile	BITX	241
8.2.4.3.	Blinken von Bildschirmbereichen	BIBL	243
8.2.4.4.	Einblendung mehrerer Textzeilen	BYTX	244
8.2.4.5.	Auswertung Alarmcode mit Texteinblendung	TXTF	247
8.2.4.6.	Bildeinblendung in Abhaengigkeit vom Alarmcode	FELD	249
8.2.4.7.	Textausgabe von Verarbeitungsprogrammen	HIPU	251
8.2.4.8.	Trenddarstellung von analogen Groessen	KUTR	252
8.2.4.9.	Darstellung von waagerechten und senkrechten Balken analoger Groessen	BALK	253
8.2.4.10.	Zahlendarstellung von Analogsignalen	FLKO	254
8.2.4.11.	Module zur Darstellung des aktuellen Anzeigeanfangs und -bereichs bei Kurzzeittrend und Balkendarstellung	ANBI	256
8.2.4.12.	Zahlendarstellung von Integerwerten	INTE	257
8.2.4.13.	Zahlendarstellung von Prozessgroessen, Gleitkommazahlen und Integerzahlen	ZIFF	258
8.3.	Bedienhandlungen zur Aktivierung der Strukturierung auf dem AR		261
8.4.	Abarbeitung der erstellten Bilder unter dem Echtzeitbetriebssystem des AR		262
8.4.1.	Prinzipielle Einordnung in die ON-LINE-Programmabarbeitung		262
8.4.2.	Notwendige und moegliche Zusatzfunktionen fuer die ON-LINE-Programmabarbeitung (Anwenderunterstuetzungsprogrammssystem)		263
8.4.2.1.	Grundlagen		263
8.4.2.2.	Handhabung des Anwenderunterstuetzungsprogramm-systems		267
8.4.2.3.	Kurzbeschreibung der einzelnen Programmbestandteile		268
8.5.	Anwendungsbeispiel		270
9.	Aenderung oder Erstellung des Zeichengenerators		275
9.1.	Uebersicht		275
9.2.	Startbedingungen		275
9.3.	Bearbeitungsablauf		275
9.4.	Ausgabe des Zeichengenerators		276
9.4.1.	Ausgabe auf Drucker		276
9.4.2.	Ausgabe auf FD/MB		276
9.4.3.	Programmierung der EPROM's		276
9.5.	Zusammenfassung der Bedienkommandos		277

3. Standardfunktionen des Systems (Bedienungsanleitung)

3.1. Funktionen - Ueberblick

Im System sind verschiedene Standardfunktionen implementiert, die der Systemstrukturierung, dem Systemtest, dem Funktionsnachweis, der Off-line-Bedienung und dem Programmtest dienen.

Im einzelnen handelt es sich um folgende Funktionskomplexe:

- Grundmenue
- Systemstrukturierung
- Systembedienung
- Monitor
- Hardcopy

Die Funktionen sind auf sw- und Farbbildschirm lauffaehig.

3.2. Grundmenue

Die Funktion "Grundmenue" ist als VAP im System eingebunden. Es dient vor allem der Off-line-Bedienung des AR. Alle im System eingebundenen VAP werden mit VAP-Nr. und VAP-Namen (aus VAP-Kopf) in Tabellenform angezeigt (siehe Bildschirmdialog!)

In der Kommunikationszeile sind folgende Eingaben moeglich:

- N ----> Blaettern in der Liste
- J Nr. ----> Programmstart: Ausloesen Neustart des VAP
- G ----> Umschalten des AR auf On-line
 Nach Eingabe von "G" erfolgt die Ausschrift
 "----> ON LINE". Erst nach Quittieren mit
 "ENTER" wird die Umschaltung ausgefuehrt.
- H ----> Umschaltung des AR auf Off-line
 Nach Eingabe von "H" erfolgt die Ausschrift
 "----> Off-line". Erst nach Quittieren mit
 "ENTER" wird die Umschaltung ausgefuehrt.

Die Eingaben N, J, G und H brauchen nicht quittiert zu werden. Fehleingaben werden nicht bearbeitet.

Bildschirmdialog:

GRUNDMENUE

```
00 GRUNDMENUE
01 STRUKTURIERUNG
02 SYSTEMBEDIENUNG
03 HACO
04
05
```

#M:X XX

<== Dialogzeile

3.3. Systemstrukturierung

Die Systemstrukturierung ist als VAP im System verfuegbar. Es umfasst folgende Teilfunktionen, die bei Programmstart als Menue angezeigt werden.

SYSTEMSTRUKTURIERUNG: X

```
0  PRIO 1
1  PRIO 2
2  PRIO 3
3  SYSTEMDATEN
4  ANMELDELISTE
5  STEUERLISTE VAP
6  PROZESSABBILD
7  AR-ABBILD
```

Der Start der Teilfunktion erfolgt ueber Anwahl der Nummern 0 bis 7.

Bei der Listenorganisation existieren 2 Varianten:

- Listen mit konstanter/definierter Listenlaenge (c)
- Listen mit variabler Laenge (v)

Die Anzeigelisten werden mit Positionsnummer und Belegung angezeigt. Ueber N ist ein Blaettern in der Liste moeglich. Werteingaben erfolgen im aktuell angezeigten Bereich ueber Anwahl der Positionsnummer. Dabei erfolgt die Eingabe zeilenweise. Nicht veraenderte Werte koennen quittiert werden. Bei Eingabe der Taste "OFF" bzw. Ende des Anzeigefeldes erfolgt der Ruecksprung zur Dialogzeile. Am Zeilenende erfolgt der Uebergang zur Folgezeile. Bei Listen mit variabler Listenlaenge ist neben der Worteingabe das Einfuegen und Streichen von Datenbloecken moeglich.

Zuweisung der Tasten:

```
INS
MODE ----> Worteingabe in Zeile
```

```
INS
LINE ----> Einfuegen von Zeile
```

```
DEL
LINE ----> Streichen der Zeile
```

Die Eingaben werden mit der Zeilennummer kombiniert. Die Datentypen sind bei den einzelnen Listen und Blockelementen unterschiedlich.

3.3.1. Funktion 0 bis 2

Aufgabe: Erstellung bzw. Korrektur der Prioritaetenliste
1 bis 3

Listentyp: variable Laenge (max. 31 Elemente)

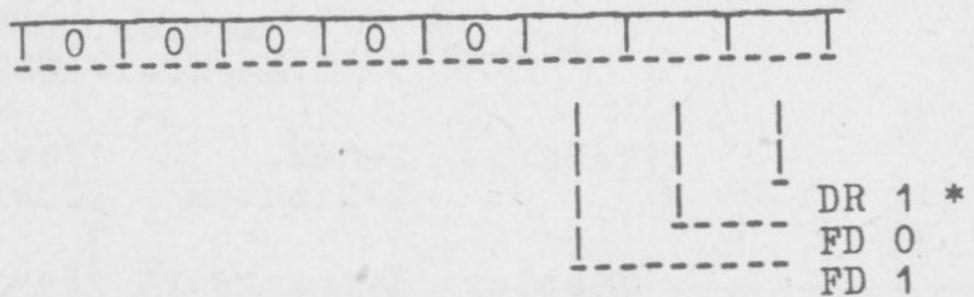
Blockaufbau: Eingabe der VAP-Nummer (Zuweisung des VAP zur
Prioritaetsebene fuer die On-line-Bearbeitung)

Bemerkungen: - VAP-Nr =< zul. Nummer
- jede VAP-Nr. darf nur einmal zugewiesen
werden

3.3.2. Systemdaten (Funktion 3)

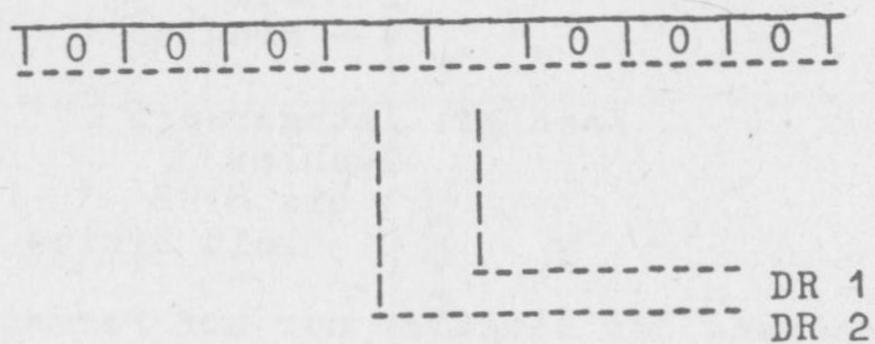
Es werden folgende Daten angezeigt und zur Eingabe angefordert:

- . maximale VAP-Nummer
- . VAP-Nr. Off-line
- . Geraetesteuerbyte 1
Belegung ISI 1



* Wenn nur eine ISI eingesetzt wird.

- . Geraetesteuerbyte 2
Belegung ISI 2



- . Unterstationsadressen fuer
angeschlossene BSE 1 bis 4: 4mal Hexa 1 Byte

Nicht belegte Unterstationen sind durch FF zu belegen!

3.3.3. Anmeldeliste (Funktion 4)

Aufgabe: Erstellung und Korrektur fuer die Programm Anmeldung bei der Umschaltung Off-line auf On-line

Listentyp: variable Laenge (max. 9 Elemente)

Blockaufbau: . VAP-Nr.: Integerwert

. Steuerbyte 1:

```
|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|-----|
```

|----- Anforderung zum Neustart 0/1

. Steuerbyte 2:

```
|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|-----|
```

|----- Zaehler: RST bei Nulldurchgang
 |----- Zaehler: NST bei Nulldurchgang

- . Takt: 2-Sek. } Verzoeigerung
 4-Minuten } fuer
- . Zaehler: Integerwert } Neustart oder Restart
 1 bis 8192
- . Takt: 0 - ohne Zyklus }
 1 - Grundtakt }
 2 - Sek. } Restartszyklus
 4 - Minuten } (Voreinstellwert)
- . Zaehler: Integerwert }
 Zaehler }
 1 bis 8192 }
 0 - kein Zyklus }

Ueberwachung der Eingabe: nur auf Formate

3.3.4. Steuerliste VAP

Aufgabe: Einbindung der VAP in das System

Listentyp: feste Laenge

- Blockaufbau: . Datenebene 1 bis 4 (Eingabe wird ueberwacht)
- . Programmebene 1 bis 4 (Eingabe wird ueberwacht)
- . Programmstartadresse (zeigt auf VAP-Kennung)
- . Steuerwert: Hexa 2 Byte

Variante	Steuerbyte 1	Steuerbyte 2	Zähler Verzögerung (s. Beisp.)	Zyklus- zähler (s. Beisp.)	Wirkung
1	1 000 0000	0000 0000	0	0	ohne Setzen Zykluszähler Neustart VAP kein Restart
			2	7	mit Setzen Zykluszähler
2	1 000 0000	0010 0000	2	7	Neustart VAP verzögerter Restart (7 sec) kein zyklischer Restart RST — RST — 7 sec
3			2	2	Neustart VAP verzögerter Restart (7 sec) zyklischer Restart (2 sec) RST — RST — RST — RST — 7 sec 2 sec 2 sec
4	0000 0000	1 000 0000	2	5	Verzögerter Neustart (5 sec) kein Restart 0 — RST — 5 sec
5			2	2	Verzögerter Neustart 0 — RST — RST — RST — 5 sec 2 sec 2 sec

3.3.5. Strukturierung des Prozessabbildes (PRAB)

Zur Strukturierung des PRAB wird folgendes Untermenue angeboten:

STRUKTURIERUNG PRAB: X

- 0 NEUSTART
- 1 VERWALTUNG
- 2 STEUERFELDER

Die Funktionswahl erfolgt ueber Eingabe der Nummern 0 bis 7. Bei der Strukturierung des PRAB erfolgt der Aufbau der Steuerfelder zur Verwaltung (Aktualisierung, Zugriff) des Prozessabbildes und der Anforderungstelegramme.

Grundzuweisungen stellen folgende Daten dar:

- Ebene
- Adresse Prozessabbild
- Laenge Prozessabbild
- Adresse Telegramme
- Laenge Telegramme

Die Bereiche werden dynamisch verwaltet, das heisst, die einzelnen Listen und Bereiche haben variable Laenge und werden dichtgespeichert. Der Umfang entspricht dem unter Punkt 1.10. angegebene Leistungsumfang. Bei der Strukturierung werden die Eingaben auf Sinnfaelligkeit geprueft. Bei Verletzung der Bereichslaengen wird die Eingabe abgelehnt.

- Prozessabbildtypen: 1 Byte/2 Byte/3 Byte/5 Byte
- Laenge pro Anforderungselement: 3 Byte
- max. 80 Einzelanforderungen
- Laenge Empfangstelegramm <= 256 Byte

3.3.5.1. Neustart

Bei Neustart der Funktion werden die Steuerlisten geloescht. Die Eingabe ist nur bei Neuzuweisung und nicht bei Korrektur auszufuehren. Die Eingabe ist durch weitere Quittungsanforderung ("J") verriegelt.

3.3.5.2. Verwaltung

Die Verwaltung erfolgt in mehreren Stufen:

- Grundzuweisung fuer Ebene, Adressen, Laenge
- Reservierung der Prozessabbildbereiche mit 1, 2, 3 und 5 Byte (Pestlegung der Elementezahl)

Dialog:

GRUNDRESERVIERUNG PRAB

GEN.-DATEN VOM EPRON

B:	A-PRAB:	L-PRAB:	A-TEL:	L-TEL:
X	XXXX	XXXXX	XXXX	XXXXX
PRAB 1:	XXX	} Elementezahl für die 4 Bereiche (1/2/3/5 Byte)		
PRAB 2:	XXX			
PRAB 3:	XXX			
PRAB 4:	XXX			

- Grundzuweisung der BSE, PRAB-Typ und Laengen fuer die 3 verschiedenen Zeittakte

Dialog:

PRAB TAKT 1(2 SEC)
ANZAHL BLOECKE: XX

	PE	TYP	ANZAHL	
00	X	X	XX	} Grundreservierung für die Anforderungsblöcke, die angeforderte Anzahl entspricht der Blockanzahl
01	X	X	XX	
02	X	X	XX	
03	X	X	XX	
04	X	X	XX	
05	X	X	XX	
06	X	X	XX	
07	X	X	XX	
08	X	X	XX	

A C H T U N G !

Die Verwaltung ist bei Korrektur stets komplett abzuarbeiten, da bei jeder Eingabe Steuerlisten korrigiert werden. Dabei sind von vornherein Reserven zu strukturieren, da sonst der Anschluss der Datenverarbeitung bei jeder Korrektur angepasst werden muss.

3.3.5.3. Steuerfelder

Ueber diese Funktion werden die einzelnen Werte des PRAB zugewiesen (Aufbau der Sendetelegramme fuer das Prozessabbild). Pro Element sind Datentyp IMEN und Relativadresse zuzuweisen. Dazu sind Taktnummer (1*2 sec/2*6 sec/3*60 sec) und Anforderungsnummer einzugeben.

Falls das Anforderungselement entsprechend 3.3.5.2. reserviert ist, werden Funktionseinheit (1 bis 4), Typ des PRAB (0 bis 3), reservierte Elementzahl und Anfangsnummer im Prozessabbild angezeigt.

Die einzugebende Belegung kann kleiner oder gleich der reservierten Elementzahl sein.

Die einzelnen Elemente werden in Tabellenform entsprechend Bild strukturiert.

Listentyp: konstante Laenge (* Belegung)

Blockaufbau: . Datentyp - Ueberwachung auf Zuverlaessigkeit
 . IMEN (0 ... 255)
 . Relativadresse (0 ... 34)

Fuer verschiedene Felder des Prozessabbildes sind folgende Datentypen zulaessig (siehe Punkt 1.1.1.):

PRAB Typ-Nr. Zugriff auf Relativadr. von Bedeutung

1 Byte	15	IW Aggregate KOM	-
	16	IW-Leit KOM	-
2 Byte	17	IW Bin.Geber	-
	18	1 Byte aus KOM	x
	19	bin. Merker	-
3 Byte	21	IW analoger KOM	-
	22	2 Byte aus KOM	x
	23	analoger Merker	-
5 Byte	25	IW Zaehl KOM	-
	26	4 Byte aus KOM	x

IW = Istwert

Fuer den gezielten Zugriff auf Werte (ausser Istwert) im KOM sind zur Festlegung der Relativadresse /2 / und /3 / zu nutzen. Die maximale IMEN der jeweiligen BSE ist vom Anwender zu beachten.

3.4 Systembedienung

Die Systembedienung ist als VAP im System verfuegbar. Es umfasst folgende Teilfunktionen, die bei Programmstart als Menue angezeigt werden.

SYSTEMBEDienung/ ANZEIGE:

- 0 ZUGRIFF PROZESSABBILD/ AR ABBILD
- 1 DATENEINGABE IN BSE
- 2 ANZEIGE/ EINGABE HEXA
- 3 FEHLERZUSTAND
- 4 FEHLERPUFFER
- 5 STATUS AR
- 6 AUSGABE SD
- 7 BEDienung FDE
- 8 EPROM PROGRAMMIERUNG
- 9 UHRZEIT/ DATUM

Die Anwahl der Funktionen erfolgt ueber Anwahl der Nummer 0 bis 9.

3.4.1 Zugriff auf Prozessabbild und AR Abbild

Die Funktion wird mit folgendem Untermenue gestartet:

ANZEIGE/ EINGABE PROZESS ABBILD + AR ABBILD

- 0 AKTUALISIERUNG AUS
- 1 AKTUALISIERUNG-EIN
- 2 ANZEIGE PRAB/AR-AB
- 3 ZUGRIFF PRAB/AR-AB

Der Start der Funktion erfolgt ueber Anwahl der Nummer 0 3. Die Eingabe von "OFF" fuehrt zurueck zum Menue Systembedienung.

Ueber die Funktion 0 und 1 ist fuer Test und Simulationszwecke die Aktualisierung des Prozessabbildes ein und ausschaltbar.

3.4.1.1 Anzeige PRAB/ AR Abbild

Es werden Typ Nummer und Konvertierungstyp angefordert.

Typ: 0...6 (0 3 PRAB /
4 6 AR-Abbild)

Nummer: Anfangennummer der Anzeige fuer den angewählten Typ.

Konvertierungstyp: Da im jeweiligen Abbild verschiedene interpretierbare Daten abgelegt sind, ist der Anzeigetyp durch den Anwender wahlbar.

Name		Datum		Nr. AG		Anlagen-Nr.		AG-Nummer		Auftraggeber		Anzahl		Ort	
Empf. Prof.															
Stabs. Prof.															
AG-Nr.		AG-Nr.		AG-Nr.		AG-Nr.		AG-Nr.		AG-Nr.		AG-Nr.		AG-Nr.	

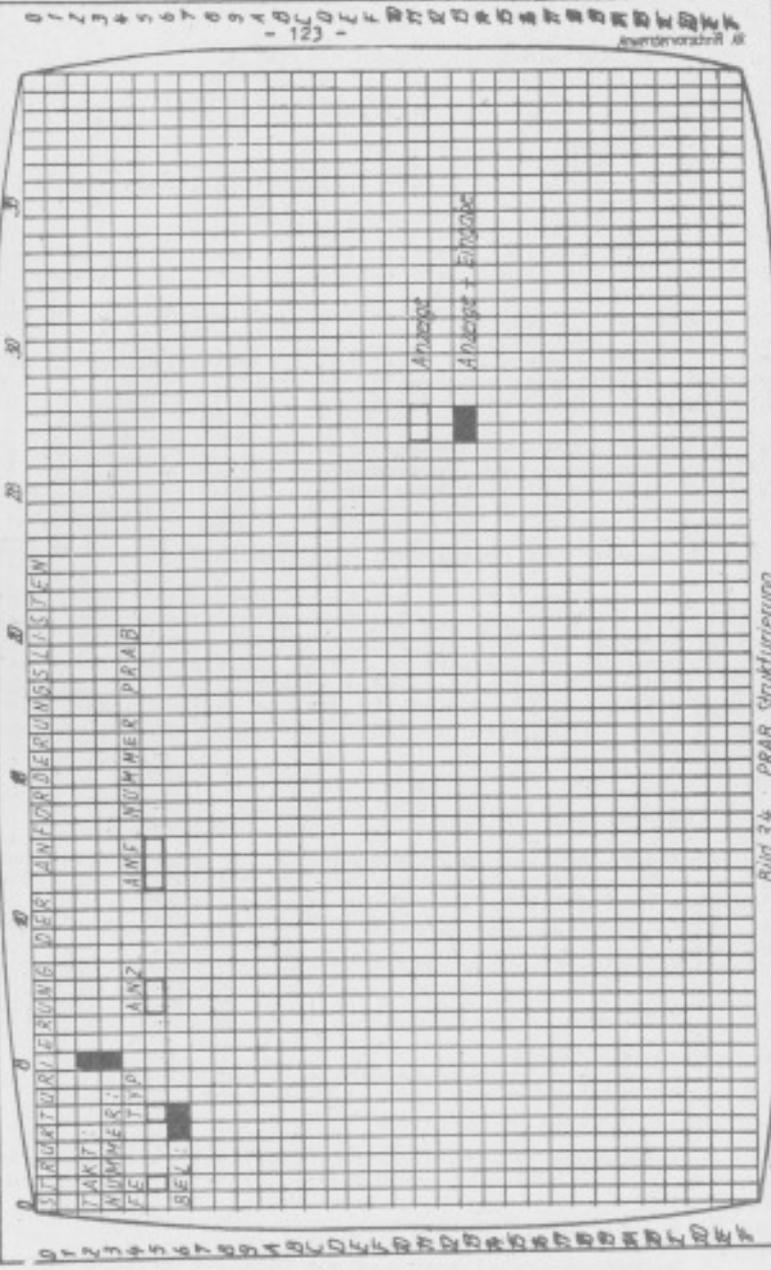


Bild 3.4.: PRAB-Strukturierung

Es sind folgende Anzeigevarianten moeglich:

Nummer	Feld	Konvertierungstyp
0	PRAB 1 Byte	
1	PRAB 2 Byte	Integer, Binaer, Hexa
2	PRAB 3 Byte	Integer, Hexa, Festkomma
3	PRAB 5 Byte	Hexe, Integer
4	AR-Abbild 2 Byte	Integer, Binaer, Hexa
5	AR-Abbild 3 Byte	Integer, Hexa, Festkomma
6	AR-Abbild 5 Byte	Hexa, Integer, Gleitkomma

Die Anzeige der Elemente erfolgt getrennt fuer Statusbyte und Wert (siehe Tabelle).

Nach Eingabe der angeforderten Parameter erfolgt die zyklische Aktivierung von jeweils 10 Werten.

Dabei sind folgende Eingaben moeglich:

- ↑ ↓ Verschiebung des Anzeigebereiches
- OFF Ruecksprung zum Unterprogramm
(uebergeordneten Programmen)

3.4.1.2. Zugriff auf Prozessabbild und AR-Abbild

Die Eingabe von Typ, Nummer und Konvertierungstyp entspricht Punkt 3.3.1.1. Nach Anzeige des Wertes ist eine Neuweisung von Statusbyte und Wert moeglich. Mit der Taste "OFF" kann die Eingabe uebergangen werden. Nach Quittierung bzw. bei "OFF" wird zu Typeingabe zurueckgekehrt. Bei der Eingabe erfolgt ueber "OFF" ein Ruecksprung in der Eingabe. Bei Eingabe der "OFF"-Taste bei Typeingabe wird zum Grundmenue zurueckgekehrt.

3.4.2. Dateneingabe in BSE

Die Funktion erlaubt Daten aus Merker und KOM-Blocken zu lesen und zu veraendern. Fuer die Interpretation der KOM-Blocke sind Strukturinformationen abgelegt (siehe und). Die Funktion greift mit Prioritaet 2 auf die Datenuebertragung zu.

Es sind folgende Eingaben erforderlich:

BSE: 1 bis 4

ZUGRIFF: 0 KOM-Block
 1 Binaere Merker (Anzeige: Binaer)
 2 Analoge Merker (Anzeige: Integer)
 3 Analoge Merker (Anzeige: Festkomma)

Der Konvertierungstyp fuer KOM-Blockdaten ist wertabhaengig (Format: Binaer, Hexa, Festkomma, Prozessgroesse-dimensioniert)

Nummer: IMEN bei KOM-Blocken
 Merkernummer bei analogen und binaeren Merkern
 Nummer: Parameter (nur bei KOM-Blocken)

Nach Anwahl der IHEN wird der KOM-Typ angezeigt:

- A - analog stetig
- U - analog unstetig
- Z - Zähler-KOM
- B - Binärer Aggregate-KOM
- G - Geber-KOM
- L - Leit-KOM

Die angegebenen Anwahlnummern werden auf Zulässigkeit geprüft. Bei Zugriff auf nicht zulässige KOM-Typen erfolgt eine Meldung. DUE-Fehler werden ebenfalls signalisiert. Nach Anwahl des Wertes erfolgt eine zyklische Aktualisierung des Wertes im entsprechenden Datenformat. Dabei sind folgende Eingaben möglich:

- ↑ ↓ Veränderung des Zugriff - Merker
- Parameternummer im
KOM-Block
- OFF Abbruch der Anzeige und Uebergang zur Eingabe

Ueber Werteingabe ist eine Wertaenderung möglich. Nach Quit-tierung bzw. "OFF"-Taste erfolgt der Ruecksprung zur Anwahl.

3.4.3. Anzeige der Eingabe in Hexa

Diese Funktion umfasst folgende Teilfunktionen :

- Anzeige von Daten in Hexaformat entsprechend Steuerliste
- Dateneingabe im Hexaformat und ASCII.
- Kommandofunktion

Die Bedienung erfolgt ueber Kommunikationszeile. Es sind folgende Eingaben möglich.

- D Aufbau der Steuerliste zur Hexaanzeige
- S Eingabe von Daten
- K Kommando
- OFF Ruecksprung zum Menue Systembedienung
- A Dateneingabe mit zyklischer Aktualisierung

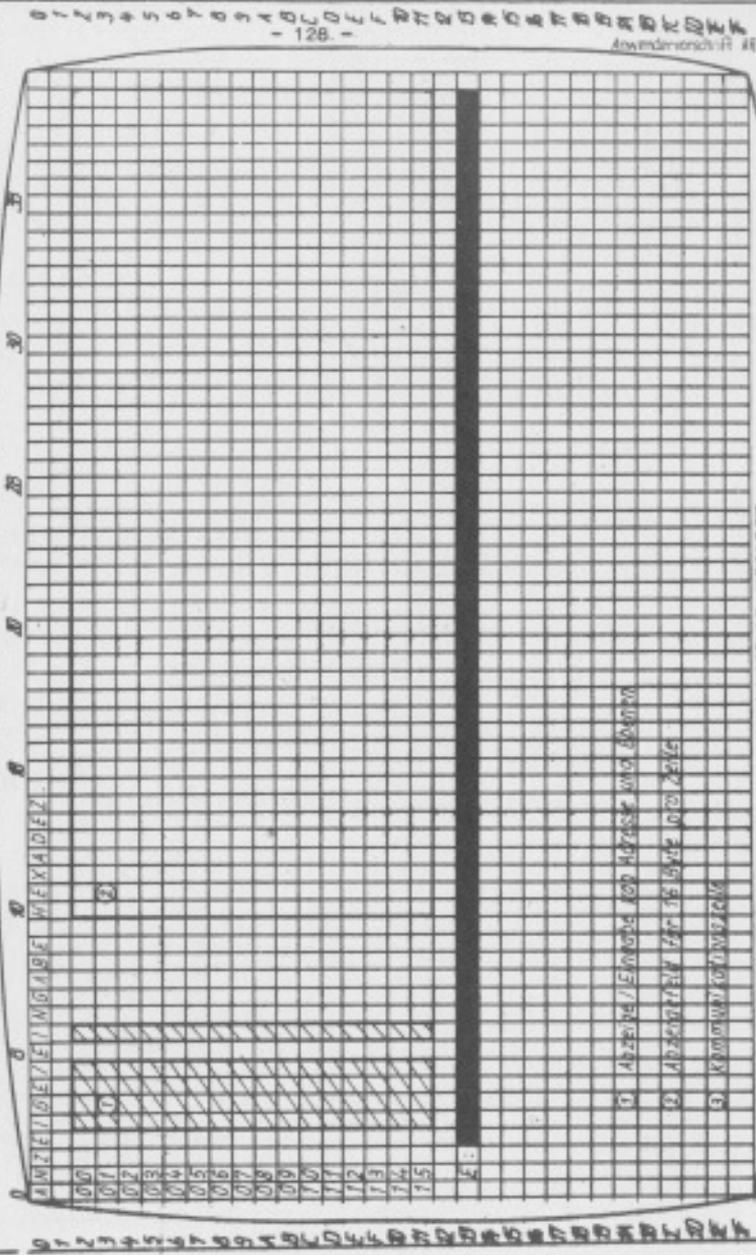
3.4.3.1. Aufbau der Steuerlisten zur Anzeige

Nach Funktionswahl ueber "D" wird die Eingabeseile angefordert.

Das Eingabefeld kann in beliebig viele Teilfelder zerlegt werden. Die Eingabe von Datenadresse und Ebene erfolgt in der Tabelle in der linken Spalte. Dabei werden alle folgenden Zeilen bis zum naechsten zugewiesenen Anzeigesatz fortlaufend belegt. Die Anzeige der Datenebene fuer Folgezeilen wird zur Merklung unterdrueckt.

Zur Anzeige stehen 16 Zeilen (0 bis 15) zur Verfuegung.

Name		Datum		Nr.		AS		Anforderung		Anfordergeber		Name		Datum	
Stabschef		Stabschef		Stabschef		Stabschef		Stabschef		Stabschef		Stabschef		Stabschef	
MSE-M		VA		AS-M		LAP		AD		AS-M		VA		AS-M	



- 1) Abzähl-/Eingabe Top-Hexadez und Bottom
- 2) Abzähl-/Eingabe Top-Hexadez und Bottom
- 3) Kombinationen

Bild 3.4.3 EINGABE / ANZEIGE HEXADEZ.

Name		Datum		M. AS	Andrager-Nr.	AS	Andrager	Auftraggeber	Kasse	Ordnung	100. Nr.:	
Begriff		Prüfung									ASZ-Mit	100
											100-100	100
											100-100	100
											100-100	100

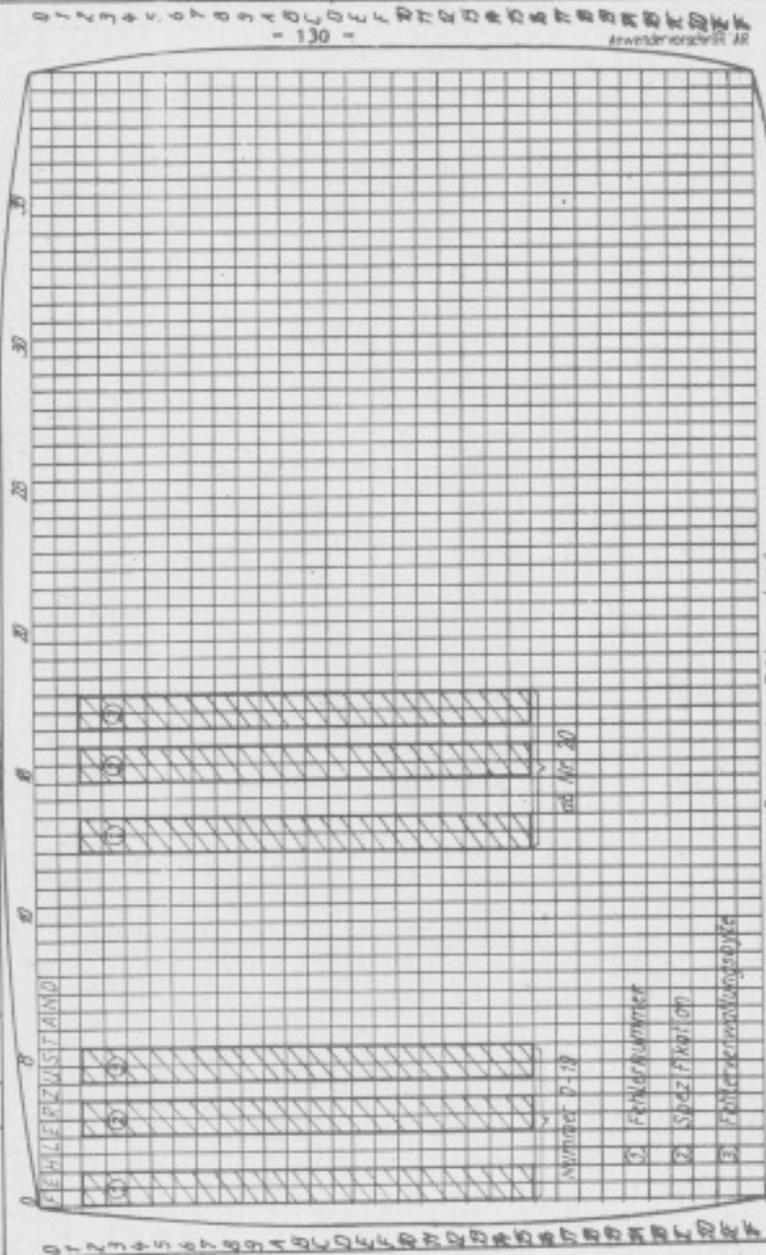


Bild 3.4.5. Anzeige Fehlerzustand

Des weiteren ist strukturierbar, mit welcher Taste die Hardcopyfunktion ausgeloesst werden soll und auf welchem Gerat die Ausgabe erfolgen soll (nur Drucker sind sinnvoll). vgl. Tab. Belegung Strukturier-EPRCH Abschnitt 4.9.

Geratenummer:	516DH	;SGN Drucker
Anwahltaste:	566EH	;OFFH, keine Taste, sonst Tastencode

3.4.6. Bedienung der FDE (Floppy-Disk-Einheit)

Die FDE wird ueber die ISI vom Applikationsrechner angesteuert. Sie enthaelt 2 Laufwerke fuer 5 1/4"-Disketten, die von unterschiedlichen Funktionseinheiten des Systems audatec oder auch beide durch den Applikationsrechner genutzt werden koennen. Mit Hilfe der FDE kann der Nutzer Programme oder Daten auslagern, um sie spaeter auf diesem Gerat oder einem anderen wieder einzulesen.

Die Aufzeichnung auf den Disketten erfolgt im SCPX-Format wahlweise 2x 148k oder 1x 624k und ist damit kompatibel zu Buero- und Personalcomputern.

Im audatec-Applikationsrechner besteht die Moeglichkeit, Dateien mit einem Kopfblock zu versehen. Das ist ein SOH langer Block, der den eigentlichen Daten vorangestellt ist und die Anfangsadresse, Laenge und Ebene enthaelt, so dass diese Informationen beim Laden nicht angegeben werden muessen. Eine Datei kann auch mehrere Kopfbloেকে enthalten, wenn sie abschnittsweise auf verschiedene Bereiche geladen werden soll.

Ebenso ist die Arbeit mit BAK-Dateien moeglich, wie sie unter SCPX bekannt sind. Das heisst, beim Schreiben ist immer noch ein Sicherheitsdoppel der Datei vorhanden.

Bei allen Bedienfunktionen kann parallel eine Seite der Bibliothek angezeigt bleiben. Ueberall dort, wo Dateinamen angefordert werden, kann auch die Nummer der Datei angegeben werden, wenn sich diese auf der angezeigten Bildschirmseite befindet.

Nach dem Einschalten der FDE (Netz ein, RESET) muss die FDE-Systemdiskette eingelegt werden. Nach Systemausfall ist ebenfalls mittels RESET die Systemdiskette neu zu laden. Danach kann sie durch eine Anwenderdiskette ersetzt werden.

Nach Aufruf der Funktion **BEDIENUNG FDE** im **VAP SYSTEMBEDIENUNG/ANZEIGE** meldet sie sich mit der Ausschrift **FD KDO**. Hinter **FD** ist die Laufwerksnummer 0 oder 1 einzutragen. Wird **OFF** eingegeben, kann das Programm hier wieder verlassen werden. Ist das Laufwerk nicht zuweisbar, erfolgt eine Fehlerausschrift. Diese erfolgt mit Fehler-Nr. und Klartext ueber die Bedienseite.

Mögliche Fehlerursachen sind:

- FD-Laufwerk durch anderes Geræat belegt
- Geræat durch anderes VAP belegt
- Systemprogramm in der FDE nicht gestartet
- Geræat/Kabel defekt
- Diskette nicht initialisiert (Sektor nicht gefunden)
- Diskette nicht vorhanden (Sektor nicht gefunden)

Das Format der Diskette wird automatisch erkannt und eingestellt. Erfolgt die Ausschrift "Sektor nicht gefunden", ist die Diskette evtl. zu initialisieren.

Ist das Laufwerk verfuegbar, wird anschliessend über einen Kommandobuchstaben die gewünschte Funktion aufgerufen.

H	Hilfsmenue
B	Bibliothek
I	Initialisieren
Y	Systemdiskette duplizieren
D	Alle Dateien duplizieren
X	Fehler-/Statusabfrage
F	Laufwerk freigeben
K	Datei kopieren
C	" streichen
R	" umbenennen
L	" lesen (D, KR, KA)
S	" schreiben (D, DB, K, KB)
W	" weiterschreiben (" ")

OFF Rueckkehr zur Geræateanforderung

Typ:

- D - direkt
- K - mit Kopfblock
- B - mit BAK-Datei schreiben
- A - absolut lesen
- R - relativ lesen

3.4.6.1. Hilfs-Menue H

Mit diesem Kommando werden alle möglichen Floppy-Disk-Funktionen und ihre Kommandobuchstaben aufgelistet. Es wird dazu der rechte obere Bildschirm belegt. Eine evtl. angezeigte Bibliothek wird nicht zerstoert.

3.4.6.2. Bibliothek auflisten

Mit diesem Kommando kann das Bibliotheksverzeichnis einer Diskette auf dem Bildschirm angezeigt werden. Der gesamte Bildschirm wird benoetigt. Ein evtl. angezeigtes Hilfs-Menue wird nicht zerstoert.

Nach Aufruf des Kommandos wird die Anzahl der Dateien, die Diskettenkapazitaet insgesamt und die davon noch freie Diskettenkapazitaet angezeigt.

Das Inhaltsverzeichnis wird von oben beginnend in zwei Kolonnen ausgedruckt. Von jeder Datei wird die Nummer, der Name/Typ und die Laenge der Datei angezeigt. Es sind 40 Dateien auf einer Seite darstellbar, mit ENTER kann umgeblattet werden und mit OFF oder einem beliebigen anderen Buchstaben zur Kommandoingabe zurueckgekehrt werden.

Mit Statusmeldung 14 ist das Ende der Bibliothek erreicht.

Am Bibliotheksende wird mit ENTER wieder der Anfang angezeigt.

3.4.6.3. Diskette initialisieren I

Disketten, die auf der FDE (Floppy-Disk-Einheit) oder unter SCPX genutzt werden sollen, sind vor Gebrauch zu initialisieren.

Beim Initialisieren wird der Disketteninhalt zerstoert.

Nach Kommandostart werden die Versions-Nr. und das Erstellungsdatum der Betriebssystemsoftware der FDE angezeigt.

Anschliessend wird angefordert, ob die Diskette im 148k-Format (einfach, einseitig) oder im 624k-Format (doppelt, zwei-seitig) initialisiert werden soll. Das kann durch beliebigen Tastendruck eingestellt werden und ist mit ENTER zu bestaetigen. (Beim 624k-Format wird die Rueckseite der Diskette mit beschrieben !)

Das Programm kehrt sofort wieder zur Kommandoingabe zurueck, so dass andere Funktionen des Applikationsrechners zwischenzeitlich bedient werden koennen.

Nach dem Initialisieren kann ueber X (Statusabfrage) ermittelt werden, ob waehrend des Initialisierens Fehler aufgetreten sind. Wurde das Programm zwischenzeitlich verlassen, wird bei Wiederanwahl des Laufwerks nur der Fehlercode (ohne Text) zur Anzeige gebracht. Ist das Initialisieren noch nicht abgeschlossen, wird Fehler 92 angezeigt.

3.4.6.4. Systemdiskette duplizieren Y

Mit diesem Kommando kann die Systemdiskette dupliziert werden, weil hier die Systemspuren mit kopiert werden.

Nach Ausschrift des Textes:

SYSTEM-DISK DUPL wird hinter ==> die Laufwerks-Nr. des Ziellaufwerkes ausgeschrieben. Nach Bestätigung mit ENTER beginnt der Dupliziervorgang. Das Programm kehrt sofort zur Kommandoeingabe zurück, damit die weitere Arbeit mit dem Applikationsrechner möglich ist.

3.4.6.5. Alle Dateien duplizieren D

Mit diesem Kommando kann eine komplette Diskette dupliziert werden.

Nach Ausschrift des Textes: ALLE DAT.DUPL ist die Bedienung analog 3.4.6.4.

3.4.6.6. Fehler-/Statusabfrage X

Während oder direkt nach Ausführung eines langen Kommandos (I, Y, D) kann der Fehler oder Status abgefragt werden, da diese Kommandos sofort nach Start ohne Fehlermeldung in die Kommandoeingabe zurückkehren.

3.4.6.7. Laufwerk-Freigabe F

Dieses Kommando braucht nur genutzt zu werden, wenn zwei audatec-Funktionseinheiten mit einer FDE verbunden sind.

Mit Bestätigung der Ausschrift: LAUFWERK FREIG. kann dieses Laufwerk wieder durch die andere Funktionseinheit belegt werden.

3.4.6.8. Datei kopieren K

Es kann eine Datei auf die Diskette, die im jeweils anderen Laufwerk ist, kopiert werden. Nach Kommandoeingabe wird der Name der zu duplizierenden Datei angefordert. Daraufhin wird die Datei gesucht, und wenn sie vorhanden ist, wird das andere Laufwerk belegt. Es wird hinter ==> die Ziellaufwerksnummer angezeigt. Wird die Anzeige mit ENTER bestätigt, beginnt der Kopiervorgang.

3.4.6.9. Datei streichen C

Nach Kommandoaufruf wird der Name angefordert. Ist die Datei vorhanden, wird sie gestrichen und das Programm kehrt zum Kommandoaufruf zurück.

3.4.6.10. Datei umbenennen R

Es wird der Name der umzubennenden Datei sowie der neue Name angefordert. Nach Bestätigung mit ENTER wird die Datei umbenannt und das Programm kehrt zum Kommandoaufruf zurück.

3.4.6.11. Datei lesen L

Es ist der TYP anzugeben:

- D - direkt (ohne Kopfblock)
- KA - mit Kopfblock absolut
- KR - mit Kopfblock relativ

Anschliessend wird der Name der Datei angefordert. Ist zugleich die Bibliothek auf dem Bildschirm angezeigt, kann auch anstelle des Namens die Nummer der Datei angegeben werden. Dieser wird dann herausgesucht und muss mit ENTER bestätigt werden. Es ist auch möglich, bei nicht vollständig bekannten Namen '?' fuer die fehlenden Zeichen einzugeben (z.B.: XXXXX???.YYY).

Wurde der TYP KA angegeben, wird die Datei sodann geladen. Die Kopfinformationen Anfang/Ebene, Ende werden laufend angezeigt.

Beim TYP KR sind die angezeigten Adressen fuer Anfang/Ebene mit ENTER zu bestätigen oder koennen korrigiert werden, so dass die Datei auf die neue Adresse/Ebene geladen wird.

Beim TYP D muss eine Anfangsadresse/Ebene angegeben werden sowie eine Endadresse, die nicht ueberlesen werden soll.

Bei allen Eingaben ist ein Ruecksprung mit OFF moeglich. Nach dem Laden kehrt der Cursor zur Kommandoabfrage zurück, es sei denn, die Datei konnte aus Speicherplatzgruenden nicht vollständig geladen werden; dann kann eine neue Anfangsadresse/Ebene angegeben werden, um den Rest zu laden oder es wird mit OFF abgebrochen. Nach dem Laden wird die erreichte Endadresse angezeigt.

3.4.6.12. Datei schreiben S

Es ist der TYP anzugeben:

- D - direkt (ohne Kopfblock)
- DB - direkt mit BAK-Datei
- K - mit Kopfblock
- KB - mit Kopfblock und BAK-Datei

Anschliessend wird der Name angefordert. Er ist wie unter SCPX aufgebaut:

XXXXXXXX.YYY

x : Buchstabe/Ziffer
y : Typbezeichnung

Die Typbezeichnung im Namen ist nicht mit dem zuvor angegebenen Typ identisch, sondern kann unter SCPX zur Dateikennzeichnung verwendet werden.

Anschließend wird die Anfangsadresse/Ebene/Endadresse angefordert. Wie die Datei mit Kopfblock geschrieben, sind mehrere Bereiche angebar; als Abschluss ist hinter END eine 0 einzutragen.

Soll eine Datei mit BAK-Datei geschrieben werden, muss eine Datei gleichen Namens vorhanden sein, die zur BAK-Datei gemacht wird. Ist das nicht der Fall, wird die Datei nur einfach geschrieben.

Bei allen Eingaben ist ein Ruecksprung mit OPF moeglich. Nach dem Schreiben kehrt der Cursor zur Kommandoabfrage zurueck.

3.4.6.13. Datei weiterschreiben W

Es ist moeglich, eine schon vorhandene Datei fortzuschreiben. Die Bedienung erfolgt analog zu 3.4.6.12.

Soll eine nicht vorhandene Datei weitergeschrieben werden, so wird sie neu eroeffnet.

3.4.6.14. Ausgewahlte Fehler-/Statusmeldungen (vgl.auch 1.9.1.)

- | | |
|----|---------------------------------------|
| C1 | Geraet nicht strukturiert |
| AD | Systemprogramm der FDE laeuft nicht |
| 92 | eigenes Kommando laeuft noch |
| 9B | Datenebertragung zur FDE unterbrochen |
| 20 | Geraetefehler |
| | - Sektor nicht gefunden |
| | - Laufwerk nicht zugewiesen |
| | - Datei existiert schon |
| | - Datei existiert nicht |
| | - Diskette voll |
| | - Dateiverzeichnis voll |
| | - Fehler beim Formatieren |
| 14 | Dateiende beim Lesen |
| 11 | Schreibdatei eroeffnet |

3.4.7. EPROM Programmierung und Pruefsummenbestimmung

Das Programm ist zur Arbeit mit 1K EPROM vom Typ U555 (2708) vorgesehen und kann nur in OFF-line-Zustand des Rechners arbeiten. Im Rechner muss sich eine Anschlusskarte zur PROM-Programmierung PPE 0420 mit einer PROM-Aufnahme PAB X 0422 befinden.

Die Hardwareadresse der PPE ist auf 00H einzustellen. Nach Aufruf des Programms meldet es sich auf der letzten Bildschirmseite mit der Aufschrift entsprechend Bild 3.4.8. Aus folgenden Programmfunktionen kann durch Eingabe eines Kommandobuchstabens ausgewaehlt werden.

- L EPROM lesen
- P EPROM programmieren
- C Pruefsumme (CRC) vom Speicher bestimmen
- R Rekursive Pruefsumme vom Speicher (RAM) bilden
- V EPROM mit Speicher vergleichen

Mit OFF kann das Programm wieder verlassen werden. Anschliessend ist die Ebene (1-4) und der hoehwertige Adressteil des Speichers anzugeben, mit dem gearbeitet werden soll.

Fuer alle Funktionen wird ein 1k-Zwischenpuffer ab 6000H genutzt.

Bei Abarbeiten einer Funktion werden also Daten, die zuvor auf diesen Bereich standen, zerstoeert.

Mit Eingabe von OFF kann zur jeweils vorhergehenden Anforderung zurueckgekehrt werden.

Vor Quittierung der Adresse muss der EPROM gesteckt worden sein. Der folgende Ablauf ist funktionspezifisch.

EPROM lesen

Der EPROM wird auf den Zwischenpuffer gelesen und auf den angegebenen Adressbereich (1K ab Anfangsadresse) ungespeichert. Die Pruefsumme des EPROM wird angezeigt.
Moeglicher Fehler: OB-angegebener Adressbereich kein RAM

EPROM programmieren

Um ein versehentliches Programmieren zu vermeiden, ist die Eingabe noch einmal mit P zu quittieren. Anschliessend wird der angegebene Adressbereich auf den Zwischenpuffer ungespeichert. Der EPROM wird zuerst auf Programmierfaehigkeit ueberprueft. Das heisst, es darf kein Bit mit 0 belegt auftreten, das mit 1 programmiert werden soll. Fehler werden wie unten beschrieben angezeigt.

Die Programmierzyklen beginnen. Der Zykluszaehler wird auf 1 runtergezuehlt und angezeigt. Mit OFF kann die Programmierung unterbrochen werden. Die Programmierdauer betraegt ca. 2 min. Nach Abschluss der Programmierung wird die Pruefsumme angezeigt. Die erste Fehleradresse des EPROM, Inhalt des EPROM Inhalt alt des Speichers werden angezeigt. Durch Betaetigen der Leertaste wird die naechste fehlerhafte Adresse angezeigt.

SE Bl. Nr.:			140-141	141	142-143	143	144	145	146	147	148	149	150
Name	Datum	Nr.	AG	Anwendung	Auftraggeber	Name	Ort						
Nach Prüf.	Name	Datum											
Bild 3.4.8													
Legend: Anwendungsvorschrift AS, Maßstab 1:1													
ausdrucken													

3.5. Monitor

Mit Hilfe des Monitors ist es möglich, Anwenderprogramme abschnittsweise und im Echtzeitlauf zu testen.

In den Monitor gelangt man entweder durch betätigen der Taste "ERASE INP", oder wenn, wie im folgenden beschrieben, ein Break -oder Anzeigepunkt gesetzt ist oder bei Programmfehler wenn der Befehl RST 7 (OPPH) erkannt wird.

Es folgt eine Ausschrift gemäss Bild 3.5.1 auf den letzten 9 Bildschirmzeilen. Eventuell vorhandene Anzeigen der Anwenderprogramme in diesem Bildschirmbereich werden ueberschrieben.

Unter den Registernamen wird erst der Erstregistersatz und dann der Zweitregistersatz angezeigt. Dann folgt der Interruptvektor. Unter PL wird der Befehlszahlzustand und die ersten 4 Byte des Inhalts des PC angezeigt. Es folgt der Stackpointer SP und unter (SP) werden die letzten beiden Werte des Stackinhalts angegeben. Es schliessen sich der Inhalt der Adressen HL und DE an. Unter 74 wird die Belegung der ZRE PIO Tor B angezeigt. Das erste Halbbyte schaltet die 7000' der Speicherebenen, das zweite Halbbyte die 1000' der. Darunter wird der Inhalt der PIO Tor A (ISI Interrupt und Reset-Belegung) angezeigt.

In der letzten Zeile hinter den # koennen folgende Kommandos eingetragen werden:

```
R   Register schreiben
B   Break (Unterbrechungspunkt) setzen
A   Anzeigepunkt stetzen
S   Speicher schreiben
D   Dump , Speicheranzeige
G   Go , fortsetzen
V   VAP abbrechen
M   Move, Speicher verschieben
F   Fuellen Speicher
RESET Rechner Reset
```

Alle Kommandoeingaben koennen mit OFF wieder verlassen werden, wenn sie nicht gueltig sind. Die letzten 5 gueltigen Kommandoeingaben werden nach oben gerollt und bleiben zur Protokollierung erhalten.

Register schreiben R

Nach Anwahl des Kommandos R wird der Erstregistersatz sowie der PC und SP ausgeschreiben. Saemtliche Werte koennen ueberschrieben werden und werden erst mit Quittierung durch Enter uebernommen. (Beispiel siehe Bild 3.5.2 a)

Break (Unterbrechungspunkt) setzen B

Nach Eingabe des Kommandos B wird die Adresse und Ebene angefordert, bei der der Programmauf unterbrochen werden soll. Es koennen nur Softwarebreaks gesetzt werden, d.h. das zu testende Programm muss auf RAM stehen. Auf der angegebenen Adresse wird ein RST 7 (OPPH) eingetragen der bei Erreichen des Breaks wieder zurueckgesetzt wird.

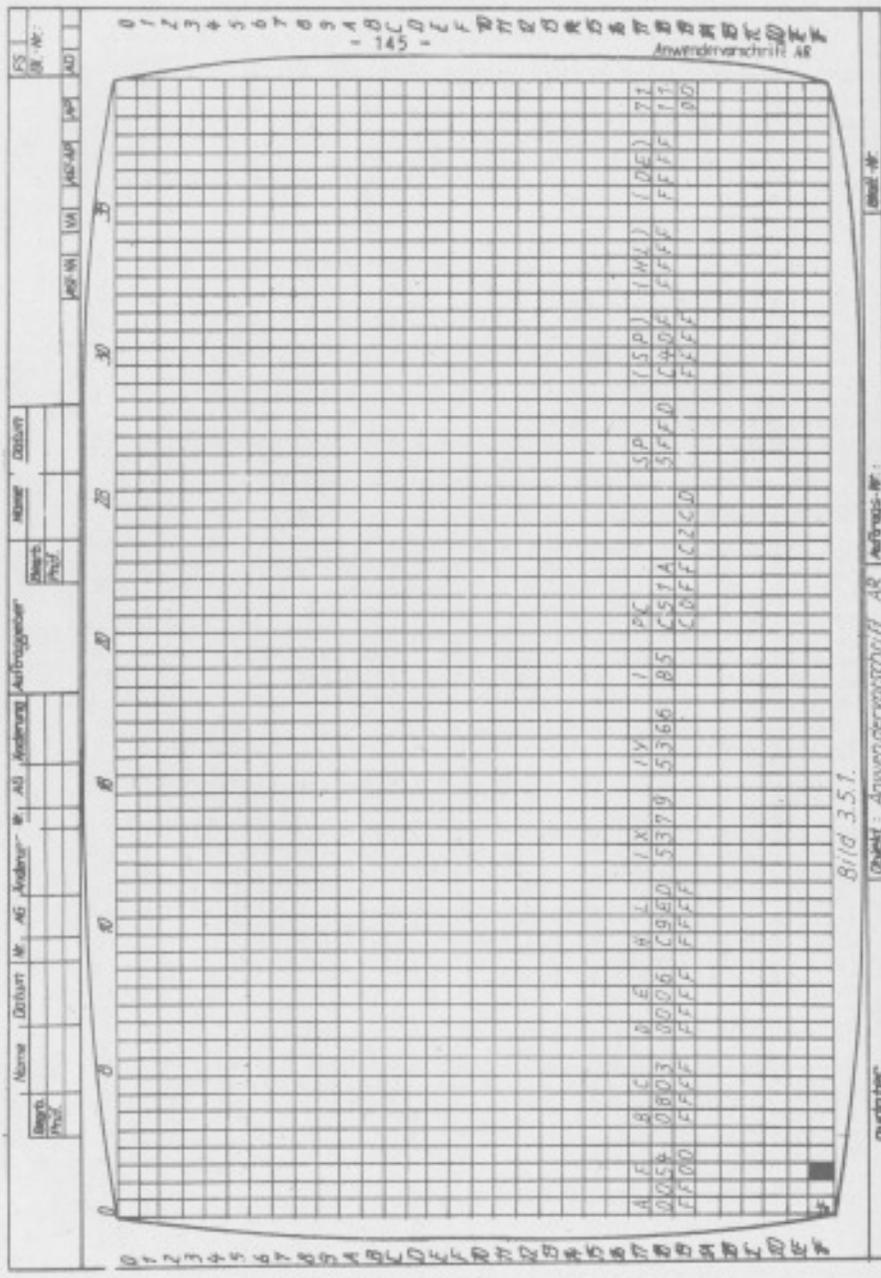


Bild 3.51.

Objekt: Anwendungsvorschritt AR Aufgabe-10.

Quelle: bsc

Bild-Nr.

Name		Datum		Nr.		AG		Anforderung		Auftraggeber		Name		Datum	
Begrü. Prof.												Anw. Prof.			
RST-W		VA		AD-AP		AP		AD		AD		AD		AD	

FS
BR-Nr.

Anwendungsvorschritt AR

Mit Erreichen des Breaks werden alle zur Zeit bearbeiteten Programme unterbrochen und die aktuelle Registerbelegung angezeigt. Der Stand des Taktzyklus wird registriert und bei Fortsetzung wieder eingestellt. Damit bleiben die Zeitverhältnisse erhalten. Während des Breaks koennen weitere Kommandos n.o. eingegeben werden. (Beispiel siehe Bild 3.5.2a)

Anzeigepunkt setzen A

Die Bedingung erfolgt wie beim Unterbrechungspunkt setzen B. Im Unterschied zum Break wird die Programmabarbeitung nicht unterbrochen, sondern es werden bei jedem Durchlauf die Registerbelegungen und wenn durch D generiert die Speicherbelegung angezeigt, die an dieser Stelle zu diesem Zeitpunkt vorhanden sind.

Ein Anzeigepunkt A und ein Unterbrechungspunkt B kann nicht gleichzeitig gesetzt sein.

Mit Eintragung eines neuen A oder B wird der alte gelöscht. (Beispiel siehe Bild 3.5.2a)

Speicher schreiben S

Mit Eingabe der Kommandos S wird die Adresse und Ebene angefordert, ab der der Speicher beschrieben werden soll. Anschliessend wird mit H bzw. " angegeben, ob HEXA oder ASCII geschrieben werden soll. Ist das eingegeben, werden die folgenden 16 Speicherplaetze ausgeschrieben und koennen durch ueberschreiben veraendert werden. Das angezeigte Format ist einzuhalten. Erst mit Quittierung durch ENTER wird die Eingabe uebernommen. (Beispiel Bild 3.5.2a)

Speicheranzeige D

Fuer die Speicheranzeige stehen 5 Zeilen zur Verfuegung, die sich ueber der Registeranzeige befinden.

Mit Eingabe des Kommandos D wird die Zeile (1-5) angefordert, in der die Anzeige erfolgen soll. Anschliessend wird Ebene, Adresse und Anzahl der Werte angefordert. Es koennen nur 16 Werte je Adresse angezeigt werden. Diese Anzeige wird bei jedem Anzeige- oder Unterbrechungspunkt aktualisiert, so dass auch die Belegung von Speicherzellen beim Programmlauf verfolgt werden kann.

(Beispiel Bild 3.5.2b)

Go Fortsetzen G

Mit Kommando G kann ein Programm das durch Tastendruck oder Break B unterbrochen worden ist fortgesetzt werden. Nach Eingabe des Kommandos ist die ausschrift GO ? mit ENTER zu quittieren, sonst erfolgt keine Fortsetzung.

(Beispiel Bild 3.5.2.b)

VAP Abbruch V

Mit dem Kommando V kann die Weiterbearbeitung eines VAP nach Erreichen eines Break verhindert werden. Demzufolge wird die Funktion bei Einschalten des Monitors durch "ERASE INP" nicht abgearbeitet, da hier kein Break vorliegt.

Nach Kommandoingabe wird die Nummer des abzubrechenden VAP zur Bestätigung ausgegeben. Nach Quittierung mit ENTER wird das VAP abgebrochen und die Abarbeitung eines anderen VAP fortgesetzt.

(Beispiel Bild 3.5.2b)

Reset

Achtung mit diesem Kommando kann ein Hardware-RESET des Rechners ausgelöst werden, wenn die Ausschrift RESET N/W mit N oder W bestätigt wird (N löst einen Neuanlauf und W einen Wiederanlauf ueber den USB aus).

Speicher verschieben M

Mit dem Kommando M kann ein Speicherbereich von einer Adresse auf eine andere verschoben werden. Auch die Verschiebung auf eine andere Ebene ist moeglich.

Es erfolgt die Anforderung: von Adresse/Ebene, bis Adresse nach Adresse/Ebene.

(Beispiel Bild 3.5.3.)

Speicher fuellen F

Nach Eingabe des Kommandos F wird die Anfangsadresse/Ebene und die Endadresse sowie ein Byte angefordert, mit dem der Speicherbereich gefuellt werden soll. (Beispiel Bild 3.5.3.)

4. Erstellung und Einbindung von Anwenderprogrammen

4.1. Programmerstellung

Das Anwenderprogramm kann als Assembler- oder BASIC-Programm erstellt werden.
Anwenderprogramme sind als VAP und in Ausnahmefällen als ISR zu gestalten. Dabei sind die Gestaltungshinweise entsprechend Punkt 1 zu beachten.

4.2. Einordnung in die zeitliche Abarbeitung

Die Einordnung in die zeitliche Abarbeitung muss entsprechend Echtzeitanforderungen des Prozesses und der Randbedingungen des AR erfolgen. Die Randbedingungen ergeben sich aus einer Zeitscheibenerganisation mit einem Grundtakt von 1/3 Sekunde. Grundebenenprogramme dürfen als Einzelprogramm diese Zeit nicht überschreiten.

Bei Hinterprogrammen existiert diese Schranke nicht. Die Gesamtrechnzeit der im Grundtakt arbeitenden Programme darf die Zykluszeit ebenfalls nicht überschreiten.
Die Programme sind in verschiedene Prioritätsstufen einzuordnen und die Zykluszeiten sind festzulegen.

4.3. Systemstrukturierung

Zur Systemstrukturierung sind entsprechende Systemfunktionen (Punkt 3.3.) vorhanden.

Bild 4.3.1 zeigt den prinzipiellen Ablauf.

Die Strukturierung umfasst folgende Komplexe:

- ISI-Geräteezuordnung
- Prioritätslisten
- VAP-Steuerlisten
- Anmeldung OFF-Line, Umschaltung OFF -> ON Line
- Prozessbild
- AR-Abbild
- Steuerlisten Speichertest

4.4. Programmtest

Zum Programmtest sind die Standardfunktionen Strukturierung, Systembedienung und Monitor komplex zu nutzen.

4.5. Programmeinbindung

Die Verarbeitungsprogramme können entweder auf EPROM abgelegt werden oder über Kassette eingelesen werden. Nach Einbindung der Verarbeitungsfunktion muss die Einbindung in die Abarbeitung erfolgen (VAP-Steuerlisten, Prioritätslisten, Zeitverwaltung)

In Ausnahmefällen (z.B. für Testzwecke) ist die Arbeit über Lochbandperipherie bzw. Transport über CMOS RAM möglich.

Funktionskomplex	Information	Adresse	Blocklänge	Blockanzahl	Struktur, feste Belegung	Erzeugung			
Speicherbeschleuniger	Kennung: Speicherbeschreibung	RAM : 5000H EPROM : 8000H	3 Byte	1 * 3 Byte	FFH FFH FFH	* Ist * - Grundbelegung EPROM			
Speicherbeschleuniger	RAM-Belegung für schnellen und zyklischen RAM-Test (nur Ebenen) - RAM-Test im Anlauf : siehe 1)	RAM : 5003H EPROM : 8003H	3 Byte	3 * 3 Byte + 8 * 3 Byte	<table border="1" data-bbox="274 577 357 733"> <tr><td>Ebene</td></tr> <tr><td>HWT Anfang</td></tr> <tr><td>HWT Ende + 1</td></tr> </table> Datenebene 1 - 01H Datenebene 2 - 02H Datenebene 3 - 03H Datenebene 4 - 04H Programmbene 2 - 20H Programmbene 3 - 40H Programmbene 4 - 80H + 1 Byte Endeerkennung OFFH	Ebene	HWT Anfang	HWT Ende + 1	Änderung bzw. Erweiterung im RAM → Abzug für neuen EPROM Änderung bzw. Erweiterung im RAM → Abzug für neuen EPROM
Ebene									
HWT Anfang									
HWT Ende + 1									
Strukturkomplex	Strukturkomplex - EPROM (Objekt - EPROM)	RAM : 5025H EPROM : 8025H	3 Byte	1 * 3 Byte	<table border="1" data-bbox="495 577 567 733"> <tr><td>Ebene</td></tr> <tr><td>HWT Anfang</td></tr> <tr><td>HWT Ende + 1</td></tr> </table> 00H FFH 80H oder Endeerkennung 84H ROM + 19 Byte Endeerkennung (OFFH)	Ebene	HWT Anfang	HWT Ende + 1	Änderung bzw. Erweiterung im RAM → Abzug für neuen EPROM Änderung bzw. Erweiterung im RAM → Abzug für neuen EPROM
Ebene									
HWT Anfang									
HWT Ende + 1									
EPROM-Belegung	EPROM - Belegung der Ebenen (Steuersatz)	RAM : 5028H EPROM : 8028H	3 Byte	8 * 3 Byte	<table border="1" data-bbox="591 577 664 733"> <tr><td>Ebene</td></tr> <tr><td>HWT Anfang</td></tr> <tr><td>HWT Ende + 1</td></tr> </table> + 19 Byte Endeerkennung (OFFH)	Ebene	HWT Anfang	HWT Ende + 1	Änderung bzw. Erweiterung im RAM → Abzug für neuen EPROM
Ebene									
HWT Anfang									
HWT Ende + 1									
Prüfsumme für Anwender-EPROM	Prüfsumme für Objekt - EPROM	RAM : 5047H EPROM : 8047H	2 Byte	1 * 2 Byte	- Prüfsumme für Objekt - EPROM - rekursiv ermitteln	EPROM - Programmierung Punkt 3.4.7			
Prüfsumme für Anwender-EPROM	Prüfsummen für Anwender-EPROM entsprechend a.g. Steuersatz	RAM : 5043H EPROM : 8043H	32 Byte je Steuersatz	8 * 32 Byte	Prüfsummen: 2 Byte je 1K EPROM (Reihenfolge: low - high) (32 Byte je Steuersatz - Worte) 2)	EPROM - Programmierung Punkt 3.4.7			

Funktionskomplex	Information	Adresse	Blocklänge	Blockanzahl	Struktur	Erzeugung				
Adressen	Funktionseinheitenadresse der angekoppelten BSE'n	RAM : 5143H EPROM: B143H	1 Byte	4 * 1 Byte	z.B. 80H, 81H ... nicht vorhandene Funktionseinheit:	Strukturierung				
	HACO Gerät	516 DH	1 Byte	—	SGN HACO	Eingabe Hexa				
		516 EM	1 Byte		FF- Kein HACO, Tastencode HACO aktiv	Eingabe Hexa				
		Anwahl Hardcopy	516 FH	1 Byte	—	FF - kein Monitor Tastencode - Monitor aktiv	Eingabe Hexa			
	Anwahl Monitor	516 FH	1 Byte	—	Positionen in aufsteigender Reihenfolge	Eingabe Hexa				
Programmverwaltung	Tabulatorpositionen	RAM : 5151H EPROM: B151H	7 Byte	—						
	Programmfolge- liste 1	5170H / EPROM: (B 170H)	1 Byte	max 30 * 1 Byte	VAP- Nr. + Endeckennung (FFH)	Strukturierung				
	Programmfolge- liste 2	5190H	1 Byte	max 30 * 1 Byte	VAP- Nr. + Endeckennung (FFH)	Strukturierung				
	Programmfolge- liste 3	51B0H	1 Byte	max 30 * 1 Byte	VAP- Nr. + Endeckennung (FFH)	Strukturierung				
	max. VAP-Nr.	51D0H	1 Byte	—	—	Strukturierung				
	VAP- Steuerliste	51D1H	6 Byte	30 * 6 Byte	<table border="1"> <tr><td>Ebene</td></tr> <tr><td>Ebene</td></tr> <tr><td>VAP- Adresse</td></tr> <tr><td>Steuerbyte</td></tr> </table>	Ebene	Ebene	VAP- Adresse	Steuerbyte	Strukturierung
	Ebene									
	Ebene									
	VAP- Adresse									
	Steuerbyte									
Off- LINE VAP	5285H	1 Byte	—	—	—	Strukturierung				
Anmelde liste Off → On-line	5286H	7 Byte	max 9 * 7 Byte	<table border="1"> <tr><td>VAP-Nr.</td></tr> <tr><td>Steuerbyte 1</td></tr> <tr><td>Steuerbyte 2</td></tr> <tr><td>Zähler</td></tr> <tr><td>VEW</td></tr> </table> + Endeckennung (FFH)	VAP-Nr.	Steuerbyte 1	Steuerbyte 2	Zähler	VEW	Strukturierung
VAP-Nr.										
Steuerbyte 1										
Steuerbyte 2										
Zähler										
VEW										

Funktionskomplex	Information	Adresse	Blocklänge	Blockanzahl	Struktur	Erzeugung
E/A-Geräte		52 C6	2 Byte	—	siehe Punkt 3.3.2	Strukturierung
UP-Aufruf mit Ebenenwechsel	Zeiger auf UP-Adreßliste im Grundbereich	52 C9	2 Byte	—	Adresse 2 Byte	Eingabe Hexa
Prozeßabbild AR-Abbild	Ebene PRAB	52 D0	1 Byte	—	Nr. 1-4	Strukturierung
	Adresse PRAB	52 D1	2 Byte	—	2 Byte Adresse	Strukturierung
	Länge PRAB	52 D3	2 Byte	—	Integer 2 Byte	Strukturierung
	Adresse Steuersatz für Anforderungstelegramme	52 D5	2 Byte	—	2 Byte Adresse	Strukturierung
	Länge des Steuersatzes für Telegramme	52 D7	2 Byte	—	Integer 2 Byte	Strukturierung
	Ebene AR-AB	52 D9	1 Byte	—	Nr. 1-4	Strukturierung
	Adresse AR-AB	52 DA	2 Byte	—	2 Byte Adresse	Strukturierung
	Länge AR-AB	52 DC	2 Byte	—	Integer 2 Byte	Strukturierung
	Steuerliste Zugriff zum PRAB	52 DE	7 Byte	4 * 7 Byte	siehe Bild 1.11.1	Strukturierung
	Steuerliste zum AR-AB	52 FA	7 Byte	3 * 7 Byte	siehe Bild 1.11.8	Strukturierung
RST-Adressen	Steuerliste für Anforderungstelegramme PRAB	53 D F	227 Byte	—	1/ 8 * 8 Byte 1/ 12 * 8 Byte 1/ 8 * 8 Byte	Strukturierung
	Adressen für Restart	53 F 2	3 Byte	4 * 3 Byte	4 Springbefehle	Hexa
Ausgleichszellen	2 Zellen für rekursive PS-Bestimmung	53 F E	2 Byte	—	—	rekursive PS-Bestimmung

Funktions- komplex	Information	Adresse	Blocklänge	Blockanzahl	Struktur	Erzeugung
Adresse	eigene Funktions- einheitenadresse	RAM: 5147 H	1 Byte	—	zB.	Eingabe Hexa
HACO	Gerätenummer	516 DH	1 Byte	—	SGN HACO	Eingabe Hexa
	Anwahltaster	516 EH	1 Byte	—	FF-kein HACO Tastencode - HACO aktiv	Eingabe Hexa
Prozessabbild	Zykluszeiten	52 CDH	3 Byte	—	Zahlwort als Anzahl Grundakte	Eingabe Hexa
DÜ	DÜ EIN/AUS	52C8H	1 Byte	—		Eingabe Hexa

- 1) Der erste Block von 3x3 Byte beschreibt in der unter 2) angegebenen Struktur den Bildwiederholungspeicher sowie die konkrete ISI-Belegung. Nicht benutzte Plätze sind mit 00H zu beschreiben.

Beispiel:

Adr. B003H:

00H	\	
FOH	>	Bildwiederholungspeicher
00H	/	
=====		
COH *	\	Koppel-RAM
EOH	>	1. Geräete-ISI
E4H	/	
=====		
COH	\	Koppel-RAM
E8H	>	Kopplungs-ISI
FOH	/	

- 2) Hier sind auch die Prüfsummen für die zwei Anwender EPROM's im Grundbereich einzutragen (D8/DC). Die restlichen 28 Byte dieses Bereiches bleiben frei. Die zum nächsten Steuersatz des Anwender-EPROM-Bereiches gehörenden Prüfsummen sind dann ab Adresse 5063H (bzw. B063H) abzulegen.

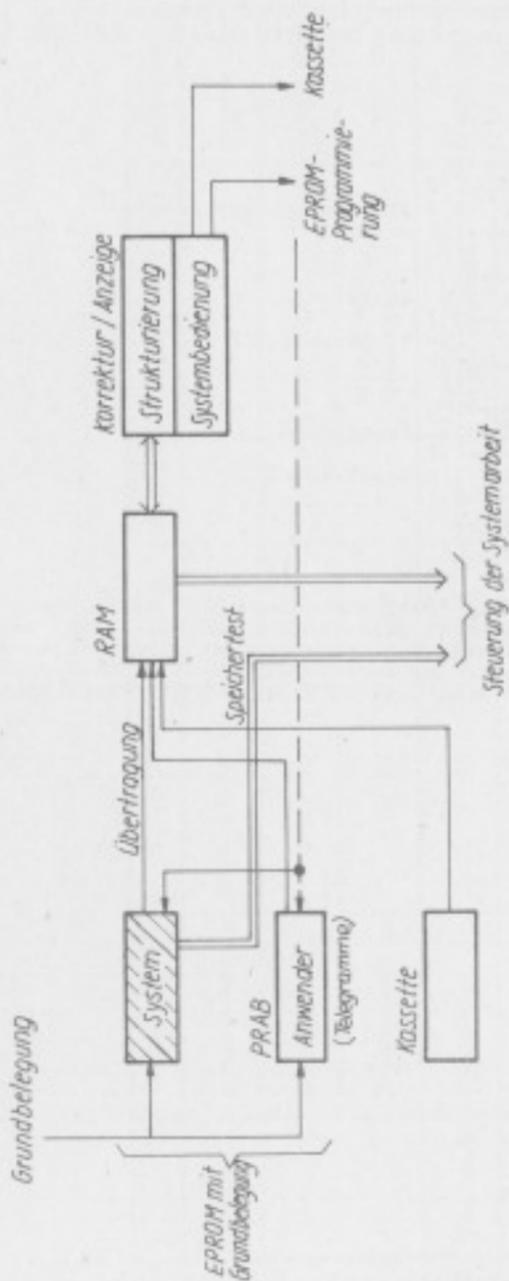


Bild 4.1.: Prinzipieller Ablauf der Korrektur und Sicherung der Strukturierdaten

5. Gleitkommaprogrammpaket5.1. Allgemeine Hinweise

5.1.1. Bestandteile des Gleitkommapaketes FP

Das Gleitkommapaket FP besteht aus folgenden Komponenten:

Arithmetikgrundmodul (im Grundbereich)

Addition	$X:=X+Y$	ADD
Subtraktion	$X:=X-Y$	SUB
Multiplikation	$X:=X*Y$	MULT
Division	$X:=X/Y$	DIV
Quadratbildung	$X:=X**2$	SQU
Kehrwert	$X:=1./X$	RVS
Betrag	$X:=ABS(X)$	ABS
Konvertierung 16 Bit-Zweierkomplement in Gleitkommadarstellung		FLOAT
Konvertierung Gleitkommaformat in 16 Bit- Zweierkomplement (ganzzahlig) rundend abschneidend		ROUND TRUNC

Arithmetikerweiterungsmodul (auf einer Anwenderenebene)

Polynomrechnung	$X:=A(0)+A(1)*Y+.....+A(N)*Y**N$	POLY
Quadratwurzel	$X:=SQRT(X)$	SQRT
Exponentialfunktion	$X:=EXP(X)$	EXP
nat. Logarithmus	$X:=LN(X)$	LN
Cosinus	$X:=COS(X)$	COS
Sinus	$X:=SIN(X)$	SIN
Tangens	$X:=TAN(X)$	TAN
Arkustangens (Hauptwert)	$X:=ARCTAN(X)$	ARCTAN
dek. Logarithmus	$X:=LG(X)$	LG
Potenzfunktion	$X:=X**Y$	POT
Vergleich zweier Gleitkommazahlen stellt Flags wie Befehl CMP		COMP

E/A-Konvertierungsmodul (auf Ebene 1)

Eingabekonvertierung: ISO-7-Bit-Code in freien Format in Gleitkommazahl	GET
Ausgabekonvertierung: Gleitkommazahl in ISO-7-Bit-Code	PUT

5.1.2. Zahlendarstellung

Die Gleitkommazahlen werden als **b i n a e r e** Gleitkommazahlendarstellung:

$$Z = M \cdot 2^{**} E$$

M ist die Binäermantisse, E der Binärexponent. Die Mantisse M hat 24 Binärstellen in Vorzeichenbetragsdarstellung, wobei sich das Komma hinter der höchstwertigen Binärstelle befindet. Es wird festgelegt, dass die Gleitkommazahlen stets normiert sind, d.h. das höchstwertige Mantissenbit ist stets Eins, ausser bei der Zahl Null. Der Zahlenbereich der Mantisse (als Binäerszahl geschrieben) liegt bei:

$$1,00000000000000000000000000000000B \leq /M/ \leq 1,11111111111111111111111111111111B$$

$$1 \leq /M/ < 2$$

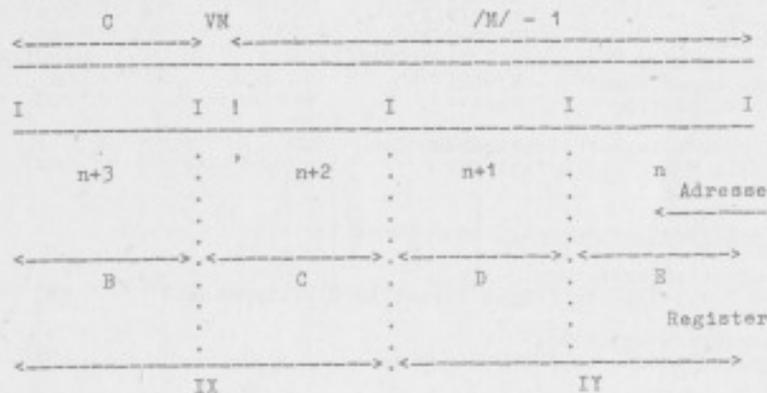
Das höchstwertige Bit wird nicht mit abgespeichert, sondern ist (vereinbarungsgemäss) immer Eins. Man spricht von der "versteckten" Eins. An seine Stelle tritt das Mantissenvorzeichen VM.

In der Gleitkommazahl wird nur der **g e b r o c h e n e** Anteil der Mantisse $/M/ - 1$ abgespeichert.

An Stelle des Binärexponenten E wird die sogenannte Charakteristik C verwendet, die sich aus E nach der Beziehung

$$C = E + 80H$$

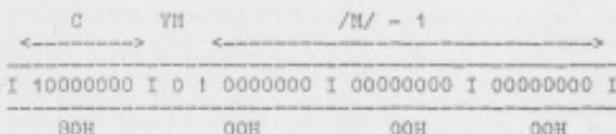
berechnet. Das folgende Schema zeigt den prinzipiellen Aufbau der Gleitkommazahlen sowie ihre Darstellung im Speicher und in den Registern:



VM ist das Mantissenvorzeichen, 0 bedeutet '+', 1 bedeutet '-'.
IX, IY sind die Registeradressen.

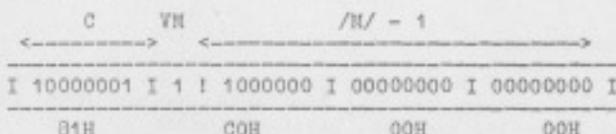
Beispiel 1:

Darstellung von $+1.0 = +1 \cdot 2^{+0}$



Beispiel 2:

Darstellung von $-3 = -1.5 \cdot 2^{+1}$



Fuer die Darstellung der Null wird festgelegt:

Die Zahl Null wird durch die Charakteristik Null bei beliebiger Mantisse dargestellt.

Damit aehnelt die Darstellung der Gleitkommazahlen stark der des SKR, beide Formen koennen leicht ineinander ueberfuehrt werden. Folgender Zahlenbereich ist darstellbar:

$$-3.4028\text{E}+38 \dots -5.8775\text{E}-39, 0, +5.8775\text{E}-39 \dots +3.4028\text{E}+38$$

Der relative Fehler der Zahlendarstellung ist dabei

$$\text{delta} \leq 2^{*(-24)} = 5.96\text{E}-08$$

d.h. die Genauigkeit der Zahlendarstellung betraegt bei Rundung 6 v o l l e Dezimalstellen. Bei der Ausgabe mit 7 Dezimalstellen kann die letzte Stelle um 1 vom wahren Wert abweichen.

Zur Erlaeuterung:

$$\frac{1}{1 + \text{deltamax}} = \frac{1}{1 + 2^{*(-24)}} = 0.9999999404$$

$$= 0.9999999 \quad \text{bei Rundung auf 7 Stellen}$$

$$= 1.00000 \quad \text{bei Rundung auf 6 Stellen}$$

Deshalb wird im Ausgabeprogramm PUT auf 6 Stellen gerundet und mit 6 Dezimalen ausgegeben.

5.1.3. Parameteruebergabe

Alle Komponenten des Gleitkommapakets FP haben standardisierte Schnittstellen:

Der erste Operand bzw. das Argument der Funktion wird in den Registern BCDE uebergeben. Hier steht nach Ausfuehrung der Operation/Funktion auch das Ergebnis. Wenn ein zweiter Operand vorhanden ist, wird er in den Registern IX und IY uebergeben. IX und IY bleiben stets unveraendert. Gewisse Abweichungen zu diesen Festlegungen bestehen beim Polynomprogramm POLY und bei den E/A-Routinen PUT und GET.

5.1.4. Fehlermeldungen

Bei Fehlern wird die Berechnung abgebrochen und das Carry-Flag gesetzt. Zur Kennzeichnung bestimmter Fehler wird zusätzlich das Zero-Flag gesetzt. Dabei gilt i.A.

cy = 1 z = 0 arithmetischer Ueberlauf

cy = 1 z = 1 arithmetischer Unterlauf

Zur genauen Bedeutung der Fehlerflags siehe Beschreibung der einzelnen Funktionen !

Kennzeichen fuer fehlerfreie Programmabarbeitung ist das rueckgesetzte Carry-Flag. Wird eine besondere Fehlerbehandlung gewuenscht, so kann z.B. ein entsprechendes Rahmenprogramm geschrieben werden. Wenn z.B. bei arithmetischem Unterlauf das Ergebnis durch Null ersetzt werden soll, koennte dieses Rahmenprogramm fuer die Addition folgendermassen aussehen:

;ADDITION MIT FEHLERBEHANDLUNG

```

;
ADDF:  CALL ADD           ;AUFRUF ADDITION
      RNC                ;RETURN, WENN KEIN FEHLER
      LD B,0             ;ERGEBNIS=0
      RE                 ;RETURN BEI UNTERLAUF
      POP BC             ;AUFRUPADRESSE IN BC
      JMP UEBERLAUF      ;SPRUNG ZUR FEHLERMELDUNG

```

5.1.5. Adressen Arithmetik

ADD	B490H	\	
SUB	B493H		
MULT	B496H		
DIV	B499H		
SQU	B49CH		
RVS	B49FH	>	Grundmodul
ABS	B4A2H		
FLOAT	B4A5H		
TRUNC	B4A8H		
ROUND	B4ABH	/	
POLY	B4AEH	\	
SQRT	B4B1H		
EXP	B4B4H		
LN	B4B7H		
COS	B4BAH		
SIN	B4BDH	>	Erweiterungsmodul
TAN	B4C0H		
ARCTAN	B4C3H		
LG	B4C6H		
POT	B4C9H		
COMP	B4CCH	/	

Aufruf E/A-Konvertierung

PUT	RST 28H
	DB 3H
GET	RST 28H
	DB 2H

5.1.6. Hinweise zu den Aufwandsangaben

T a k t a n z a h l e n wurden ausgezählt (kein WAIT im M1-Zyklus).

R e c h e n z e i t e n wurden durch Taktzahlen mit CTC (mit WAIT-Generierung im M1-Zyklus) und Multiplikation mit der Standardtaktperiode (407 ns) ermittelt.

Der **S t a c k b e d a r f** bezieht sich auf den von Programm belegten Stack einschliesslich Rueckkehradresse. Der Stackbedarf von aufgerufenen Programmen ist dabei mit enthalten! Da bei Rueckkehr aus dem Arithmetikprogramm der benoetigte Stack wieder freigegeben wird, entspricht der fuer ein Programm vorzusehende Stack dem Maximum des Stackbedarfs der verwendeten Arithmetikprogramme.

5.2. Programmbeschreibungen Arithmetikgrundmodul

5.2.1. Gleitkommaaddition ADD

Kurzbeschreibung:

Addition zweier Gleitkommazahlen $X:=X+Y$

Eintritt: <BCDE> - Summand X
<IXIY> - Summand Y

Austritt: <BCDE> - Summe $X:=X+Y$
<IXIY> - unverändert
<AHL> - unbestimmt

Fehlermeldungen: CY=1 Z=0 arithmet. Ueberlauf
CY=1 Z=1 arithmet. Unterlauf

Takte: 275...650

Rechenzeit: im Mittel 0,25 Millisek.
maximal 0,42 Millisek.

Stack: 6 Byte

5.2.2. Gleitkommasubtraktion SUB

Kurzbeschreibung:

Subtraktion zweier Gleitkommazahlen $X:=X-Y$

Eintritt: <BCDE> - Minuend X
<IXIY> - Subtrahend Y

Austritt: <BCDE> - Differenz $X:=X-Y$
<IXIY> - unverändert
<AHL> - unbestimmt

Fehlermeldungen: CY=1 Z=0 arithmet. Ueberlauf
CY=1 Z=1 arithmet. Unterlauf

Takte: 300...675

Rechenzeit: im Mittel 0,26 Millisek.
maximal 0,43 Millisek.

Stack: 6 Byte

5.2.3. Gleitkommanmultiplikation MULT

Kursbeschreibung:

Multiplikation zweier Gleitkommazahlen $X:=X*Y$ Eintritt: <BCDE> - Faktor X
<IXIY> - Faktor YAustritt: <BCDE> - Produkt $X:=X*Y$
<IXIY> - unverändert
<AHL> - unbestimmtFehlermeldungen: CY=1 Z=0 arithmet. Ueberlauf
CY=1 Z=1 arithmet. UnterlaufTakte: Y ist Potenz von 2: ca. 180
Y ganz, /Y/ < 256 : ca. 800
sonst im Mittel : 1450
maximal : 1650Rechenzeit: Y ganz, /Y/ < 65536 : ca. 0,54 Millisek.
sonst im Mittel : ca. 0,7 Millisek.
maximal : 0,76 Millisek.

Stack: 6 Byte

Hinweis: Rechenseiteinsparung bei Multiplikation mit
ganzen Zahlen, wenn diese in IXIY stehen!

5.2.4. Gleitkommadivision DIV

Kursbeschreibung:

Division zweier Gleitkommazahlen $X:=X/Y$ Eintritt: <BCDE> - Divident X
<IXIY> - Divisor YAustritt: <BCDE> - Quotient $X:=X/Y$
<IXIY> - unverändert
<AHL> - unbestimmtFehlermeldungen: CY=1 Z=0 arithmet. Ueberlauf od. Div.
durch Null
CY=1 Z=1 arithmet. UnterlaufTakte: in Mittel: 1750
maximal : 2200Rechenzeit: in Mittel: 0,96 Millisek.
maximal : 1,15 Millisek.

Stack: 10 Byte

5.2.5. Quadratfunktion SQU

Kurzbeschreibung:

Quadrat einer Gleitkommazahl $X: = X**2$

Eintritt: <BCDE> - Argument X

Austritt: <BCDE> - Quadrat $X: = X**2$
<IXIY> - unverändert
<AHL> - unbestimmtFehlermeldungen: CY=1 Z=0 arithmet. Überlauf
CY=1 Z=1 arithmet. UnterlaufTakte: im Mittel : 1380
maximal : 1580Rechenzeit: im Mittel : 0,68 Millisek.
maximal : 0,74 Millisek.

Stack: 6 Byte

5.2.6. Kehrwert RVS

Kurzbeschreibung:

Kehrwert einer Gleitkommazahl $X: = 1./X$

Eintritt: <BCDE> - Argument X

Austritt: <BCDE> - Kehrwert $X: = 1./X$
<IXIY> - unverändert
<AHL> - unbestimmtFehlermeldungen: CY=1 Z=0 Division durch Null
CY=1 Z=1 arithmet. UnterlaufTakte: im Mittel : 1650
maximal : 2100Rechenzeit: im Mittel : 0,91 Millisek.
maximal : 1,1 Millisek.

Stack: 10 Byte

5.2.7. Betragsfunktion ABS

Kurzbeschreibung:

Betrag einer Gleitkommazahl X:=/X/

Eintritt: <BCDE> - Argument X

Austritt: <BCDE> - Betrag X:=/X/
<AHLIXIY> - unverändert

Fehlermeldungen: keine

Takte: 18

Stack: 2 Byte

5.2.8. Konvertierung 16-Bit-Zweierkomplementdarstellung
in Gleitkommaformat FLOAT

Kurzbeschreibung:

Die ganze Zahl im Register HL wird in eine äquivalente
Gleitkommazahl im Register BCDE umgewandelt. X:=FLOAT(I)

Eintritt: <HL> - 16-Bit-Ganze-Zahl, Zweierkomplement

Austritt: <BCDE> - Zahl im Gleitkommaformat
<HLIXIY> - unverändert
<A> - unbestimmt

Fehlermeldungen: keine

Takte: I > 0: max. 371
I < 0: max. 385

Rechenzeit: ca. 0,19 Millisek.

Stack: 6 Byte

5.2.9. Konvertierung Gleitkommadarstellung in
16-Bit-Zweierkomplementdarstellg. TRUNC
ROUND

Kursbeschreibung:

Umwandlung der Gleitkommazahl im Register BCDE in eine 16-Bit-Zahl in Zweierkomplementdarstellung. Nachkommastellen werden bei TRUNC abgeschnitten, bei ROUND gerundet.

Eintritt: <BCDE> - Gleitkommazahl X

Austritt: <HL> - konvertierte Zahl I:=TRUNC(X) bzw.
I:=ROUND(X)
<BCDEIXIY> - unverändert
<A> - unbestimmt

Fehlermeldungen: CY=1 Zahlenbereichsueberschreitung

Takte: fuer X<1 : 43
ROUND : max. 571
TRUNC : max. 506

Stack: 6 Byte

5.3. Programmbeschreibungen Arithmetikerweiterungsmodul

5.3.1. Polynom Berechnung POLY

Kurzbeschreibung:

Berechnung des Funktionswertes eines Polynoms

$$X := A(0) + A(1)*Y + A(2)*Y**2 + \dots + A(N)*Y**N$$

Eintritt: <C> - Grad N des Polynoms (1 ≤ N ≤ 255)
 ! N=0 ist identisch mit N=255 !
 <DE> - Anfangsadresse der Koeffiziententabelle
 <IXIY> - Argument Y

Austritt: <BCDE> - Funktionswert X
 <IXIY> - unverändert
 <AHL> - unbestimmt

Fehlermeldungen: CY=1 Z=0 arith. Ueberlauf bei ADD/MULT
 CY=1 Z=1 arith. Unterlauf bei ADD/MULT

Rechenzeit: im Mittel: 1.2*N Millisek.

Stack: N*6+10 Byte

Beispiel: Berechnung des Polynoms
 $Y = X**3 + 1.5*X**2 - 2*X + 0.5$ fuer X=3.1415926
 Abs. Anfangsadresse von PP sei 0C000H.

```
LD IX,8149H
LD IY,0FDBH      ;LADEN DES ARGUMENTS NACH IXIY
LD DE,KOTAB      ;ADRESSE DER KOEFFIZIENTENTABELLE
LD C,3           ;POLYOMGRAD
CALL POLY        ;AUF RUF POLYNOMBEBRECHUNG
```

```
POLY: EQU 0C460H ;ABSOLUTE ADRESSE VON POLY
KOTAB: DA 0000H  ;NIEDERWERTIGER TEIL VON A(0)=0.5
      DA 7F00H  ;HOHERWERTIGER      - " -
      DA 0000H  ;NIEDERWERTIGER TEIL VON A(1)=-2.
      DA 8180H  ;HOHERWERTIGER      - " -
      DA 0000H  ;NIEDERWERTIGER TEIL VON A(2)=1.5
      DA 8040H  ;HOHERWERTIGER      - " -
      DA 0000H  ;NIEDERWERTIGER TEIL VON A(3)=1.0
      DA 8000H  ;HOHERWERTIGER      - " -
```

5.3.2. Quadratwurzel SQRT

Kursbeschreibung:

Quadratwurzel aus der Gleitkommazahl X X:=SQRT(X)

Eintritt: <BCDE> - Argument X

Austritt: <BCDE> - Funktionswert X:= SQRT(X)
<IXIY> - unverändert
<AHL> - unbestimmt

Fehlermeldungen: CY=1 negativer Radikand

Takte: X ist Potenz von 2: 3250
 sonst im Mittel: 4250
 maximal: 5400Rechenzeit: im Mittel: 1,94 Millisek.
 maximal: 2,47 Millisek.

Stack: 8 Byte

5.3.3. Exponentialfunktion EXP

Kursbeschreibung:

Berechnung der Exponentialfunktion X:=EXP(X)

Eintritt: <BCDE> - Argument X

Austritt: <BCDE> - Funktionswert X:=EXP(X)
<IXIY> - unverändert
<AHL> - unbestimmtFehlermeldungen: CY=1 Z=0 arithmet. Ueberlauf
 CY=1 Z=1 arithmet. UnterlaufRechenzeit: im Mittel: 5,0 Millisek.
 maximal: 5,3 Millisek.

Stack: 42 Byte

5.3.4. Natuerlicher Logarithmus LN

Kurzbeschreibung:

Berechnung des natuerlichen Logarithmus $X:=LN(X)$

Eintritt: <BCDE> - Argument X

Austritt: <BCDE> - Funktionswert $X:=LN(X)$
<IXIY> - unveraendert
<AHL> - unbestimmtFehlermeldungen: CY=1 Z=0 negatives Argument
CY=1 Z=1 Argument NullRechenzeit: im Mittel: 8,8 Millisek.
maximal: 9,0 Millisek.

Stack: 3 Byte

5.3.5. Cosinusfunktion COS

Kurzbeschreibung:

Berechnung der Cosinusfunktion, Argument im Bogenmass

Eintritt: <BCDE> - Argument X im Bogenmass

Austritt: <BCDE> - Funktionswert $X:=COS(X)$
<IXIY> - unveraendert
<AHL> - unbestimmt

Fehlermeldungen: keine

Rechenzeit: im Mittel: 6,9 Millisek.
maximal: 7,1 Millisek.

Stack: 48 Byte

5.3.6. Sinusfunktion SIN

Kurzbeschreibung:

Berechnung der Sinusfunktion, Argument im Bogenmass. $X:=SIN(X)$

Eintritt: <BCDE> - Argument X

Austritt: <BCDE> - Funktionswert $X:=SIN(X)$
<IXIY> - unveraendert
<AHL> - unbestimmt

Fehlermeldungen: keine

Rechenzeit: im Mittel: 6,7 Millisek.
maximal: 7,0 Millisek.

Stack: 48 Byte

5.3.7. Tangensfunktion TAN

Kurzbeschreibung:

Berechnung der Tangensfunktion, Argument im Bogenmass $X:=\text{TAN}(X)$

Eintritt: <BCDE> - Argument X im Bogenmass

Austritt: <BCDE> - Funktionswert $X:=\text{TAN}(X)$
 <IXIY> - unverändert
 <AHL> - unbestimmt

Fehlermeldungen: CY=1 arith. Ueberlauf (Argument= $(2k-1)*\text{PI}$)

Rechenzeit: im Mittel: 10,4 Millisek.
 maximal: 11,0 Millisek:

Stack: 50 Byte

5.3.8. Arcustangensfunktion ARCTAN

Kurzbeschreibung:

Berechnung des Hauptwertes des Arkustangens, Ergebnis im Bogenmass. $X:=\text{ARCTAN}(X)$

Eintritt: <BCDE> - Argument X

Austritt: <BCDE> - Funktionswert $X:=\text{ARCTAN}(X)$
 <IXIY> - unverändert
 <AHL> - unbestimmt

Fehlermeldungen: keine

Rechenzeit: im Mittel 6,4 Millisek.
 maximal 6,7 Millisek.

Stack: 56 Byte

5.3.9. Dekadischer Logarithmus LG

Kurzbeschreibung:

Berechnung des dekadischen Logarithmus $X:=\text{LG}(X)$

Eintritt: <BCDE> - Argument X

Austritt: <BCDE> - Funktionswert $X:=\text{LG}(X)$
 <IXIY> - unverändert
 <AHL> - unbestimmt

Fehlermeldungen: CY=1 Z=0 Argument negativ
 CY=1 Z=1 Argument Null

Rechenzeit: im Mittel: 9,7 Millisek.
 maximal: 10,0 Millisek.

Stack: 52 Byte

5.3.10. Potenzfunktion POT

Kurzbeschreibung:

Berechnung der Potenzfunktion $X := X^{**}Y$.
 Fuer ganze Y, $|Y| < 256$ kann X negativ sein. Wenn Y nicht
 ganzzahlig ist oder $|Y| > 255$ wird nach der Gleichung
 $X := \text{EXP}(Y \cdot \text{LN}(X))$

gerechnet. Deshalb ist dann der Wertebereich fuer X auf
 positive Werte eingeschraenkt. Die Funktion liefert speziell:

$X^{**}0 = 1$ fuer alle X, also auch
 $0^{**}0 = 1$
 $0^{**}Y = 0$ fuer alle $Y > 0$

Eintritt: <BCDE> - Mantisse X
 <IXIY> - Exponent Y

Austritt: <BCDE> - $X := X^{**}Y$
 <IXIY> - unveraendert
 <AHL> - unbestimmt

Fehlermeldungen: CY=1 Z=0 arithmet. Ueberlauf od. Argument<0
 CY=1 Z=1 arithmet. Unterlauf

Rechenzeit: fuer ganze Y, $|Y| < 256$: im Mittel 5,8 Millisek.
 maximal 11,4 Millisek.
 sonst im Mittel 14,2 Millisek.
 maximal 14,6 Millisek.

Stack: maximal 76 Byte

5.3.11. Gleitkommavergleich COMP

Kurzbeschreibung:

Vergleich der Gleitkommazahlen in <BCDE> und <IXIY>.
 UP wirkt wie Befehl CMP, BCDE entspricht dabei dem Akkumula-
 tor.

Flags:		CY	Z	
<BCDE>	>	<IXIY>	0	0
<BCDE>	=	<IXIY>	0	1
<BCDE>	<	<IXIY>	1	0

Eintritt: <BCDE> - Gleitkommazahl X
 <IXIY> - Gleitkommazahl Y

Austritt: <BCDE> - unveraendert
 <IXIY> - - ' ' -
 <AHL> - unbestimmt
 Flags - gesetzt entspr. Vergleichsergebnis

Fehlermeldungen: keine

Takte: maximal 226

Stack: 4 Byte

5.4. Programmbeschreibungen E/A-Konvertierungsmodul

5.4.1. Eingabekonvertierung GET

Kursbeschreibung:

Konvertierung einer als Zeichenkette im ISO-7-Bit-Code im Speicher abgelegten Gleitkommazahl in die interne Gleitkomma-Darstellung. Fuer die Gleitkommazahl sind alle gebräuchlichen Darstellungsarten zulässig, wobei lediglich zu beachten ist, dass der Dezimalpunkt verwendet und der Exponent durch den Buchstaben E gekennzeichnet wird. Fuer die Zahl 3000 sind u.a. folgende Darstellungsarten zulässig:

3000 +3000 3000. +3000. 3000.0 +3000.0 3.E3 3.E+3
+3.E+3 3.0E3 3.0E+3 3.0E+03 3.E+03 +3.00000000000E+3 usw.

Fuehrende Leerzeichen und Steuerzeichen (z.B. LF, HT u.a.) sind zulässig und werden uebergangen. Jedes Zeichen, das syntaktisch nicht zur Zahl gehoeren kann, wird als Endekennung akzeptiert. Das kann auch der Buchstabe E sein, wenn danach keine Ziffer (bzw. Vorzeichen-Ziffer) folgt. Fehler koennen deshalb nur sein:

1. Zahl nicht darstellbar (zu gross/zu Klein)
2. Zahl leer (keine Ziffer, nur Endekennung)

In die Konvertierung werden 8 gueltige Stellen einbezogen, Vornullen sind zulässig.

Eintritt: <IX> - Zeiger auf erstes Zeichen der dezimalen Gleitkommazahl

Austritt: <IX> - zeigt auf das der Endekennung folgende Byte

<BCDE> - interne Form der Gleitkommazahl

<IY> - unverändert

<AHL> - unbestimmt

Fehlermeldung: CY=1 Konvertierungsfehler (entweder Zahl leer oder Zahl nicht darstellbar)

dann <BCDEAHL> - unbestimmt

<IX> - zeigt auf das dem fehlerhaften Zeichen folgende Zeichen

Rechenzeit: ist stark vom Exponenten abhaengig

im Mittel: ca. 6 Millisek.

maximal: 30 Millisek.

Stack: 10 Byte

5.4.2. Ausgabekontvertierung PUT

Kurzbeschreibung:

Die in den Registern BCDE stehende binaere Gleitkommazahl wird in eine dezimale Gleitkommazahl umgewandelt und im Speicher als ISO-7-Bit Zeichenkette abgelegt. Das Ausgabeformat kann eingestellt werden. Es ist moeglich, die Zahl mit und ohne Exponenten mit der gewuenschten Stellenanzahl auszugeben.

Passt die Zahl nicht in das gewuenschte Ausgabeformat, wird das Feld mit § ausgefuellt. Das Programm hat interne Arbeitzzellen und darf nicht dort eingesetzt werden, wo es unterbrochen werden kann.

Eintritt:	<BCDE>	Gleitkommazahl x
	<IX>	Zeiger auf erstes Zeichen des Ausgabefeldes
	<A>	Format
		Bit 4-7 Stellenanzahl gesamt wenn 0 dann Exponentendarstellung
		Bit 0-2 Stellen nach dem Komma bzw. Stellenzahl bei Exponentendarstellung
Austritt:	<BCDE>	unveraendert
	<IX>	zeigt auf erstes Zeichen hinter dem Ausgabefeld
	<IY>	unversendert
	<AHL>	unbestimmt

Fehlermeldungen: keine

Rechenzeit: im Mittel ca. 7 Millisek.
maximal bis 30 Millisek.

Stack: 10 Byte

VEB Geräte- und Regler-Werke „Wilhelm Pieck“ Teltow

Betrieb des VEB Kombinat Automatisierungsanlagenbau

DDR · 1530 Teltow, Oderstraße 74-76 · Telefon 440 · Telex 015441



Nachdruck bzw. Vervielfältigung ist nur mit Genehmigung des VEB GRW Teltow zulässig. Änderungen im Sinne des technischen Fortschritts vorbehalten.

AUSGABE: Januar 1989