

1. TESTOB, Timer Interrupt Test2. Funktion des Programms

Der Timer Interrupt Test prueft die Interruptfaehigkeit des programmierbaren Schaltkreises 8253A (Zeitgeber PIT) und den programmierbaren Schaltkreis 8259A (Interrupt Controller PIC) auf der ZVE des A 7100. Der Test besteht aus zwei Routinen.

- ROUTINE_01:

Das Maskenregister des PIC wird mit den Bitmustern 55 und AA beschrieben. Es wird geprueft, ob diese Muster wieder lesbar sind. Im Fehlerfalle erfolgt die Ausschrift:

====> PIC MASKREGISTER ERROR.

- ROUTINE_02:

Counter_0 wird programmiert im Modus 0 (Interrupt, wenn Zaehlerstand 0 erreicht ist). Es wird geprueft, ob nach einer bestimmten Zeit ein Interrupt auf der Ebene 2 des PIC anliegt (verursacht durch Counter_0) und ob der PIC daraufhin einen Trap zur Vektoradresse 2 veranlasst. Im Fehlerfalle erfolgen alternativ die Ausschriften:

====> NO INTERRUPT Level 2 occurred

====> INTERRUPT at WRONG VECTOR ADDRESS

====> LATE INTERRUPT Level_2 occurred

3. Voraussetzungen

Im Folgenden werden Speicheradressen in der Form Segment:Offset angegeben. Die Zuordnung von Segmentnamen und absoluten Adressen ist aus der MP2-Liste zu entnehmen.

3.1. Geraeteausruistung

Es ist die Grundausruestung des Rechners A 7100 erforderlich. Die Ein/Ausgabe richtet sich nach der im Monitor generierten Geraete-konfiguration.

3.2. Speicherbedarf

ca. 1 KB fuer TESTOB.OBJ

Die absoluten Adressen der durch den Locator zugewiesenen Speicherbereiche sind anhand der MP2-Liste zu berechnen.

3.3. Test zusaetzlicher Baugruppen

TESTOB erfordert u.a. fehlerfrei arbeitende Speicher (ZPS oder OPS). Insbesondere wird das richtige Zeitverhalten von ZPS oder OPS geprueft.

3.4. Nutzung anderer Programmodule

Der Testmodul TESTOB.OBJ ist mit dem Bibliotheksmodul LIBRAR.OBJ, dem Leitprogrammmodul LACS.OBJ und der Bibliothek LACS.LIB verbunden.

Aus LACS.OBJ werden genutzt:

Anspruenge: - RETURN_PASS
 - RETURN_FAIL
 Variable: - TDERRONLY
 - TDDEBUG
 Routinen: - ASCII_CONV
 - TDMASKEDMESSAGE
 - TDDISPLAY

Aus LIBRAR.OBJ werden genutzt:

Routinen: - SAVE_INTR_MON
 - REST_INTR_MON
 - RESET_59A
 - INIT_59A
 - NMI_DISABLE
 - FILL_WRONGSWI
 - T_O_INT
 - CHECK_ROUT
 - CHECK_ITERA
 - CHECK_ERROR

4. Ladeprozedur fuer autonomen Ablauf

Das Testprogramm wird im Verbund mit dem Leitprogramm LACS von Minidiskette gebootet (Kommandobeispiele siehe Punkt 4.1. - 4.3.).

4.1. Laden mit Angabe des Geraetenamens

.B :Fn:TEST<CR> (n = Laufwerksnr.)
 Nach Erreichen des Monitorbedienzustandes:
 .B :Fn:GOB<CR> (n = Laufwerksnr.)
 Anschliessend verweilt das Programm im Generierhalt.

4.2. Laden ohne Angabe des Geraetenamens

.B TEST<CR>
 Nach Erreichen des Monitorbedienzustandes:
 .B GOB<CR>
 Bei letzterem Kommando werden TEST bzw. GOB auf allen Laufwerken gesucht, beginnend mit FO. Anschliessend verweilt das Programm im Generierhalt.

4.3. Laden durch Kommandoeingabe im ACT (A 7100-Confidence-Test)

Nach Netzeinschalten oder nach Druecken der Taste RESET besteht die verkuerzte Lademoeglichkeit von LACS durch Eingabe des Zeichens "T" innerhalb des PIC - Tests von ACT. Nach Erreichen des Monitorbedienzustandes weiter wie oben.

5. Generierinformationen

Der Bediener hat die Moeglichkeit, die Ausgabe von Durchlauf- und Fehlermeldungen sowie allgemeine Mitteilungen ueber den Zustand des Prueflings durch Eingabe von Generierinformationen zu steuern. (siehe Punkt 5.1.). Ausserdem besteht die Moeglichkeit, zur Fehlereingrenzung, Fehlerverdichtung oder fuer Messzwecke den normalen Ablauf des Testprogrammes zu veraendern (siehe Punkt 5.2.). Dazu muss der Programmablauf unterbrochen werden (z.B. durch CTRL-C). Mittels des Monitorkommandos SW kann dann eine Aenderung der Generierung vorgenommen werden.

5.1. Generierung von Meldungen und Mitteilungen

TESTOB laeuft unter der Regie des Leitprogrammmodules LACS. Deshalb werden Meldungen ueber den Programmablauf durch die Variablen TDERRONLY (Adr. DATA_LACS:0000) und TDDEBUG (Adr. DATA_LACS:0002) gesteuert. (siehe Punkt 5.1. der Beschreibung des Leitprogrammes LACS).

5.2. Generierung spezieller Testablaeufe

Spezielle Testablaeufe werden gesteuert durch die Variablen

-ROUT_TESTOB (Adr. DATA_TESTOB:0000) - Routinesteuerwort

Bit 0 = ROUTINE_01, Bit 1 = ROUTINE_02

-ERR_MAX (Adr. DATA_TESTOB:000A) - Fehlerzaehler

Gestaetet die Vorwahl einer maximalen Fehlerzahl. Das Testprogramm behaelt im Fehlerfalle solange die Regie, bis die vorgewahlte Fehlerzahl erreicht ist. Erst dann erfolgt die Rueckkehr zum Leitprogramm LACS und damit Uebergang in den Bedienzustand des Monitors.

Voraussetzung: TDDEBUG Bit 1 gesetzt, ITERA_OB muss mindestens so gross wie ERR_MAX sein.

-ITERA_OB (Adr. DATA_TESTOB:000C) - Iterationszaehler

Gestaetet die Veraenderung der Anzahl der Durchlaeufe von TESTOB, bevor die Regie an LACS abgegeben wird.

-FIRST_OB (Adr. DATA_TESTOB:000E) - Versionsausgabe

Wenn FIRST_OB=FF, dann Ausgabe der Versionsnummer, sonst nicht. (siehe Punkt 8.1.)

6. Startprozedur6.1. Start des Testprogramms

Das Testprogramm TESTOB laeuft unter der Regie von LACS. Nach dem Laden (siehe Punkt 4.) erfolgt der Start:

- Aus dem Generierhalt durch das Monitorkommando
.G<CR>
- Restart erfolgt durch das Monitorkommando
.G <CODE_LACS>:40<CR>.

6.2. Bedieneraktionen

Der Bediener hat folgende Eingriffsmoeglichkeiten waehrend des Programmablaufs:

- CTRL-S = Unterbrechung der Ausgabe, Fortsetzung mit CTRL-Q.
- CTRL-Q = Fortsetzung einer unterbrochenen Ausgabe.
- CTRL-P = Hardcopy der Bildschirmausgabe auf angeschlossenen Drucker. Erneutes CTRL-P setzt die Hardcopy zurueck.
- CTRL-C = Uebergang in den Bedienzustand des Monitors.

Die Wirkung dieser Kommandos erfolgt erst unmittelbar vor Ausgabe einer Meldung auf den Bildschirm. Der Uebergang in den Bedienzustand des Monitors durch Druucken der Taste BREAK auf der Tastatur ist praktisch nicht moeglich, da die Ebene 1 des PIC waehrend der Abarbeitung von TESTOB fast immer maskiert bleiben muss.

7. Einstellungsmoeglichkeiten

Die Standardeinstellung fuer TESTOB lautet:

- TDERRONLY: 0 ; siehe Punkt 5.1. LACS-Beschreibung
- TDDEBUG: 3 ; " " " " "
- ROUT TESTOB: 3 ; 2 Routinen generiert
- ERR MAX: 1 ; INT 3 nach dem ersten Fehler
- ITERA_OB: 888D ; TESTOB laeuft 888 mal, dann Regie an LACS

(siehe Punkte 5. und 8.2.1. der Programmbeschreibung TESTOB)

8. Programmbeschreibung8.1. Programmablauf

Vor dem 1. Durchlauf erfolgt die Ausgabe der Programmnummern, Modulnamen und der Zeichnungs- und Ausgabennummern aller beteiligten Module. Sie hat fuer TESTOB.OBJ folgendes Aussehen (siehe Punkt 8.2.1.):

```
TESTOB yy-mm-dd 1.56.703012.1/67-aa-56ACnnn
(yy-mm-dd = Erstellungsdatum
aa = Ausgabennummer
56ACnnn = Nr. der Aenderungsmittelung).
```

Anschliessend beginnt die eigentliche Pruefung. Zuerst werden die

testprogrammspezifischen Einstellungen (Routinesteuerwort, Fehlerzaehler und Iterationszaehler) geprueft. Falls in irgendeiner der drei Variablen der Wert 0 gefunden wurde, wird der Monitor-Bedienzustand angesprochen und der Bediener zur Korrektur aufgefordert. Danach kann mit G<CR> weitergestartet werden. Dann werden ROUTINE_01 und ROUTINE_02 abgearbeitet, und zwar so oft wie in ITERA_OB angewiesen ist. Die Abarbeitungszeit von TESTOB betraegt bei Fehlerfreiheit ca. 3 sec. Die Rueckkehr zum Leitprogramm erfolgt im fehlerfreien Fall durch einen

```
JMP RETURN_PASS.
```

Im Fehlerfall informiert das Testprogramm den Bediener durch Fehlerausschriften (beachte Punkt 5. und 9.) und uebergibt die Regie an das Leitprogramm durch einen

```
JMP RETURN_FAIL
```

Abbruch des Testprogramms TESTOB siehe Punkt 10.

8.2. Anschlussbedingungen

8.2.1. Struktur des Quellprogramms

Der Testmodul TESTOB ist folgendermassen aufgebaut:

1. #TITLE(TESTOB Zeichn.nr.-Ausgabenr.-Aend.Mitt.Nr.)
2. NAME TESTOB
3. Anschliessend folgt die Programmbeschreibung nach dem Muster der vorliegenden Beschreibung des Leitprogrammes LACS.
4. Es folgen die EXTRN-Erklaerungen.
5. Es folgt das Datensegment mit der Bezeichnung DATA TESTOB.
6. Es folgt das Codesegment mit der Bezeichnung CODE TESTOB.
7. PUBLIC-Erklaerungen stehen in dem Segment, in dem die betreffenden Symbole vorkommen, ganz am Anfang des Segments.

Das Datensegment DATA TESTOB enthaelt folgende Variablen:

- ROUT TESTOB	Wort	0000	Routinesteuerwort
- ERRORLY	"	0002	Entspricht TDERRONLY aus LACS
- DEBUG	"	0004	" TDDEBUG " "
- ROUT TOPIC	"	0006	Zeigt auf die aktuelle ROUTINE
- ERROR REG	"	0008	Enthaelt die aktuelle Fehlerzahl
- ERR MAX	"	000A	Fehlerzaehler
- ITERA OB	"	000C	Iterationszaehler
- FIRST OB	Byte	000E	Versionsausgabe
- MASK 55 01	"	000F	Zuerst gelesenes Maskenbyte
- MASK AA =01	"	0010	Zuletzt " "
- NOINTER 02	"	0011	Enthaelt 01 wenn kein Interrupt
- RES_INT 02	"	0012	55=guter,77=spaeter,AA=schlechter INT

Anschliessend folgen Texte fuer Fehlerausschriften.

Im Segment CODE TESTOB stehen nach den PUBLIC-Erklaerungen und den Symboldefinitionen folgende Daten:

```
VERSION_TESTOB DB ' TESTOB '
                DB 'yy-mm-dd'
                DB 'Zeichn.Nr.-Ausg.Nr.-Aend.Mitt.Nr.',0AH,0DH,0
                ORG 40H
```

dabei sind: yy-mm-dd Datum der letzten Aenderung
Dieser Text wird nach dem Start des Testmoduls in Abhaengigkeit von TDDEBUG (s.Pkt.5.1.) ausgegeben. Die naechste Zeile nach dem ORG-Befehl enthaelt die Ansprungsstelle des Testmoduls mit der Marke TESTOB.

8.2.2. Stack- und Registernutzung

Vom Testmodul TESTOB wird der Stack des Leitprogrammes genutzt, d.h. beim Start von TESTOB wird kein SS sondern nur DS initialisiert.

8.2.3. Protokollsteuerung

Die Ausgaben sind je nach den Werten, die die Variablen TDERRONLY und TDDEBUG enthalten, abgestuft unterdrueckbar (s.Pkt.5.1.). Es gelten die Richtlinien gemaess Punkt 8.2.3. der LACS-Beschreibung.

8.2.4. Nutzbare TESTOB-Routinen

TESTOB enthaelt keine Routinen, die fuer andere Nutzer verfuegbar sind.

8.2.5. Nutzbarkeit anderer Routinen

Siehe Punkt 3.4.

9. Fehler

9.1. Fehlerausgaben

Jede Fehlerausschrift beginnt mit einem eingerueckten Pfeil ' ==>'.
Das Programm TESTOB gibt im Fehlerfall einen Text aus, welcher das fehlerhafte Testobjekt (8253A PIT oder 8259A PIC) kennzeichnet. Dann wird das Leitprogramm angesprochen, welches durch INT 3 den Monitor-Bedienzustand erreicht (Standardgenerierung). Der Bediener kann mittels Monitorkommandos den Fehler eingrenzen, verdichten oder ausblenden (siehe Punkt 5.). Falls undefiniertes Verhalten oder 'Abstuerze' des Rechners den normalen Programmablauf verhindern oder falls keine Fehlerausschriften generiert sind, kann aus den Variablen MASK_55_01 und MASK_AA_01 (in ROUTINE_01) sowie NOINTER_02 und RES_INT_02 (in ROUTINE_02) Information ueber den Fehler gewonnen werden (s.Punkt 8.2.1.). Aus ROUT_TOPIC ist zu entnehmen, in welcher Routine sich das Testprogramm befindet.

9.2. Fortsetzung nach Fehler

In Abhaengigkeit von ERR MAX (s. Punkt 5.) erfolgt nach einer (Standardgenerierung) oder mehreren Fehlerausschriften der Uebergang zum Leitprogramm. Dieses erhoert seinen Fehlerzaehler und geht zum Monitor. Die Wiederholung des Tests erfolgt durch das Monitorkommando

G<CR>

10. Verschiedenes10.1. Programmabbruchmoeglichkeiten

Der Programmablauf kann durch die BREAK-Taste bedingt unterbrochen werden (siehe Punkt 6.2.). Weiterstart mit G<CR> ist erlaubt. Restart auf Adresse CODE LACS:40 moeglich. Ein geordneter Abbruch kann durch Eingabe von CTRL-C erfolgen, damit erfolgt der Uebergang in den Monitorbedienzustand vor der naechsten Ein/Ausgabe.

10.2. Kontrolle der Programmdurchfuehrung

Die Durchfuehrung des Testprogramms wird durch die folgende LACS-Meldung auf dem Bildschirm bestaetigt (Standardfall):
(Testprogr.Nr.) (Benennung des Testprogr.) "PASSED"
Fehler- und Durchlaufzaehler werden in folgender Form ausgegeben:
ERRORS / CYCLES: xxxx / yyyy

Danach beginnt ein neuer Programmzyklus. Nach Unterbrechung des Programms kann die Durchfuehrungszahl des Testprogramms TESTOB auch aus der Zelle

CONST_LACS:00A6

ermittelt werden (Monitorkommando DW). Die Anzahl der aufgetretenen Fehlermeldungen kann man der Zelle

CONST_LACS:00A4

entnehmen.

TESTOB

Betriebsdokumentation A 7100, Bd.3

1. TESTOC, Software Interrupt Test

2. Funktion des Programms

Der Testmodul TESTOC fuehrt die folgende Funktion aus:
Test der CPU-internen Interruptlogik fuer alle 256 Interruptvektoren. Das Interruptvektorfeld wird mit testspezifischen Interruptvektoren "verfaelscht". Die einzelnen Interrupts werden durch den Befehl INT <n> (n= 0,1,...,OFFH) ausgeloezt. Vorher wurde der entsprechende Interruptvektor mit dem aktuellen Testvektor nochmals ueberschrieben. Im Verlaufe des Tests erfolgt der SOLL-IST Vergleich, wobei der IST-Wert bei fehlerfreiem Ablauf des Interrupts durch die aktuelle Interruptroutine bereitgestellt wird. Zusaetzlich werden der "SINGLE-STEP-INTERRUPT" getestet, indem das TRAP-Flag gesetzt wird. Der "INTERRUPT ON OVERFLOW" wird zusaetzlich durch INTO getestet, wobei der Interrupt durch gesetztes/nicht gesetztes O-Flag erlaubt bzw. verboten wird.

3. Voraussetzungen

In der Beschreibung werden Speicheradressen hexadezimal in der Form Segment:Offset angegeben.

3.1. Gerateausruestung

Es ist die Grundausruestung des Rechners A7100 erforderlich. Die Ein/Ausgabe richtet sich nach der im Monitor generierten Geratekonfiguration.

3.2. Speicherbedarf

2 KB fuer den Testmodul TESTOC und 1,5 KB fuer den Modul KRUTIL. Die absoluten Adressen der durch den Locator zugewiesenen Speicherbereiche sind anhand der MP2-Liste zu berechnen.

3.3. Test zusaetzlicher Baugruppen

Durch den Testmodul werden explizit keine zusaetzlichen Baugruppen getestet.

3.4. Nutzung anderer Programmodule

Der Testmodul TESTOC.OBJ ist mit dem Bibliotheksmodul LIBRAR.OBJ, dem Leitmodul LACS.OBJ, der Bibliothek LACS.LIB und dem Hilfsmodul KRUTIL.OBJC verbunden.

Aus LACS.OBJ werden genutzt:
 Anspruege: -RETURN_PASS
 -RETURN_FAIL
 Routinen: -ASCII_CONV
 -NEWLINE
 -TDMASKEDMESSAGE
 -THREESPACE

Aus KRUTIL.OBJ werden genutzt:
 -SAVE_INTR_UT
 -RESTORE_INTR_UT
 -TIMER_UT
 -INIT_59A_UT
 -RESET_59A_UT
 -FILL_WRONGSWI_UT
 -SWI_INT_UT
 -WRONG_SWI_UT

4. Ladeprozedur

Das Testprogramm TESTOC wird im Verbund mit dem Leitprogramm LACS von Minidiskette gebootet (Kommandobeispiele siehe Pkt.4.1 - 4.3).

4.1. Laden mit Angabe des Geraetenamens

.B :Fn:TEST<CR> (n = Laufwerksnr.)
 Nach Erreichen des Monitorbedienzustandes:
 .B :Fn:GOC<CR> (n = Laufwerksnr.)
 Anschliessend verweilt das Programm im Generierhalt.

4.2. Laden ohne Angabe des Geraetenamens

.B TEST<CR>
 Nach Erreichen des Monitorbedienzustandes:
 .B GOC<CR>
 Bei letzterem Kommando werden TEST bzw. GOC auf allen Laufwerken gesucht, beginnend mit FO.
 Anschliessend verweilt das Programm im Generierhalt.

4.3. Laden durch Kommandoeingabe im ACT (A 7100-Confidence-Test)

Nach Netzeinschalten oder nach Druecken der Taste RESET besteht die verkuerzte Lademoeglichkeit von LACS durch Eingabe des Zeichens "T" innerhalb des PIC - Tests von ACT.

5. Generierinformationen

Der Bediener hat die Möglichkeit, die Ausgabe von Durchlauf- und Fehlermeldungen durch Eingabe von Generierinformationen zu steuern. Dazu muss der Programmablauf unterbrochen werden (z.B. durch CTRL-C), bzw. das Programm muss sich im Generierhalt befinden (s. Pkt. 4.1./ 4.2.). Mittels des Monitorkommandos SW kann dann eine Aenderung der Generierung vorgenommen werden.

5.1. Generierung der Meldungen

TESTOC laeuft unter der Regie des Leitprogrammmodules LACS. Deshalb werden Meldungen ueber den Programmablauf durch die Wort-Variablen TDERRONLY (Adr. DATA_LACS:0000) und TDDEBUG (Adr. DATA_LACS:0002) gesteuert. (s. Pkt. 5.1 der Beschreibung des LACS-Moduls)

6. Startprozedur6.1. Start des Testprogramms

Das Testprogramm TESTOC laeuft unter der Regie von LACS. Nach dem Laden (siehe Pkt. 4.) erfolgt der Start:

- Aus dem Generierhalt durch das Monitorkommando
:G<CR>
- Restart erfolgt durch das Monitorkommando
:G <CODE_LACS>:40<CR>.

6.2. Bedieneraktionen

Der Bediener hat folgende Eingriffsmoeglichkeiten waehrend des Programmablaufs:

- CTRL-S Unterbrechung der Ausgabe, Fortsetzung mit CTRL-Q
- CTRL-Q Fortsetzung einer unterbrochenen Ausgabe
- CTRL-P Hardcopy der Bildschirmausgabe auf angeschlossenen Drucker, erneutes CTRL-P setzt die Hardcopy-Funktion zurueck
- CTRL-C Uebergang in den Bedienzustand des Monitors

Die Wirkung dieser Kommandos erfolgt erst unmittelbar vor einer Ausgabe einer Meldung auf den Bildschirm.

7. Einstellungsmoeglichkeiten

Gemaess Standardeinstellung wird der Test zyklisch wiederholt. Pro Zyklus wird das Testprogramm einmal abgearbeitet.

8. Programmbeschreibung

8.1. Programmablauf

Vor dem 1. Durchlauf erfolgt die Ausgabe der Programmnummern, Modulnamen und der Zeichnungs- und Ausgabennummern aller beteiligten Module. Sie hat fuer TESTOC folgendes Aussehen (siehe Punkt 8.2.1.1):

```
TESTOC yy-mm-dd 1.56.703013.8/67-aa-56ACnnn
mit yy-mm-dd = Erstellungsdatum
aa = Ausgabennummer
nnn = Nr. der Aenderungsmittellung
```

Anschliessend erfolgt der eigentliche Test. Seine Laufzeit betraegt kleiner 1 sec. Waehrend der Testdurchfuehrung koennen entsprechend den Erfordernissen die Generierinformationen umgestellt werden. Die Rueckkehr zum Leitprogrammmodul erfolgt im fehlerfreien Fall durch einen

```
JMP RETURN_PASS
```

Im Fehlerfall wird der Test wie folgt beendet:

1. Ausgabe der Fehlermeldungen:

```
SOFTWARE INTERRUPT LEVEL <XX> H FAILED
mit XX: 0...0FFH
```

oder

```
SOFTWARE INTERRUPT ON OVERFLOW FAILED
```

oder

```
SINGLESTEP INTERRUPT FAILED
```

2. Ruecksprung zum Leitprogrammmodul mit

```
JMP RETURN_FAIL
```

Es erfolgt daraufhin automatisch eine Ausgabe durch die Routine DISPLAY_RESULTS. Die Parameter werden vor dem JMP in folgenden Registern uebergeben:

- BX
- DX: Sollwert
- AX: Istwert
- ES

Zum Abbruch des Tests s. Pkt. 10.

8.2. Anschlussbedingungen

8.2.1. Struktur des Quellprogramms

Der Testmodul TESTOC ist folgendermassen aufgebaut:

1. #TITLE(TESTOC Zeichnungsnr.-Ausgabenr.-Aend.Mitt.Nr.)
2. Programmbeschreibung
3. NAME TESTOC
4. Es folgen die EXTRN- und PUBLIC-Erklarungen.
5. Es folgt das Datenssegment mit der Bezeichnung DATA TESTOC.
6. Es folgt das Codesegment mit der Bezeichnung CODE TESTOC.

Das Datensegment DATA_TESTOC enthaelt die folgenden Variablen:

- TOC_CS Codesegment des Testprogrammes
- TOC_IP Offset der Programmzeile, in deren unmittelbarer Naehе der Fehler geortet wurde

Es folgen Variable zum Retten von Registern:

- TOC_AX Enthaeht i.a. den Ist-Wert
- TOC_BX
- TOC_CX
- TOC_DX Enthaeht i.a. den Soll-Wert
- TOC_SI
- TOC_DI
- TOC_ES

Weitere Variable sind:

- TOC_BAD_CODE Enthaeht den Fehlercode: Niveau des fehlerhaften Interrupts bzw. OF000H bei INTO-Fehler, OF001H bei Fehler bei SINGLE STEP INTERRUPT.
- TOC_VERS_SWITCH Datenbyte, das auf 1 gesetzt die Bildschirmausgabe der Version bewirkt
- TOC_ERROR Textrahmen fuer Fehlermitteilung

Im Segment CODE TESTOC stehen am Anfang folgende Zeilen:

```
VERSION_TESTOC DB '          TESTOC '
                DB 'yy-mm-dd '
                DB 'Zeichn.nr.-Ausg.nr.-Aend.Mitt.Nr.',0AH,ODH,0
                ORG 40H
```

dabei sind: yy-mm-dd Datum der letzten Aenderung

Dieser Text wird nach dem Start des Testmoduls in Abhaengigkeit von TDDEBUG (s.Pkt.5.1) ausgegeben. Die naechste Zeile nach dem ORG-Befehl enthaelt die Anspringstelle des Testmoduls mit der Marke TESTOC.

8.2.2. Stack- und Registernutzung

Vom Testmodul TESTOC wird der Stack des Leitprogrammmoduls genutzt, d.h. beim Start von TESTOC wird kein SS sondern nur DS initialisiert.

8.2.3. Protokollsteuerung

Die Ausgaben sind je nach den Werten, die die Variablen TDERRONLY und TDDEBUG enthalten, abgestuft unterdrueckbar (s.Pkt.5.1.). Es gelten die Richtlinien gemaess Punkt 8.2.3 der LACS-Beschreibung.

8.2.4. Nutzbare TESTOC-Routinen

TESTOC enthaelt keine Routinen, die fuer andere Nutzer verfuegbar sind.

8.2.5. Nutzbarkeit anderer Routinen

Siehe Punkt 3.4.

9. Fehler9.1. Fehlerausgaben

Das Programm TESTOC gibt im Fehlerfall einen Text aus, welcher das fehlerhafte Testobjekt kennzeichnet. Dann wird der Leitprogrammmodul angesprochen, welcher durch INT 3 den Monitor-Bedienzustand erreicht (Standardgenerierung). Vom Leitprogrammmodul werden die Standardfehlermeldungen ausgegeben (s. LACS-Beschreibung). Der Bediener kann mittels Monitorkommandos den Fehler eingrenzen, verdichten oder ausblenden (siehe Punkt 5.).

9.2. Fortsetzung nach Fehler

Die Wiederholung des Tests erfolgt durch das Monitorkommando G<CODE_LACS>:40<CR>

10. Verschiedenes10.1. Programmabbruchmöglichkeiten

Der Programmablauf kann durch die BREAK-Taste unterbrochen werden. Weiterstart mit G<CR> führt zu einem Fehler. Restart auf Adresse <CODE_LACS>:40 möglich. Ein geordneter Abbruch kann durch Eingabe von CTRL-C erfolgen, damit erfolgt der Uebergang in den Monitorbedienzustand vor der naechsten Ein/Ausgabe.

10.2. Kontrolle der Programmdurchfuehrung

Die Durchfuehrung des Testprogramms wird durch die folgende LACS-Meldung auf dem Bildschirm bestaetigt (Standardfall):

```
TESTOC: SOFTWARE INTERRUPT TEST          "PASSED"
Fehler- und Durchlaufzaehler werden in folgender Form ausgegeben:
      ERRORS / CYCLES:  xxxx / yyyy
```

Danach beginnt ein neuer Programmzyklus. Nach Unterbrechung des Programms kann die Durchfuehrungszahl des Testprogramms auch aus der Zelle

```
<CONST_LACS>:0B4
```

ermittelt werden (Monitorkommando DW).

Die Anzahl der aufgetretenen Fehlermeldungen kann man der Zelle

```
<CONST_LACS>:0B2
```

entnehmen.

1. TESTOD, PROM Checksum Test2. Funktion des Programms

Der PROM Checksum Test berechnet die Pruefsumme der vier ZVE PROMs 2764 und vergleicht diese mit der auf den PROMs gespeicherten (Adresse F800:7FFE, Wort). Der Test besteht aus einer Routine.

ROUTINE_01:

Die Berechnung der Pruefsumme erfolgt nach demselben Algorithmus wie im A C T Confidence Test A 7100. Wenn PROMs nicht gelesen werden koennen oder die berechnete Pruefsumme nicht mit der gespeicherten uebereinstimmt, erscheint die Fehlerausschrift:

```
==> PROM CHECKSUM ERROR detected
```

```
Expected eeee Received rrrr Failed Bits bbbbbbbbbbbbbbb
```

Mit eeee = erwarteter Wert in hexadezimaler Form

```
rrrr = empfangener " " " "
```

```
bbbbbbbbbbbbbbbb = fehlerhafte Bits
```

3. Voraussetzungen

Im Folgenden werden Speicheradressen in der Form Segment:Offset angegeben. Die Zuordnung von Segmentnamen und absoluten Adressen ist aus der MP2-Liste zu entnehmen.

3.1. Geraetaausruestung

Es ist die Grundausruestung des Rechners A 7100 erforderlich. Die Ein/Ausgabe richtet sich nach der im Monitor generierten Geraetekonfiguration.

3.2. Speicherbedarf

ca. 1 KB fuer TESTOD.OBJ

Die absoluten Adressen der durch den Locator zugewiesenen Speicherbereiche sind anhand der MP2-Liste zu berechnen.

3.3. Test zusaetzlicher Baugruppen

TESTOD testet keine zusaetzlichen Baugruppen.

3.4. Nutzung anderer Programmmodule

Der Testmodul TESTOD.OBJ ist mit dem Bibliotheksmodul LIBRAR.OBJ, dem Leitprogrammmodul LACS.OBJ und der Bibliothek LACS.LIB verbunden.

Aus LACS.OBJ werden genutzt:
 Anspuenge: - RETURN_PASS
 - RETURN_FAIL
 Variable: - TDERRONLY
 - TDDEBUG
 Routinen: ASCII_CONV
 - TDMASKEDMESSAGE
 - TDDISPLAY
 - TDDISPLAYNUMBER

Aus LIBRAR.OBJ werden genutzt:
 Routinen: - LF_CR_LF
 - CHECK_ROUT
 - CHECK_ITERA
 CHECK_ERROR

4. Ladeprozedur fuer autonomen Ablauf

Das Testprogramm wird im Verbund mit dem Leitprogramm LACS von Minidiskette gebootet (Kommandobeispiele siehe Punkt 4.1. - 4.3.).

4.1. Laden mit Angabe des Geraetenamens

.B :Fn:TEST<CR> (n = Laufwerksnr.)
 Nach Erreichen des Monitorbedienzustandes:
 .B :Fn:GOD<CR> (n = Laufwerksnr.)
 Anschliessend verweilt das Programm im Generierhalt.

4.2. Laden ohne Angabe des Geraetenamens

.B TEST<CR>
 Nach Erreichen des Monitorbedienzustandes:
 .B GOD<CR>
 Bei letzterem Kommando werden TEST bzw. GOD auf allen Laufwerken gesucht, beginnend mit FO.
 Anschliessend verweilt das Programm im Generierhalt.

4.3. Laden durch Kommandoeingabe im ACT (A 7100 - Confidence - Test)

Nach Netzeinschalten oder nach Druecken der Taste RESET besteht die verkuerzte Lademoeglichkeit von LACS durch Eingabe des Zeichens "T" innerhalb des PIC - Tests von ACT.
 Nach Erreichen des Monitorbedienzustandes weiter wie oben.

5. Generierinformationen

Der Bediener hat die Moeglichkeit, die Ausgabe von Durchlauf- und Fehlermeldungen sowie allgemeine Mitteilungen ueber den Zustand des Prueflings durch Eingabe von Generierinformationen zu steuern. (siehe Punkt 5.1.).

Ausserdem besteht die Moeglichkeit, zur Fehlereingrenzung, Fehlerverdichtung oder fuer Messzwecke den normalen Ablauf des Testprogrammes zu veraendern (siehe Punkt 5.2.).

Dazu muss der Programmablauf unterbrochen werden (z.B. durch CTRL-C). Mittels des Monitorkommandos SW kann dann eine Aenderung der Generierung vorgenommen werden.

5.1. Generierung von Meldungen und Mitteilungen

TESTOD laeuft unter der Regie des Leitprogrammmodules LACS. Deshalb werden Meldungen ueber den Programmablauf durch die Variablen TDERRONLY (Adr. DATA_LACS:0000) und TDDEBUG (Adr. DATA_LACS:0002) gesteuert. (siehe Punkt 5.1. der Beschreibung des Leitprogrammes LACS).

5.2. Generierung spezieller Testablaeufe

Spezielle Testablaeufe werden gesteuert durch die Variablen

-ROUT_TESTOD (Adr. DATA_TESTOD:0000) - Routinesteuerwort

Bit 0 = ROUTINE 01

-ERR_MAX (Adr. DATA_TESTOD:000A) - Fehlerzaehler

Gestattet die Vorwahl einer maximalen Fehlerzahl. Das Testprogramm behaelt im Fehlerfalle solange die Regie, bis die vorgewahlte Fehlerzahl erreicht ist. Erst dann erfolgt die Rueckkehr zum Leitprogramm LACS und damit Uebergang in den Bedienzustand des Monitors.

Voraussetzung: TDDEBUG Bit 1 gesetzt, ITERA_OD muss mindestens so gross wie ERR_MAX sein.

-ITERA_OD (Adr. DATA_TESTOD:000C) - Iterationszaehler

Gestattet die Veraenderung der Anzahl der Durchlaeufe von TESTOD, bevor die Regie an LACS abgegeben wird.

-FIRST_OD (Adr. DATA_TESTOD:000E) - Versionsausgabe

Wenn FIRST_OD=FF, dann Ausgabe der Versionsnummer, sonst nicht. (siehe Punkt 8.1.)

6. Startprozedur6.1. Start des Testprogramms

Das Testprogramm TESTOA laeuft unter der Regie von LACS. Nach dem Laden (siehe Punkt 4.) erfolgt der Start:

- Aus dem Generierhalt durch das Monitorkommando

.G<CR>

- Restart erfolgt durch das Monitorkommando

.G <CODE_LACS>:40<CR>.

6.2. Bedieneraktionen

Der Bediener hat folgende Eingriffsmoeglichkeiten waehrend des Programmablaufs:

- CTRL-S = Unterbrechung der Ausgabe, Fortsetzung mit CTRL-Q.
 - CTRL-Q = Fortsetzung einer unterbrochenen Ausgabe.
 - CTRL-P = Hardcopy der Bildschirmausgabe auf angeschlossenen Drucker
Erneutes CTRL-P setzt die Hardcopy zurueck.
 - CTRL-C = Uebergang in den Bedienzustand des Monitors.
- Die Wirkung dieser Kommandos erfolgt erst unmittelbar vor Ausgabe einer Meldung auf den Bildschirm.
Der Uebergang in den Bedienzustand des Monitors durch Druucken der Taste BREAK auf der Tastatur ist moeglich. Dadurch entstehen keine Fehler. Mittels G<CR> kann TESTOD fortgesetzt werden.

7. Einstellungsmoeglichkeiten

Die Standardeinstellung fuer TESTOD lautet:

- TDERRONLY: 0 ; siehe Punkt 5.1. LACS-Beschreibung
- TDDEBUG: 3 ; " " " " "
- ROUT TESTOD: 1 ; 1 Routine generiert
- ERR MAX: 1 ; INT 3 nach dem ersten Fehler
- ITERA_OD: 11D ; TESTOD laeuft 11 mal,
; dann Regie an LACS

8. Programmbeschreibung

8.1. Programmablauf

Vor dem 1. Durchlauf erfolgt die Ausgabe der Programmnummern, Modulnamen und der Zeichnungs- und Ausgabennummern aller beteiligten Module. Sie hat fuer TESTOD folgendes Aussehen (siehe Punkt 8.2.1.):

```
TESTOD yy-mm-dd 1.56.703014.6/67-aa-56ACnnn
      (yy-mm-dd = Erstellungsdatum
      aa       = Ausgabennummer
      56ACnnn = Nr. der Aenderungsmittelung).
```

Anschliessend beginnt die eigentliche Pruefung. Zuerst werden die testprogrammspezifischen Einstellungen (Routinesteuerwort, Fehlerzaehler und Iterationszaehler) geprueft. Falls in irgendeiner der drei Variablen der Wert 0 gefunden wurde, wird der Monitor-Bedienzustand angesprungen und der Bediener zur Korrektur aufgefordert. Danach kann mit G<CR> weitergestartet werden. Dann wird ROUTINE 01 abgearbeitet, und zwar so oft wie in ITERA_OD angewiesen ist. Die Abarbeitungszeit von TESTOD betraegt bei Fehlerfreiheit ca. 3 sec.

Die Rueckkehr zum Leitprogramm erfolgt im fehlerfreien Fall durch einen

```
JMP RETURN_PASS
```

Im Fehlerfall informiert das Testprogramm den Bediener durch Fehlerausschriften (beachte Punkt 5. und 9.) und uebergibt die

Regie an das Leitprogramm durch einen

JMP RETURN_FAIL

Abbruch des Testprogramms TESTOD siehe Punkt 10.

8.2. Anschlussbedingungen

8.2.1. Struktur des Quellprogramms

Der Testmodul TESTOD ist folgendermassen aufgebaut:

1. TITLE(TESTOD Zeichn.nr.-Ausgabenr.-Aend.Mitt.Nr.)
2. NAME TESTOD
3. Anschliessend folgt die Programmbeschreibung nach dem Muster der vorliegenden Beschreibung des Leitprogramms LACS.
4. Es folgen die EXTRN-Erklarungen.
5. Es folgt das Datenssegment mit der Bezeichnung DATA TESTOD.
6. Es folgt das Codesegment mit der Bezeichnung CODE TESTOD.
7. PUBLIC-Erklarungen stehen in dem Segment, in dem die betreffenden Symbole vorkommen, ganz am Anfang des Segments.

Das Datenssegment DATA TESTOD enthaelt folgende Variablen:

- ROUT TESTOD	Wort	0000	Routinesteuerwort
- ERRORONLY	"	0002	Entspricht TDERRORONLY aus LACS
- DEBUG	"	0004	" TDDEBUG " "
- ROUT TOPIC	"	0006	Zeigt auf die aktuelle ROUTINE
- ERROR_REG	"	0008	Enthaelt die aktuelle Fehlerzahl
- ERR MAX	"	000A	Fehlerzaehler
- ITERA_OD	"	000C	Iterationszaehler
- FIRST_OD	Byte	000E	Versionsausgabe
- CHECKSUM_01	Wort	000F	Enth. berechn. Pruefsumme bei Fehler

Anschliessend folgen Texte fuer Fehlerausschriften.

Im Segment CODE TESTOD stehen nach den PUBLIC-Erklarungen und den Symboldefinitionen folgende Daten:

```
VERSION_TESTOD DB 'TESTOD '
                DB 'yy-mm-dd '
                DB 'Zeichn.Nr.-Ausg.nr.-Aend.Mitt.Nr.',
                OAH,ODH,0
                ORG 40H
```

dabei sind:

yy-mm-dd Datum der letzten Aenderung

Dieser Text wird nach dem Start des Testmoduls in Abhaengigkeit von TDDEBUG (s.Pkt.5.1.) ausgegeben. Die naechste Zeile nach dem ORG-Befehl enthaelt die Ansprungstelle des Testmoduls mit der Marke TESTOD.

8.2.2. Stack- und Registernutzung

Vom Testmodul TESTOD wird der Stack des Leitprogrammes genutzt, d.h. beim Start von TESTOD wird kein SS sondern nur DS initialisiert.

8.2.3. Protokollsteuerung

Die Ausgaben sind je nach den Werten, die die Variablen TDERRONLY und TDDEBUG enthalten, abgestuft unterdrueckbar (s.Pkt.5.1.). Es gelten die Richtlinien gemaess Punkt 8.2.3. der LACS-Beschreibung. Ausnahme: Fehlerausschriften erscheinen nur, wenn TDDEBUG ungerade ist.

8.2.4. Nutzbare TESTOD-Routinen

TESTOD enthaelt keine Routinen, die fuer andere Nutzer verfuegbar sind.

8.2.5. Nutzbarkeit anderer Routinen

Siehe Punkt 3.4.

9. Fehler

9.1. Fehlerausgaben

Jede Fehlerausschrift beginnt mit einem eingerueckten Pfeil ' ===> '.

Das Programm TESTOD gibt im Fehlerfall einen Text aus, welcher das fehlerhafte Testobjekt (4 ZVE PROMs 2764 oder ZVE) kennzeichnet. Dann wird das Leitprogramm angesprungen, welches durch INT 3 den Monitor-Bedienzustand erreicht (Standardgenerierung). Der Bediener kann mittels Monitorkommandos den Fehler verdichten (siehe Punkt 5.).

Falls undefiniertes Verhalten oder 'Abstuerze' des Rechners den normalen Programmablauf verhindern oder falls keine Fehlerausschriften generiert sind, kann aus der Variablen CHECKSUM_01 (in ROUTINE_01) die fehlerhafte Pruefsumme gelesen werden (s. Punkt 8.2.1.). Die erwartete Pruefsumme steht als Wort auf der Adresse F800:7FFE. Aus ROUT TOPIC ist zu entnehmen, in welcher Routine sich das Testprogramm befindet.

9.2. Fortsetzung nach Fehler

In Abhaengigkeit von ERR_MAX (s. Punkt 5.) erfolgt nach einer (Standardgenerierung) oder mehreren Fehlerausschriften der Uebergang zum Leitprogramm. Dieses erhoecht seinen Fehlerzaehler und geht zum Monitor. Die Wiederholung des Tests erfolgt durch das Monitorkommando

G<CR>

10. Verschiedenes

10.1. Programmabbruchmoeglichkeiten

Der Programmablauf kann durch die BREAK-Taste unterbrochen werden (siehe Punkt 6.2.). Weiterstart mit G<CR> ist erlaubt. Restart auf Adresse CODE_LACS:40 moeglich. Ein geordneter Abbruch kann durch Eingabe von CTRL-C erfolgen, damit erfolgt der Uebergang in den Monitorbedienzustand vor der naechsten Ein/Ausgabe.

10.2. Kontrolle der Programmdurchfuehrung

Die Durchfuehrung des Testprogramms wird durch die folgende LACS-Meldung auf dem Bildschirm bestaetigt (Standardfall):
(Testprogr.Nr.) (Benennung des Testprogr.) "PASSED"

Fehler- und Durchlaufzaehler werden in folgender Form ausgegeben:
ERRORS / CYCLES: xxxx / yyyy

Danach beginnt ein neuer Programmzyklus.

Nach Unterbrechung des Programms kann die Durchfuehrungszahl des Testprogramms TESTOD auch aus der Zelle

CONST_LACS:00C2

ermittelt werden (Monitorkommando DW).

Die Anzahl der aufgetretenen Fehlermeldungen kann man der Zelle

CONST_LACS:00C0

entnehmen.

TESTOD

Betriebsdokumentation A 7100, Ba.3

1. TESTOE, Failsafe Timer Test2. Funktion des Programms

Der Failsafe Timer Test prueft die Timeout-Logik auf der ZVE. Der Test besteht aus einer Routine.

- ROUTINE_01:

Es wird versucht, die nicht vorhandene Speicherzelle F000:0000 zu lesen. Parallel dazu wird Counter_0 im Modus 0 gestartet (Zeitueberwachung). Im Timeout-Zustand stoppt die CPU die Befehlsabarbeitung, der Zaehler jedoch arbeitet weiter. Wenn Timeout abgelaufen ist, wird der Zaehlerstand gelesen. Er ist direkt proportional zu Timeout, wird umgerechnet in Millisekunden und ausgegeben. Der ermittelte Timeout-Wert wird mit einem unteren (MIN_TIME = 6,666 ms) und einem oberen (MAX_TIME = 14,999 ms) zulaessigen Grenzwert verglichen. Im Fehlerfalle erfolgt die Ausschrift

```
' ==> CPU TIMEOUT is not accurate:    mm,uuu  ms'
```

(Bedeutung mm,uuu ms siehe Punkt 8.2.3.).

3. Voraussetzungen

Im Folgenden werden Speicheradressen in der Form 'Segment:Offset' angegeben. Die Zuordnung von Segmentnamen und absoluten Adressen ist aus der MP2-Liste zu entnehmen.

Es wird vorausgesetzt, das auf der Adresse F000:0000 kein Speicher vorhanden ist (weder RAM noch ROM).

3.1. Geraeteausruetzung

Es ist die Grundausrueftung des Rechners A 7100 erforderlich. Die Ein/Ausgabe richtet sich nach der im Monitor generierten Geraetekonfiguration.

3.2. Speicherbedarf

ca. 1 KB fuer TESTOE.OBJ

Die absoluten Adressen der durch den Locator zugewiesenen Speicherbereiche sind anhand der MP2-Liste zu berechnen.

3.3. Test zusaetzlicher Baugruppen

TESTOE testet keine zusaetzlichen Baugruppen.

3.4. Nutzung anderer Programmmodule

Der Testmodul TESTOE.OBJ ist mit dem Bibliotheksmodul LIBRAR.OBJ, dem Leitprogrammmodul LACS.OBJ und der Bibliothek LACS.LIB verbunden.

Aus LACS.OBJ werden genutzt:
 Anspruege: - RETURN_PASS
 - RETURN_FAIL
 Variable: - TDERRONLY
 - TDDEBUG
 Routinen: - TDMASKEDMESSAGE
 - TDDISPLAY
 - TDDISPLAYNUMBER

Aus LIBRAR.OBJ werden genutzt:
 Routinen: - LF_CR_LF
 - RESET_59A
 - CHECK_ROUT
 - CHECK_ITERA
 - CHECK_ERROR

4. Ladeprozedur fuer autonomen Ablauf

Das Testprogramm wird im Verbund mit dem Leitprogramm LACS von Minidiskette gebootet (Kommandobeispiele siehe Punkt 4.1. - 4.3.).

4.1. Laden mit Angabe des Geraetenamens

␣B :Fn:TEST<CR> (n = Laufwerksnr.)
 Nach Erreichen des Monitorbedienzustandes:
 ␣B :Fn:GOE<CR> (n = Laufwerksnr.)
 Anschliessend verweilt das Programm im Generierhalt.

4.2. Laden ohne Angabe des Geraetenamens

␣B TEST<CR>
 Nach Erreichen des Monitorbedienzustandes:
 ␣B GOE<CR>
 Bei letzterem Kommando werden TEST bzw. GOE auf allen Laufwerken gesucht, beginnend mit FO.
 Anschliessend verweilt das Programm im Generierhalt.

4.3. Laden durch Kommandoeingabe im ACT (A 7100 - Confidence - Test)

Nach Netzeinschalten oder nach Druecken der Taste RESET besteht die verkuerzte Lademoglichkeit von LACS durch Eingabe des Zeichens "T" innerhalb des PIC - Tests von ACT.
 Nach Erreichen des Monitorbedienzustandes weiter wie oben.

5. Generierinformationen

Der Bediener hat die Möglichkeit, die Ausgabe von Durchlauf- und Fehlermeldungen sowie allgemeine Mitteilungen ueber den Zustand des Prueflings durch Eingabe von Generierinformationen zu steuern. (siehe Punkt 5.1.).
 Ausserdem besteht die Moeglichkeit, zur Fehlereingrenzung, Fehlerverdichtung oder fuer Messzwecke den normalen Ablauf des Testprogrammes zu veraendern (siehe Punkt 5.2.).
 Dazu muss der Programmablauf unterbrochen werden (z.B. durch CTRL-C). Mittels des Monitorkommandos SW kann dann eine Aenderung der Generierung vorgenommen werden.

5.1. Generierung von Meldungen und Mitteilungen

TESTOE laeuft unter der Regie des Leitprogrammmodules LACS. Deshalb werden Meldungen ueber den Programmablauf durch die Variablen TDERRONLY (Adr. DATA_LACS:0000) und TDDEBUG (Adr. DATA_LACS:0002) gesteuert. (siehe Punkt 5.1. der Beschreibung des Leitprogrammes LACS).

5.2. Generierung spezieller Testablaufe

Spezielle Testablaufe werden gesteuert durch die Variablen
 -ROUT TESTOE (Adr. DATA_TESTOE:0000) - Routinesteuerwort
 Bit 0 = ROUTINE 01
 -ERR MAX (Adr. DATA_TESTOE:000A) - Fehlerzaehler
 Gestattet die Vorwahl einer maximalen Fehlerzahl. Das Testprogramm behaelt im Fehlerfalle solange die Regie, bis die vorgewahlte Fehlerzahl erreicht ist. Erst dann erfolgt die Rueckkehr zum Leitprogramm LACS und damit Uebergang in den Bedienzustand des Monitors.
 Voraussetzung: TDDEBUG Bit 1 gesetzt, ITERA_OE muss mindestens so gross wie ERR_MAX sein.
 -ITERA OE (Adr. DATA_TESTOE:000C) - Iterationszaehler
 Gestattet die Veraenderung der Anzahl der Durchlaufe von TESTOE, bevor die Regie an LACS abgegeben wird.
 -FIRST OE (Adr. DATA_TESTOE:000E) - Versionsausgabe
 Wenn FIRST OE=FF, dann Ausgabe der Versionsnummer, sonst nicht.
 (siehe Punkt 8.1.)

6. Startprozedur

6.1. Start des Testprogramms

Das Testprogramm TESTOE laeuft unter der Regie von LACS. Nach dem Laden (siehe Punkt 4.) erfolgt der Start:

- Aus dem Generierhalt durch das Monitorkommando
 .G<CR>
- Restart erfolgt durch das Monitorkommando
 .G <CODE_LACS>:40<CR>.

6.2. Bedieneraktionen

Der Bediener hat folgende Eingriffsmoeglichkeiten waehrend des Programmablaufs:

- CTRL-S = Unterbrechung der Ausgabe, Fortsetzung mit CTRL-Q.
- CTRL-Q = Fortsetzung einer unterbrochenen Ausgabe.
- CTRL-P = Hardcopy der Bildschirmausgabe auf angeschlossenen Drucker
Erneutes CTRL-P setzt die Hardcopy zurueck.
- CTRL-C = Uebergang in den Bedienzustand des Monitors.

Die Wirkung dieser Kommandos erfolgt erst unmittelbar vor Ausgabe einer Meldung auf den Bildschirm.

Der Uebergang in den Bedienzustand des Monitors ist auch moeglich durch Druucken der Taste BREAK auf der Tastatur. Damit ist dem Bediener ein Mittel gegeben, einen Fehler im Programmablauf von TESTOE zu erzeugen. Der Fehler entsteht, wenn der Monitor-Bedienzustand mit G<CR> verlassen wird und das unterbrochene Programm wieder angesprungen wird.

7. Einstellungsmoeglichkeiten

Die Standardeinstellung fuer TESTOE lautet:

- TDERRONLY: 0 ; siehe Punkt 5.1. LACS-Beschreibung
- TDDEBUG: 3 ; " " " "
- ROUT TESTOE: 1 ; 1 Routine generiert
- BRR MAX: 1 ; INT 3 nach dem ersten Fehler
- ITERA_OE: 333D ; TESTOE laeuft 333 mal,
; dann Regie an LACS

(siehe Punkte 5. und 8.2.1. der Programmbeschreibung TESTOE)

8. Programmbeschreibung

8.1. Programmablauf

Vor dem 1. Durchlauf erfolgt die Ausgabe der Programmnummern, Modulname und der Zeichnungs- und Ausgabennummern aller beteiligten Module. Sie hat fuer TESTOE folgendes Aussehen (siehe Punkt 8.2.1.):

```
TESTOE yy-mm-dd 1.56.703015.4/67-aa-56ACnnn
      (yy-mm-dd = Erstellungsdatum
      aa       = Ausgabennummer
      56ACnnn = Nr. der Aenderungsmittelung).
```

Anschliessend beginnt die eigentliche Pruefung. Zuerst werden die testprogrammspezifischen Einstellungen (Routinesteuerwort, Fehlerzaehler und Iterationszaehler) geprueft. Falls in irgendeiner der drei Variablen der Wert 0 gefunden wurde, wird der Monitor-Bedienzustand angesprungen und der Bediener zur Korrektur aufgefordert. Danach kann mit G<CR> weitergestartet werden. Dann wird ROUTINE_01 abgearbeitet, und zwar so oft wie in ITERA_OE angewiesen ist. Die Abarbeitungszeit von TESTOE betraegt bei Fehlerfreiheit ca. 3 sec.

Die Rueckkehr zum Leitprogramm erfolgt im fehlerfreien Fall durch einen

```
JMP      RETURN_PASS
```

Im Fehlerfall informiert das Testprogramm den Bediener durch Fehlerausschriften (beachte Punkt 5. und 9.) und uebergibt die Regie an das Leitprogramm durch einen

```
JMP      RETURN_FAIL
```

Abbruch des Testprogramms TESTOE siehe Punkt 10.

8.2. Anschlussbedingungen

8.2.1. Struktur des Quellprogramms

Der Testmodul TESTOE ist folgendermassen aufgebaut:

1. TITLE(TESTOE Zeichn.nr.-Ausgabenr.-Aend.Mitt.Nr.)
2. NAME TESTOE
3. Anschliessend folgt die Programmbeschreibung nach dem Muster der vorliegenden Beschreibung des Leitprogrammes LACS.
4. Es folgen die EXTRN-Erklarungen.
5. Es folgt das Datensegment mit der Bezeichnung DATA TESTOE.
6. Es folgt das Codesegment mit der Bezeichnung CODE TESTOE.
7. PUBLIC-Erklarungen stehen in dem Segment, in dem die betreffenden Symbole vorkommen, ganz am Anfang des Segments.

Das Datensegment DATA TESTOE enthaelt folgende Variablen:

- ROUT TESTOE	Wort	0000	Routinesteuerwort			
- ERRONLY	"	0002	Entspricht TBERRONLY aus LACS			
- DEBUG	"	0004	"	TDDEBUG	"	"
- ROUT TOPIC	"	0006	Zeigt auf die aktuelle ROUTINE			
- ERROR REG	"	0008	Enthaelt die aktuelle Fehlerzahl			
- ERR MAX	"	000A	Fehlerzaehler			
- ITERA OE	"	000C	Iterationszaehler			
- FIRST OE	Byte	000E	Versionsausgabe			
- COUNTER 0 VALUE	Wort	000F	Enhaelt Stand von Counter_0			
- TIME INTEGER	"	0011	"	Timeout (Integer in ms)		
- TIME_REMAINDER	"	0013	"	"	(Remainder	
					in us)	

Anschliessend folgen Texte fuer Fehlerausschriften.

Im Segment CODE TESTOE stehen nach den PUBLIC-Erklarungen und den Symboldefinitionen folgende Daten:

```
VERSION TESTOE DB ' ' TESTOE '
                DB 'yy-mm-dd '
                DB 'Zeichn.Nr.-Ausgabenr.-Aend.Mitt.Nr.',
                  OAH,ODH,C
                ORG 40H
```

dabei sind:

yy-mm-dd Datum der letzten Aenderung

Dieser Text wird nach dem Start des Testmoduls in Abhaengigkeit von TDDEBUG (s.Pkt.5.1.) ausgegeben. Die naechste Zeile nach dem ORG-Befehl enthaelt die Anspringstelle des Testmoduls mit der Marke TESTOE.

8.2.2. Stack- und Registernutzung

Vor Testmodul TESTOE wird der Stack des Leitprogrammes genutzt, d.h. beim Start von TESTOE wird kein SS sondern nur DS initialisiert.

8.2.3. Protokollsteuerung

Die Ausgaben sind je nach den Werten, die die Variablen TDERRONLY und TDDEBUG enthalten, abgestuft unterdrueckbar (s. Pkt. 5.1.). Es gelten die Richtlinien gemaess Punkt 8.2.3. der LACS-Beschreibung. Ausnahmen:

- Fehlerausschriften erscheinen nur, wenn TDDEBUG ungerade ist.
- Die Mitteilung ueber die Grosse von Timeout wird nur unterdrueckt, wenn TDERRONLY = FF ist.

Sonst erscheint die Ausschrift:

Just the CPU TIMEOUT is accurate mm.uuu ms
(mit mm = Integer Timeout in ms, uuu = Remainder Timeout in us).

8.2.4. Nutzbare TESTOE-Routinen

TESTOE enthaelt keine Routinen, die fuer andere Nutzer verfuegbar sind.

8.2.5. Nutzbarkeit anderer Routinen

Siehe Punkt 3.4.

9. Fehler9.1. Fehlerausgaben

Jede Fehlerausschrift beginnt mit einem eingerueckten Pfeil
==>'.
.

Das Programm TESTOE gibt im Fehlerfall einen Text aus, welcher das fehlerhafte Testobjekt (Timeout-Schaltung der ZVE) kennzeichnet. Dann wird das Leitprogramm angesprochen, welches durch INT 3 den Monitor-Bedienzustand erreicht (Standardgenerierung). Der Bediener kann mittels Monitorkommandos den Fehler verdichten (siehe Punkt 5.).

Falls undefiniertes Verhalten oder 'Abstuerze' des Rechners den normalen Programmablauf verhindern oder falls keine Fehlerausschriften generiert sind, kann aus den Variablen COUNTER_0_VALUE, TIME_INTEGER und TIME_REMAINDER (in ROUTINE 01) Information ueber den Fehler gewonnen werden (s. Punkt 8.2.1.). Aus ROUT_TOPIC ist zu entnehmen, in welcher Routine sich das Testprogramm befindet.

9.2. Fortsetzung nach Fehler

In Abhaengigkeit von ERR_MAX (s. Punkt 5.) erfolgt nach einer (Standardgenerierung) oder mehreren Fehlerausschriften der Uebergang zum Leitprogramm. Dieses erhoehrt seinen Fehlerzaehler und geht zum Monitor. Die Wiederholung des Tests erfolgt durch das Monitorkommando

G<CR>

10. Verschiedenes10.1. Programmabbruchmoeglichkeiten

Der Programmablauf kann durch die BREAK-Taste unterbrochen werden (siehe Punkt 6.2.). Weiterstart mit G<CR> fuehrt zu einem Fehler. Restart auf Adresse CODE LACS:40 moeglich. Ein geordneter Abbruch kann durch Eingabe von CTRL-C erfolgen, damit erfolgt der Uebergang in den Monitorbedienzustand vor der naechsten Ein/Ausgabe.

10.2. Kontrolle der Programmdurchfuehrung

Die Durchfuehrung des Testprogramms wird durch die folgende LACS-Meldung auf dem Bildschirm bestaetigt (Standardfall):
(Testprogr.Nr.) (Benennung des Testprogr.) "PASSED"

Fehler- und Durchlaufzaehler werden in folgender Form ausgegeben:

ERRORS / CYCLES: xxxx / yyyy

Danach beginnt ein neuer Programmzyklus.

Nach Unterbrechung des Programms kann die Durchfuehrungszahl des Testprogramms TESTOE auch aus der Zelle

CONST LACS:OODO

ermittelt werden (Monitorkommando DW).

Die Anzahl der aufgetretenen Fehlermeldungen kann man der Zelle

CONST LACS:OOCE

entnehmen.

TESTOE

Betriebsdokumentation A 7100, Bd.3

1. TEST13, Parallel Interface Test2. Funktion des Programms

Der Parallel Interface Test prueft die programmierbaren Schaltkreise 8255 (Parallel Interface PPI) und 8259A (Interrupt Controller PIC) sowie die CENTRONICS - Schnittstelle (X5) auf der ZVE des A 7100 mittels des CENTRONICS - Kurzschlusssteckers.

Der Test besteht aus sechs Routinen.

- ROUTINE_01:

Initialisierung des PPI (PORTB in Ausgaberrichtung) mit anschliessender Pruefung, ob PORTC Bit 1 HIGH ist. Bei Fehler erfolgt die Ausschrift:

====> After INIT PPI(8255) is PC1 LOW instead of HIGH

Dann wird das Bitmuster 55 auf PORTB geschrieben. Es gelangt ueber den Kurzschlussstecker auf PORTA und wird von dort gelesen. Das empfangene Bitmuster muss OF sein. Bei Fehler erfolgt die Ausschrift:

====> DATASTREAM ERROR through CENTRONICS JUMPER detected

PORTB	—>	PORTA:	Expected	Received	Failed Bits
			eeeeeeee	rrrrrrrr	bbbbbbbb
			(binaer)	(binaer)	(binaer)

- ROUTINE_02:

Bitmuster-Test wie in ROUTINE_01: 55 —> 00, bei Datenstrom-Fehler Ausschrift wie in ROUTINE_01.

- ROUTINE_03:

Bitmuster-Test wie in ROUTINE_01: EE —> 0A, bei Datenstrom-Fehler Ausschrift wie in ROUTINE_01.

- ROUTINE_04:

Bitmuster-Test wie in ROUTINE_01: BB —> 05, bei Datenstrom-Fehler Ausschrift wie in ROUTINE_01.

Dann wird geprueft, ob PORTC Bit 1 LOW ist. Bei Fehler erfolgt die Ausschrift:

====> After OUTPUT TO PPI_PORTB is PC1 HIGH instead of LOW

- ROUTINE_05:

Es wird geprueft, ob die Vorderflanke des Strobe auf PORTC Bit 1 zu HIGH macht. Bei Fehler erfolgt die Ausschrift:

====> After LEADING EDGE STROBE is PC1 LOW instead of HIGH

Dann wird geprueft, ob die Rueckflanke des Strobe auf PORTC Bit 1 zu LOW macht. Bei Fehler erfolgt die Ausschrift:

====> After TRAILING EDGE STROBE is PC1 HIGH instead of LOW

- ROUTINE_06:

Ueber PORTC wird Interrupt erlaubt und geprueft, ob PORTC Bit 0 zu HIGH wird. Bei Fehler erfolgt die Ausschrift:

====> After ENABLE INTERRUPT Level_04 is PC0 LOW instead of HIGH

Dann werden Daten (FF) auf PORTB geschrieben und geprueft, ob PORTC Bit 0 zu LOW wird. Bei Fehler erfolgt die Ausschrift:

====> After OUTPUT to PPI_PORTB is PC0 HIGH instead of LOW

Zuletzt wird ueber PORTC ein Strobe gesendet als Begleitsignal fuer die auf PORTB geschriebenen Daten und das Interrupt-Flag der CPU gesetzt. Es wird geprueft, ob der PIC auf Ebene 4 einen Interrupt zur Vektoradresse 36 veranlasst. Bei Fehler erfolgen alternativ die Ausschriften:

```

===> NO INTERRUPT Level_04 occurred but PCO already HIGH
===> NO INTERRUPT Level_04 occurred and PCO is still LOW
===> INTERRUPT at WRONG VECTOR ADDRESS
===> LATE INTERRUPT Level_04 occurred

```

3. Voraussetzungen

Im Folgenden werden Speicheradressen in der Form Segment:Offset angegeben. Die Zuordnung von Segmentnamen und absoluten Adressen ist aus der MP2-Liste zu entnehmen.

3.1. Geraeteausruestung

Es ist die Grundausruestung des Rechners A 7100 erforderlich. Die Ein/Ausgabe richtet sich nach der im Monitor generierten Geraete-konfiguration.

Der CENTRONICS - Kurzschlussstecker ist auf X5 der ZVE zu stecken.

3.2. Speicherbedarf

ca. 2 KB fuer TEST13.OBJ

Die absoluten Adressen der durch den Locator zugewiesenen Speicherbereiche sind anhand der MP2-Liste zu berechnen.

3.3. Test zusaetzlicher Baugruppen

TEST13 testet auch den CENTRONICS - Kurzschlussstecker.

3.4. Nutzung anderer Programmmodule

Der Testmodul TEST13.OBJ ist mit dem Bibliotheksmodul LIBRAR.OBJ, dem Leitprogrammmodul LACS.OBJ und der Bibliothek LACS.LIB verbunden.

Aus LACS.OBJ werden genutzt:

Anspruege: - RETURN PASS
 - RETURN FAIL

Variable: - TDERRONLY
 - TDDEBUG

Routinen: - TDMASKEDMESSAGE
 - TDDISPLAY

Aus LIBRAR.OBJ werden genutzt:

Routinen:

- SAVE INTR MON
- REST_INTR MON
- RESET_59A
- LF CR LF
- FILL_WRONGSWI
- T O INT
- CHECK_ROUT
- CHECK_ITERA
- CHECK_ERROR

4. Ladeprozedur fuer autonomen Ablauf

Das Testprogramm wird im Verbund mit dem Leitprogramm LACS von Minidiskette gebootet (Kommandoispiele siehe Punkt 4.1. - 4.3.).

4.1. Laden mit Angabe des Geraetenamens

.B :Fn:TEST<CR> (n = Laufwerksnr.)
 Nach Erreichen des Monitorbedienzustandes:
 .B :Fn:G13<CR> (n = Laufwerksnr.)
 Anschliessend verweilt das Programm im Generierhalt.

4.2. Laden ohne Angabe des Geraetenamens

.B TEST<CR>
 Nach Erreichen des Monitorbedienzustandes:
 .B G13<CR>
 Bei letzterem Kommando werden TEST bzw. G13 auf allen Laufwerken gesucht, beginnend mit F0.
 Anschliessend verweilt das Programm im Generierhalt.

4.3. Laden durch Kommandoeingabe im ACT (A 7100 - Confidence - Test)

Nach Netzeinschalten oder nach Druicken der Taste RESET besteht die verkuerzte Lademoeglichkeit von LACS durch Eingabe des Zeichens "T" innerhalb des PIC - Tests von ACT.
 Nach Erreichen des Monitorbedienzustandes weiter wie oben.

5. Generierinformationen

Der Bediener hat die Moeglichkeit, die Ausgabe von Durchlauf- und Fehlermeldungen sowie allgemeine Mitteilungen ueber den Zustand des Prueflings durch Eingabe von Generierinformationen zu steuern. (siehe Punkt 5.1.).
 Ausserdem besteht die Moeglichkeit, zur Fehlereingrenzung, Fehlerverdichtung oder fuer Messzwecke den normalen Ablauf des Test-

programmes zu veraendern (siehe Punkt 5.2.).
Dazu muss der Programmablauf unterbrochen werden (z.B. durch CTRL-C). Mittels des Monitorkommandos SW kann dann eine Aenderung der Generierung vorgenommen werden.

5.1. Generierung von Meldungen und Mitteilungen

TEST13 laeuft unter der Regie des Leitprogrammmodules LACS. Deshalb werden Meldungen ueber den Programmablauf durch die Variablen TDERRONLY (Adr. DATA LACS:0000) und TDDEBUG (Adr. DATA LACS:0002) gesteuert. (siehe Punkt 5.1. der Beschreibung des Leitprogrammes LACS).

5.2. Generierung spezieller Testablaeufe

Spezielle Testablaeufe werden gesteuert durch die Variablen
-ROUT_TEST13 (Adr. DATA TEST13:0000) - Routinesteuerwort
Bit 0 = ROUTINE_01, Bit 1 = ROUTINE_02, Bit 2 = ROUTINE_03,
Bit 3 = ROUTINE_04, Bit 4 = ROUTINE_05, Bit 5 = ROUTINE_06.
ROUT_TEST13 darf folgende Werte annehmen: 01,03,07,0F,1F,3F
01,02,04,08,10,20

-ERR_MAX (Adr. DATA TEST13:000A) - Fehlerzaehler
Gestattet die Vorwahl einer maximalen Fehlerzahl. Das Testprogramm behaelt im Fehlerfalle solange die Regie, bis die vorgewaehlte Fehlerzahl erreicht ist. Erst dann erfolgt die Rueckkehr zum Leitprogramm LACS und damit Uebergang in den Bedienzustand des Monitors.
Voraussetzung: TDDEBUG Bit 1 gesetzt, ITERA_13 muss mindestens so gross wie ERR_MAX sein.

-ITERA_13 (Adr. DATA TEST13:000C) - Iterationszaehler
Gestattet die Veraenderung der Anzahl der Durchlaeufe von TEST13, bevor die Regie an LACS abgegeben wird.

-FIRST_13 (Adr. DATA TEST13:000E) - Versionsausgabe
Wenn FIRST_13=FF, dann Ausgabe der Versionsnummer, sonst nicht.
(siehe Punkt 8.1.)

6. Startprozedur

6.1. Start des Testprogramms

Das Testprogramm TEST13 laeuft unter der Regie von LACS. Nach dem Laden (siehe Punkt 4.) erfolgt der Start:

- Aus dem Generierhalt durch das Monitorkommando
.G<CR>
- Restart erfolgt durch das Monitorkommando
.G <CODE_LACS>:40<CR>.

6.2. Bedieneraktionen

Der Bediener hat folgende Eingriffsmoeglichkeiten waehrend des Programmablaufs:

- CTRL-S = Unterbrechung der Ausgabe, Fortsetzung mit CTRL-Q.
- CTRL-Q = Fortsetzung einer unterbrochenen Ausgabe.
- CTRL-P = Hardcopy der Bildschirmausgabe auf angeschlossenen Drucker
Erneutes CTRL-P setzt die Hardcopy zurueck.
- CTRL-C = Uebergang in den Bedienzustand des Monitors.

Die Wirkung dieser Kommandos erfolgt erst unmittelbar vor Ausgabe einer Meldung auf den Bildschirm.

Der Uebergang in den Bedienzustand des Monitors durch Druecken der Taste BREAK auf der Tastatur ist praktisch nicht moeglich, da die Ebene 1 des PIC waehrend der Abarbeitung von TEST13 fast immer maskiert bleiben muss.

7. Einstellungsmoeglichkeiten

Die Standardeinstellung fuer TEST13 lautet:

- TDERRONLY: 0 ; siehe Punkt 5.1. LACS-Beschreibung
- TDDEBUG: 3 ; " " " " "
- ROUT TEST13: 3F ; 6 Routinen generiert
- ERR MAX: 1 ; INT 3 nach dem ersten Fehler
- ITERA_13: 555D ; TEST13 laeuft 555 mal,
; dann Regie an LACS

(siehe Punkte 5. und 8.2.1. der Programmbeschreibung TEST13)

8. Programmbeschreibung

8.1. Programmablauf

Vor dem 1. Durchlauf erfolgt die Ausgabe der Programmnummern, Modulnamen und der Zeichnungs- und Ausgabennummern aller beteiligten Module. Sie hat fuer TEST13 folgendes Aussehen (siehe Punkt 8.2.1.):

```
TEST13 yy-mm-dd 1.56.703020.1/67-aa-56ACnnn
(yy-mm-dd = Erstellungsdatum
aa = Ausgabennummer
56ACnnn = Nr. der Aenderungsmittelung).
```

Anschliessend beginnt die eigentliche Pruefung. Zuerst werden die testprogrammspezifischen Einstellungen (Routinesteuerwort, Fehlerzaehler und Iterationszaehler) geprueft. Falls in irgendeiner der drei Variablen der Wert 0 gefunden wurde, wird der Monitor-Bedienzustand angesprungen und der Bediener zur Korrektur aufgefordert. Danach kann mit G<CR> weitergestartet werden. Dann werden ROUTINE 01 bis ROUTINE 06 abgearbeitet, und zwar so oft wie in ITERA 13 angewiesen ist. Die Abarbeitungszeit von TEST13 traegt bei Fehlerfreiheit ca. 3 sec.

Die Rueckkehr zum Leitprogramm erfolgt im fehlerfreien Fall durch einen

```
JMP RETURN_PASS
```

Im Fehlerfall informiert das Testprogramm den Bediener durch Fehlerausschriften (beachte Punkt 5. und 9.) und uebergibt die Regie an das Leitprogramm durch einen

JMP RETURN_FAIL

Abbruch des Testprogramms TEST13 siehe Punkt 10.

8.2. Anschlussbedingungen

8.2.1. Struktur des Quellprogramms

Der Testmodul TEST13 ist folgendermassen aufgebaut:

1. TITLE(TEST13 Zeichn.nr.-Ausgabenr.-Aend.Mitt.Nr.)
2. NAME TEST13
3. Anschliessend folgt die Programmbeschreibung nach dem Muster der vorliegenden Beschreibung des Leitprogrammes LACS.
4. Es folgen die EXTRN-Erklarungen.
5. Es folgt das Datensegment mit der Bezeichnung DATA TEST13.
6. Es folgt das Codesegment mit der Bezeichnung CODE TEST13.
7. PUBLIC-Erklarungen stehen in dem Segment, in dem die betreffenden Symbole vorkommen, ganz am Anfang des Segments.

Das Datensegment DATA TEST13 enthaelt folgende Variablen:

- ROUT TEST13	Word	0000	Routinesteuerwort
- ERRORONLY	"	0002	Entspricht TDERRORONLY aus LACS
- DEBUG	"	0004	TDDEBUG " "
- ROUT TOPIC	"	0006	Zeigt auf die aktuelle ROUTINE
- ERROR REG	"	0008	Enthaelt die aktuelle Fehlerzahl
- ERR MAX	"	000A	Fehlerzaehler
- ITERA 13	"	000C	Iterationszaehler
- FIRST 13	Byte	000E	Versionsausgabe
- AFTER I 01	"	000F	FF bei Fehler AFTER_I, sonst 0
- DAT_ERR_01	"	0010	Enth. die von PORTA_PPI geles. Daten
- DAT_ERR_02	"	0011	" " " " " " "
- DAT_ERR_03	"	0012	" " " " " " "
- DAT_ERR_04	"	0013	" " " " " " "
- AFTER O 04	"	0014	FF bei Fehler AFTER_O, sonst 0
- AFTER L 05	"	0015	" " " AFTER_L, " "
- AFTER T 05	"	0016	" " " AFTER_T, " "
- AFTER E 06	"	0017	" " " AFTER_E, " "
- AFTER B 06	"	0018	" " " AFTER_B, " "
- NOINT H 06	"	0019	" " " NOINT_H, " "
- NOINT L 06	"	001A	" " " NOINT_L, " "
- RES INT 06	"	001B	55=guter,77=spaeter,AA=schlechter INT

Anschliessend folgen Texte fuer Fehlerausschriften.

Im Segment CODE TEST13 stehen nach den PUBLIC-Erklarungen und den Symboldefinitionen folgende Daten:

```
VERSION_TEST13 DB TEST13
                DB 'yy-mm-dd '
                DB 'Zeichn.Nr.-Ausgabenr.-Aend.Mitt.Nr.',
                OAH,ODH,0
                ORG 40H
```

dabei sind:

yy-mm-dd Datum der letzten Aenderung

Dieser Text wird nach dem Start des Testmoduls in Abhaengigkeit von TDDEBUG (s.Pkt.5.1.) ausgegeben. Die naechste Zeile nach dem ORG-Befehl enthaelt die Ansprungstelle des Testmoduls mit der Marke TEST13.

8.2.2. Stack- und Registernutzung

Vom Testmodul TEST13 wird der Stack des Leitprogrammes genutzt, d.h. beim Start von TEST13 wird kein SS sondern nur DS initialisiert.

8.2.3. Protokollsteuerung

Die Ausgaben sind je nach den Werten, die die Variablen TDERRONLY und TDDEBUG enthalten, abgestuft unterdrueckbar (s.Pkt.5.1.). Es gelten die Richtlinien gemaess Punkt 8.2.3 der LACS-Beschreibung. Ausnahme: Fehlerausschriften erscheinen nur, wenn TDDEBUG ungegerte ist.

8.2.4. Nutzbare TEST13-Routinen

TEST13 enthaelt keine Routinen, die fuer andere Nutzer verfuegbar sind.

8.2.5. Nutzbarkeit anderer Routinen

Siehe Punkt 3.4.

9. Fehler

9.1. Fehlerausgaben

Jede Fehlerausschrift beginnt mit einem eingerueckten Pfeil
'====>'

Das Programm TEST13 gibt im Fehlerfall einen Text aus, welcher das fehlerhafte Testobjekt (8255 PPI, 8259A PIC, ZVE oder CENTRONICS - Kurzschlussstecker) kennzeichnet. Dann wird das Leitprogramm angesprochen, welches durch INT 3 den Monitor-Bedienzustand erreicht (Standardgenerierung). Der Bediener kann mittels Monitorkommandos den Fehler eingrenzen, verdichten oder ausblenden (siehe Punkt 5.).

Falls undefiniertes Verhalten oder 'Abstuerze' des Rechners den normalen Programmablauf verhindern oder falls keine Fehlerausschriften generiert sind, kann aus den Variablen AFTER I 01 und DAT ERR 01 (in ROUTINE 01), DAT ERR 02 (in ROUTINE 02), DAT ERR 03 (in ROUTINE 03), DAT ERR 04 und AFTER O 04 (in ROUTINE 04), AFTER L 05 und AFTER T 05 (in ROUTINE 05) sowie AFTER E 06, AFTER B 06, NOINT H 06, NOINT L 06 und RES_INT 06 (in

ROUTINE 06) Information ueber den Fehler gewonnen werden (s. Punkt 8.2.1.). Aus ROUT_TOPIC ist zu entnehmen, in welcher Routine sich das Testprogramm befindet.

9.2. Fortsetzung nach Fehler

In Abhaengigkeit von ERR_MAX (s. Punkt 5.) erfolgt nach einer (Standardgenerierung) oder mehreren Fehlerausschriften der Uebergang zum Leitprogramm. Dieses erhoert seinen Fehlerzaehler und geht zum Monitor. Die Wiederholung des Tests erfolgt durch das Monitorkommando

G<CR>

10. Verschiedenes

10.1. Programmabbruchmoeglichkeiten

Der Programmablauf kann durch die BREAK-Taste bedingt unterbrochen werden (siehe Punkt 6.2.). Weiterstart mit G<CR> ist erlaubt. Restart auf Adresse CODE LACS:40 moeglich. Ein geordneter Abbruch kann durch Eingabe von CTRL-C erfolgen, damit erfolgt der Uebergang in den Monitorbedienzustand vor der naechsten Ein/Ausgabe.

10.2. Kontrolle der Programmdurchfuehrung

Die Durchfuehrung des Testprogramms wird durch die folgende LACS-Meldung auf dem Bildschirm bestaetigt (Standardfall):
(Testprogr.Nr.) (Benennung des Testprogr.) "PASSED"

Fehler- und Durchlaufzaehler werden in folgender Form ausgegeben:
ERRORS / CYCLES: xxxx / yyyy

Danach beginnt ein neuer Programmzyklus.
Nach Unterbrechung des Programms kann die Durchfuehrungszahl des Testprogramms TEST13 auch aus der Zelle

CONST LACS:0116

ermittelt werden (Monitorkommando DW).

Die Anzahl der aufgetretenen Fehlermeldungen kann man der Zelle

CONST LACS:0114

entnehmen.

1. TEST14, AFS Test

2. Funktion des Programms

Der TEST14 prüft die AFS mit Hilfe von Floppy-Laufwerken und speziellen Testdisketten. Am Anfang des ersten Testdurchlaufes sind Bedienereingaben (mit Bedienerführung) bezüglich der geschlossenen Laufwerktypen vorzunehmen; dabei kann auch die Erstellung der benötigten Testdisketten vorgenommen werden.

3. Voraussetzungen

Im Folgenden werden Speicheradressen in der Form Segment:Offset angegeben. Die Zuordnung von Segmentnamen und absoluten Adressen ist aus der MP2-Liste zu entnehmen.

3.1. Geräteausrüstung

Es ist die Grundausrüstung des Rechners A 7100 (mit mindestens einem Floppy-Laufwerk) erforderlich. Die Ein/Ausgabe richtet sich nach der im Monitor generierten Gerätekongfiguration.

3.2. Speicherbedarf

ca. 17 KB fuer TEST14.OBJ

Die absoluten Adressen werden durch den Locator festgelegt und sind mit Hilfe der MP2-Liste zu ermitteln.

3.3. Test zusätzlicher Baugruppen

TEST14 benötigt einen fehlerfreien KES und fehlerfreie Floppy-Laufwerke.

3.4. Nutzung anderer Programmmodule

Der Testmodul TEST14.OBJ ist mit dem Bibliotheksmodul LIBRAR.OBJ, dem Leitprogrammmodul LACS.OBJ und der Bibliothek LACS.LIB verbunden.

Aus LACS.OBJ werden genutzt:

Routinen: - ASCII CONV
 - TDMASKEDMESSAGE
 - TDDISPLAY
 - TDREADLINE

Aus LIBRAR.OBJ werden genutzt:

Routine: - REST_INTR_MON

4. Ladeprozedur fuer autonomen Ablauf

Das Testprogramm wird im Verbund mit dem Leitprogramm LACS von Minidiskette gebootet. (Kommandobeispiele siehe Punkt 4.1. - 4.3.).

4.1. Laden mit Angabe des Geraetenamens

␣B :Fn:TEST<CR> (n = Laufwerknr.)
 Nach Erreichen des Monitorbedienzustandes:
 ␣B :Fn:G14<CR> (n = Laufwerknr.)
 Anschliessend verweilt das Programm im Generierhalt.

4.2. Laden ohne Angabe des Geraetenamens

␣B TEST<CR>
 Nach Erreichen des Monitorbedienzustandes:
 ␣B G14<CR>
 Bei letzterem Kommando werden TEST bzw. G14 auf allen Laufwerken gesucht, beginnend mit FO. Anschliessend verweilt das Programm im Generierhalt.

4.3. Laden durch Kommandoeingabe im ACT (A7100 - Confidence - Test)

Nach Netzzeinschalten oder nach Druecken der Taste RESET besteht die verkuerzte Lademoeglichkeit von LACS durch Eingabe des Zeichens "T" innerhalb des PIC - Tests von ACT.
 Nach Erreichen des Monitorbedienzustandes weiter wie oben.

5. Generierinformationen

Der Bediener hat die Moeglichkeit, die Ausgabe von Durchlauf- und Fehlermeldungen sowie allgemeine Mitteilungen ueber den Zustand des Prueflings durch Eingabe von Generierinformationen zu steuern. (siehe Punkt 5.1).
 Ausserdem besteht die Moeglichkeit zur Fehlereingrenzung, Fehlerverdicthung oder fuer Messzwecke den normalen Ablauf des Testprogrammes zu veraendern (siehe Punkt 5.2).
 Dazu muss der Programmablauf unterbrochen werden (z.B. durch CTRL-C). Mittels des Monitorkommandos SW bzw. S kann dann eine Aenderung der Generierung vorgenommen werden.

5.1. Generierung von Meldungen und Mitteilungen

TEST14 laeuft unter der Regie des Leitprogrammmodules LACS. Deshalb werden Meldungen ueber den Programmablauf durch die Variablen TDERRONLY (Adr. DATA_LACS:0000) und TDDEBUG (Adr. DATA_LACS:0002) gesteuert. (siehe Punkt 5.1 der Beschreibung des

Leitprogrammes LACS)

5.2. Generierung spezieller Testablaufe

Spezielle Testablaufe werden gesteuert durch die Variablen

- ROUT TEST14 (Adr. DATA TEST14:0) - Routine-Steuerwort
 - Bit 0 = Routine POSITIONIEREN
 - Bit 1 = Routine SEKTOR ID LESEN
 - Bit 2 = Routine FLOPY TO KES
 - Bit 3 = Routine DATEN LESEN
 - Bit 4 = Routine FORMATIEREN
 - Bit 5 = Routine DATEN SCHREIBEN
- DIALOG TEST14 (Adr. DATA TEST14:2) - Steuerbyte
 - FFH = Bedienerdialog im ersten Testdurchlauf (Standard)
 - 0 = kein Dialog, d.h. voreingestellte Laufwerkzuordnung benutzen und Testdisketten zwangsweise generieren.

Standard-Voreinstellung der Laufwerkzuordnung:

- DRIVE 0 <Adr. DATA TEST14:5> = 4DH = 'M' ;5.25", SS, DD
 - DRIVE 1 <Adr. DATA TEST14:6> = 4DH = 'M' ;5.25", SS, DD
 - DRIVE 2 <Adr. DATA TEST14:7> = 2DH = '-' ;kein Test
 - DRIVE 3 <Adr. DATA TEST14:8> = 2DH = '-' ;kein Test
- zulaessige Angaben sind noch:
- 54H = 'T' ;5.25", DS, DD
 - 53H = 'S' ;8", SS, SD
 - 44H = 'D' ;8", SS, DD.

6. Startprozedur6.1. Start des Testprogramms

Das Testprogramm TEST14 laeuft unter der Regie von LACS. Nach dem Laden (siehe Punkt 4.) erfolgt der Start:

- Aus dem Generierhalt durch das Monitorkommando
.G<CR>
- Restart erfolgt durch das Monitorkommando
.G <CODE_LACS>:40<CR>

6.2. Bedieneraktionen

Der Bediener hat folgende Eingriffsmoeglichkeiten waehrend des Programmablaufs:

- CTRL-S = Unterbrechung der Ausgabe, Fortsetzung mit CTRL-Q
 - CTRL-Q = Fortsetzung einer unterbrochenen Ausgabe.
 - CTRL-P = Hardcopy der Bildschirmausgabe auf angeschlossenen Drucker.
Erneutes CTRL-P setzt die Hardcopy zurueck.
 - CTRL-C = Uebergang in den Bedienzustand des Monitors.
- Die Wirkung dieser Kommandos erfolgt erst unmittelbar vor Ausgabe einer Meldung auf den Bildschirm.
Der Uebergang in den Bedienzustand des Monitors durch Druecken der Taste BREAK auf der Tastatur, ist nicht in jedem Testzeitpunkt moeglich.

Im ersten Testdurchlauf sind die angezeigten Laufwerkzuordnungen durch Tastatureingaben vom Bediener zu bestaetigen oder neu festzulegen:

- CR-Taste = angezeigte Laufwerkzuordnung bestaetigen
- Minus-Taste = Laufwerk der angezeigten Nummer vom Test ausschliessen
- M-Taste = 5.25" Floppy, Single Side, DD (80 Cylinders)
- T-Taste = 5.25" Floppy, Double Side, DD (80 Cylinders)
- S-Taste = 8" Floppy, SS, Single Density (Standard)
- D-Taste = 8" Floppy, SS, Double Density

Die Generierung der Testdisketten kann nach der verneinenden Antwort (N-Taste) auf die Frage "TESTDISKETTE(S) AVAILABLE ?" angeordnet werden.

Ausloesung und Kontrolle des 'Media change'-Status durch den Bediener siehe Punkt 8.1

7. Einstellungsmoeglichkeiten

Die Standardeinstellung fuer TEST14 lautet:

- TDERRONLY: 0; siehe Punkt 5.1 LACS-Beschreibung
- TDDEBUG: 3; " " " " "
- ROUT TEST14: FFH; alle Routinen generiert
- DIALOG TEST14: FFH; Bedienerdialog im ersten Testdurchlauf

8. Programmbeschreibung

8.1. Programmablauf

Vor dem 1. Durchlauf erfolgt die Ausgabe der Programmnummern, Modulnamen und der Zeichnungs- und Ausgabennummern aller beteiligten Module.

Sie hat fuer TEST14 folgendes Aussehen (siehe Punkt 8.2.1.):

```
TEST14 yy-mm-dd 1.56.703021.8/67-aa-56ACnnn
```

(yy-mm-dd = Erstellungsdatum
aa = Ausgabennummer
56ACnnn = Nr. der Aenderungsmittelung).

- Danach wird die vom TEST14 benutzte KES-Portadresse auf dem Bildschirm angezeigt (nur im ersten Durchlauf).
- Eine RESET-Kommandofolge zum KES setzt den KES, die AFS und die angeschlossenen Laufwerke zurueck. Im Systemspeicher werden die KES-Kommunikationsbloেকে generiert.
- Die Versionen der installierten Firmware-Komponenten (KES, AFS, AFP) werden komplex ermittelt, dabei wird die Existenz der AFS-Firmware festgestellt. Die Anzeige der Versionen erfolgt nur im ersten Durchlauf.

- Die Firmware-PROM-Inhalte werden mit einem CRC-Verfahren ueberprueft (Kontrollsummentest).
- Zu Registern des AFS-SIO-Schaltkreises erfolgt ein Zugriffstest (SIO-TEST 1).
- Ein spezieller Test mit dem AFS-SIO-Schaltkreis prueft bestimmte Hardware-Komplexe der AFS (SIO-TEST 2).
- In den weiteren Testablauf werden die Floppy-Laufwerke einbezogen. Dazu ist nur im ersten Testdurchlauf die Zuordnung der Laufwerknummer zum angeschlossenen Laufwerktyp mittels Bedienerdialog (mit Bedienerfuehrung) anzugeben; Zwecks Bestueckung der Laufwerke mit Testdisketten erfolgt die Programmfortsetzung erst nach einer Bedienerabfrage; hierbei kann nach verneinender Beantwortung (N-Taste) der Frage "TEST DISKETTE(S) AVAILABLE ?" die Generierung der Testdisketten angeordnet werden.
- Es folgt die Initialisierungs-Routine fuer alle testeinbezogene Laufwerke unter Beruecksichtigung des entsprechenden Laufwerktyps. Im Falle der angeordneten Generierung von Testdisketten schliesst sich an die Initialisierung fuer ein Laufwerk sofort die Generierung der dazugehoerigen Testdiskette an, ehe zur Initialisierung des naechsten Laufwerkes uebergegangen wird. Das Ende dieses Generierprozesses, der nur im ersten Durchlauf erfolgt, wird durch die Ausschrift "GENERATION COMPLETE" angezeigt.
- Die folgenden Testroutinen werden nacheinander fuer ein Laufwerk durchlaufen, danach werden sie fuer das naechste Laufwerk wiederholt. Auf dem Bildschirm wird das aktive Laufwerk und die Testroutine angezeigt.
 - Positionieren,
 - Sektor-ID-Feld-Lesen,
 - Datentransfer vom Floppy zum KES-Datenpuffer.
 - Datentransfer vom Floppy zum Systemspeicher,
 - Formatieren und
 - Datentransfer vom Systemspeicher zum Floppy.
- Der Bediener hat die Moeglichkeit an Minifloppy-Laufwerken, die waehrend dieser Routinen zeitweilig passiv sind, einen Klappenschliessvorgang vorzunehmen und die Meldung des "Media change"-Status (einschliesslich der Laufwerksnummer) zu kontrollieren.

Das Testprogramm wird im fehlerfreien Fall durch einen

RET-FAR-Befehl mit AX = 1

verlassen. Im Fehlerfall informiert das Testprogramm den Bediener mit Fehlerausschriften (beachte Punkt 5.1 und 9.) und das Testprogramm wird durch einen

RET-FAR-Befehl mit AX = 0

verlassen.

Abbruch des Testprogramms TEST14 siehe Punkt 10.

8.2. Anschlussbedingungen8.2.1. Struktur des Quellprogramms

Der Testmodul TEST14 ist folgendermassen aufgebaut:

1. TITLE(TEST14 Zeichnungsnr.-Ausgabenr.-Aend.Mitt.Nr.)
2. Die Programmbeschreibung nach dem Muster der vorliegenden Beschreibung des Leitprogramms LACS.
3. NAME TEST14
4. Es folgen die EXTRN-Erklarungen.
5. Es folgt das Datensegment mit der Bezeichnung DATA TEST14.
6. Es folgt das Codesegment mit der Bezeichnung CODE TEST14.
7. PUBLIC-Erklarungen von Symbolen stehen am Anfang von dem Segment, in dem die betreffenden Symbole definiert sind.

Das Datensegment DATA TEST14 enthaelt u.a. folgende Variablen:

```
- ROUT TEST14      Wort 0000 Routine-Steuerwort
- DIALOG TEST14   Byte 0002 Dialog-Steuerbyte
```

Nach weiteren Merkwerten folgen die Anfangsdaten fuer die Kommunikationsblöcke, der Puffer fuer die aktuellen Kommunikationsblöcke und weitere Puffer mit Initialisier- und Formatierdaten; fuer Sektor-ID-Felddaten, fuer Bildschirmangaben, fuer Tastatureingaben, fuer Datentransfers usw.

Anschliessend folgen Texte fuer Fehlerausschriften und fuer Bedienerhinweise

Im Codesegment CODE TEST14 stehen nach den PUBLIC-Erklarungen und den Symboldefinitionen folgende Daten:

```
VERSION_TEST14  DB      '      TEST14 '
                DB      'yy-mm-dd '
                DB      'Zeich.Nr.-Ausg.Nr.-Aend.Nr.', OAH, ODH, O
                ORG      40H
```

dabei sind:

yy-mm-dd Datum der letzten Aenderung

Die nächste Zeile nach dem ORG-Befehl enthaelt die Ansprungsstelle des Testmoduls mit der Marke TEST14. Das Programm gliedert sich zuerst in den Aufruf von zwei Hauptkomplexen:

VORBEREITUNG und eigentlicher AFS TEST, diese sind in den nachfolgenden Uebersichtsroutinen feiner gegliedert.

8.2.2. Stack- und Registernutzung

Vom Testmodul TEST14 wird der Stack des Leitprogrammes genutzt, d.h. beim Start von TEST14 wird nur DS initialisiert.

8.2.3. Protokollsteuerung

Die Ausgaben sind je nach den Werten, die die Variablen TDERRONLY und TDDEBUG enthalten, abgestuft unterdrueckbar (s.Pkt.5.1.). Es gelten die Richtlinien gemaess Punkt 8.2.3 der LACS-Beschreibung.

8.2.4. Nutzbare TEST14-Routinen

TEST14 enthaelt keine Routinen die fuer andere Nutzer verfuegbar sind.

8.2.5. Nutzbarkeit anderer Routinen

Siehe Punkt 3.4.

9. Fehler

9.1. Fehlerausgaben

Das Programm TEST14 gibt im Falle erkannter Fehler einen Text aus und dazu den Instructionpointer, der auf die fehlerfindende Programmstelle hinweist. I.a. folgen weitere Fehleranalysen mit fehlereingrenzenden Textausgaben bzw. mit Ausgabe der Kommunikationsbloecke, evtl. mit nachfolgender Ausgabe des Statuspuffers (Bedieneranalyse erforderlich).

9.2. Fortsetzung nach Fehler

Nach der Fehlerausgabe erfolgt der Uebergang zum Leitprogramm. Dieses erhoehrt seinen Fehlerzaehler und geht zum Monitor. Die Wiederholung des Tests erfolgt durch das Monitorkommando G<CR>.

10. Verschiedenes

10.1. Programmabbruchmoeglichkeiten

Der Programmablauf kann durch die BREAK-Taste bedingt unterbrochen werden (siehe Punkt 6.3.). Weiterstart mit G<CR> ist erlaubt. Restart ist auf Adresse CODE_LACS:40 moeglich. Ein geordneter Abbruch kann durch Eingabe von CTRL-C erfolgen, damit erfolgt der Uebergang in den Monitorbedienzustand vor der naechsten Ein/Ausgabe.

10.2. Kontrolle der Programmdurchfuehrung

Die Durchfuehrung des Testprogramms wird durch die folgende LACS-Meldung auf dem Bildschirm bestaetigt (Standardfall):

```
TEST14: AFS TEST "PASSED"  
Fehler- und Durchlaufzaehler werden in folgender Form ausgegeben:  
***** ERRORS / CYCLES : xxxx / yyyy
```

Danach beginnt ein neuer Programmzyklus.
Nach Unterbrechung des Programms kann die Durchfuehrungszahl des Testprogramms TEST14 auch aus der Zelle
CONST_LACS:0124
ermittelt werden (Monitorkommando DW).

Die Anzahl der aufgetretenen Fehlermeldungen kann man der Zelle
CONST LACS:0122
entnehmen.

10.3. Testdiskettengenerierung

Wird beim Start von TEST14 keine Testdisketten-Generierung
vorgenommen, weil die verwendeten Testdisketten bereits auf einem
anderen gleichartigen Laufwerk generiert (formatiert) wurden,
dann koennen Fehler entstehen, wenn die Laufwerke nicht zueinan-
der kompatible sind.

1. TEST1A, ASP TIMER TEST2. Funktion des Programms

Der ASP TIMER TEST prueft die Zaehlgenauigkeit des CTC Schaltkreises U 857 und die Interruptlogik der Baugruppe ASP. Die Zeitkanale des CTC werden als Zeitgeber programmiert und loesen jeweils im Abstand von einer Millisekunde einen Interrupt aus. Dieser erfolgt busvektoriert. Durch den Timer Schaltkreis 8253 wird ein Zeitfenster von 52 Millisekunden festgelegt, welches als Zeitbasis fuer die Messung dient. Die vier vorhandenen Zeitkanale werden nacheinander getestet.

3. Voraussetzungen

Im Folgenden werden Speicheradressen in der Form Segment:Offset angegeben. Die Zuordnung von Segmentnamen und absoluten Adressen ist aus der MP2-Liste zu entnehmen.

3.1. Gerateausruestung

Es ist die Grundausruestung des Rechners A 7100 erforderlich. Die Ein/Ausgabe richtet sich nach der im Monitor generierten Geratekonfiguration

3.2. Speicherbedarf

ca. 1 KB fuer TEST1A.OBJ
Die absoluten Adressen der durch den Locator zugewiesenen Speicherbereiche sind anhand der MP2-Liste zu berechnen.

3.3. Test zusaetzlicher Baugruppen

TEST1A prueft zusaetzlich die Zaehlgenauigkeit des Timer Schaltkreises der ZVE. Es wird weiterhin die Interruptlogik fuer die Realisierung eines busvektorierten Interrupts geprueft.

3.4. Nutzung anderer Programmmodule

Der Testmodul TEST1A.OBJ ist mit dem Bibliotheksmodul LIBRAR.OBJ, dem Leitprogrammmodul LACS OBJ und der Bibliothek LACS.LIB verbunden.

Aus LACS:OBJ werden genutzt:
Anspruege: - RETURN FAIL
Routinen: - TDMASKEDMESSAGE
 - NEWLINE
 - THREESPACE

Aus LIBRAR OBJ werden genutzt:
 Routinen: - REST_INTR_MON
 - NMI_DISABLE

4. Ladeprozedur fuer autonomen Ablauf

Das Testprogramm wird im Verbund mit dem Leitprogramm LACS von Minidiskette gebootet (Kommandobeispiele siehe Punkt 4.1. - 4.3.).

4.1. Laden mit Angabe des Geraetenamens

„B :Fn:TEST<CR> (n = Laufwerksnr.)
 Nach Erreichen des Monitorbedienzustandes:
 „B :Fn:G1A<CR> (n = Laufwerksnr.)
 Anschliessend verweilt das Programm im Generierhalt.

4.2. Laden ohne Angabe des Geraetenamens

„B TEST<CR>
 Nach Erreichen des Monitorbedienzustandes:
 „B G1A<CR>
 Bei letzterem Kommando werden TEST bzw. G1A auf allen Laufwerken gesucht, beginnend mit F0. Anschliessend verweilt das Programm im Generierhalt.

4.3. Laden durch Kommandoeingabe im ACT (A 7100-Confidence-Test)

Nach Netzeinschalten oder nach Druecken der Taste RESET besteht die verkuerzte Lademoeglichkeit von LACS durch Eingabe des Zeichens "T" innerhalb des PIC - Tests von ACT. Nach Erreichen des Monitorbedienzustandes weiter wie oben.

5. Generierinformationen

Der Bediener hat die Moeglichkeit, die Ausgabe von Durchlauf- und Fehlermeldungen sowie allgemeine Mitteilungen ueber den Zustand des Prueflings durch Eingabe von Generierinformationen zu steuern. (siehe Punkt 5.1.). Ausserdem besteht die Moeglichkeit, zur Fehlereingrenzung, Fehlerverdichtung oder fuer Messzwecke den normalen Ablauf des Testprogrammes zu veraendern (siehe Punkt 5.2.). Dazu muss der Programmablauf unterbrochen werden (z.B. durch CTRL-C). Mittels des Monitorkommandos SW kann dann eine Aenderung der Generierung vorgenommen werden.

5.1. Generierung von Meldungen und Mitteilungen

TEST1A laeuft unter der Regie des Leitprogrammmodules LACS. Deshalb werden Meldungen ueber den Programmablauf durch die Variablen TDERRONLY (Adr. DATA LACS:0000) und TDDEBUG (Adr. DATA LACS:0002) gesteuert (siehe Punkt 5.1. der Beschreibung des Leitprogrammes LACS).

5.2. Generierung spezieller Testablaeufer

Spezielle Testablaeufer werden gesteuert durch die Variablen -ROUT TEST1A (Adr. DATA TEST1A:0000) - Routinesteuerwort
 Bit 0 = Test Kanal 0, Bit 1 = Test Kanal 1
 Bit 2 = Test Kanal 2, Bit 3 = Test Kanal 3

6. Startprozedur6.1. Start des Testprogramms

Das Testprogramm TEST1A laeuft unter der Regie von LACS. Nach dem Laden(siehe Punkt 4.) erfolgt der Start:

- Aus dem Generierhalt durch das Monitorkommando
 .G<CR>
- Restart erfolgt durch das Monitorkommando
 .G <CODE-LACS>:40<CR>.

6.2. Bedieneraktionen

Der Bediener hat folgende Eingriffsmoeglichkeiten waehrend des Programmablaufs:

- CTRL-S = Unterbrechung der Ausgabe, Fortsetzung mit CTRL-Q
- CTRL-Q = Fortsetzung einer unterbrochenen Ausgabe
- CTRL-P = Hardcopy der Bildschirmausgabe auf angeschlossenen Drucker. Erneutes CTRL-P setzt Hardcopy zurueck.
- CTRL-C = Uebergang in den Bedienzustand des Monitors

Die Wirkung dieser Kommandos erfolgt erst unmittelbar vor Ausgabe einer Meldung auf den Bildschirm. Der Uebergang in den Bedienzustand des Monitors ist nicht durch Druecken der Taste BREAK auf der Tastatur moeglich, da der entsprechende Interrupt waehrend des Tests maskiert ist.

7. Einstellungsmoeglichkeiten

Die Standardeinstellung fuer TEST1A lautet:

- TDERRONLY: 0 ; siehe Punkt 5.1. LACS-Beschreibung
 - TDDEBUG: 3 ; " " " "
 - ROUT TEST1A: F ; 4 Routinen
- (siehe Punkt 5. der Programmbeschreibung TEST1A)

8. Programmbeschreibung8.1. Programmablauf

Vor dem 1. Durchlauf erfolgt die Ausgabe der Programmnummern, Modulnamen und der Zeichnungs- und Ausgabennummern aller beteiligten Module. Sie hat fuer TEST1A folgendes Aussehen (siehe Punkt 8.2.1.):

```
TEST1A yy-mm-dd 1.56.703027.5/67-aa-56ACnnn
        (yy-mm-dd = Erstellungsdatum
          aa       = Ausgabennummer
          56ACnnn = Nr. der Aenderungsmittlung).
```

Anschliessend beginnt die eigentliche Pruefung. Die Tests der einzelnen Zeitkanale werden in der Reihenfolge Kanal 0 —> Kanal 4 abgearbeitet. Im fehlerfreien Fall betraegt die Laufzeit ca. 0,5 s und die Rueckkehr zum Leitprogramm erfolgt durch einen Ruecksprung (RET) mit der Uebergabe des Wertes 1 im Register AX. Im Fehlerfall informiert das Testprogramm den Bediener durch Fehlerausschriften (beachte Punkt 5. und 9.) und uebergibt die Regie an das Leitprogramm durch einen

```
JMP      RETURN FAIL.
```

Abbruch des Testprogramms TEST1A siehe Punkt 10.

8.2. Anschlussbedingungen8.2.1. Struktur des Quellprogramms

Der Testmodul TEST1A ist folgendermassen aufgebaut:

1. TITLE(TEST1A Zeichn.Nr.-Ausgabenr.-Aend.Mitt.Nr.)
2. NAME TEST1A
3. Anschliessend folgt die Programmbeschreibung nach dem Muster der vorliegenden Beschreibung des Leitprogrammes LACS.
4. Es folgen die EXTRN-Erklarungen.
5. Es folgen die Vereinbarungen symbolischer Adressen.
6. Es folgt das Datensegment mit der Bezeichnung DATA TEST1A.
7. Es folgt das Codesegment mit der Bezeichnung CODE TEST1A.
8. PUBLIC-Erklarungen stehen in dem Segment, in dem die betreffenden Symbole vorkommen, ganz am Anfang des Segments.

Das Datensegment DATA TEST1A enthaelt folgende wichtige

Wortvariablen :

- ROUW TEST1A	(DATA TEST1A:0000)	Routinesteuerwort
- ERR TEST1A OFF	(DATA TEST1A:0002)	Fehlerzelle Offset
- ERR TEST1A_SEG	(DATA TEST1A:0004)	Fehlerzelle Segment
- TIME_TMP	(DATA TEST1A:000E)	Adresse des getesteten Zeitkanals
- COUNT_TMP	(DATA TEST1A:0007)	Anzahl der gezaehlten CTC Interrupts

Anschliessend folgen Texte fuer Fehlerausschriften.

Im Segment CODE TEST1A stehen nach den PUBLIC-Erklarungen und den Symboldefinitionen folgende Daten:

```

VERSION_TEST1A DB          '          TEST1A
                DB          'yy-mm-dd '
                DB          'Z.Nr.-Ausg.Nr.-Aend.Mitt.Nr.',OAH,ODH,0
                ORG          40H

```

dabei sind: yy-mm-dd Datum der letzten Aenderung
 Dieser Text wird nach dem Start des Testmoduls in Abhaengigkeit von TDDEBUG (s.Pkt.5.1.) ausgegeben. Die naechste Zeile nach dem ORG-Befehl enthaelt die Ansprungstelle des Testmoduls mit der Marke TEST1A.

8.2.2. Stack- und Registernutzung

Vom Testmodul TEST1A wird der Stack des Leitprogrammes genutzt, d.h. beim Start von TEST1A wird kein SS sondern nur DS initialisiert.

8.2.3. Protokollsteuerung

Die Ausgaben sind je nach den Werten, die die Variablen TDERRONLY und TDDEBUG enthalten, abgestuft unterdrueckbar (s.Pkt.5.1.). Es gelten die Richtlinien gemaess Punkt 8.2.3. der LACS-Beschreibung.

8.2.4. Nutzbare TEST1A-Routinen

TEST1A enthaelt die Routine ASP_RESET, die noch von den Programmen TEST1B und TEST1C genutzt wird.

8.2.5. Nutzbarkeit anderer Routinen

Siehe Punkt 3.4.

9. Fehler

9.1. Fehlerausgaben

Das Programm TEST1A gibt im Fehlerfall einen Text aus, welcher die Fehlerursache kennzeichnet. Danach wird das Leitprogramm angesprungen, welches durch INT 3 den Monitor-Bedienzustand erreicht. Falls undefiniertes Verhalten oder 'Abstuerze' des Rechners den normalen Programmablauf verhindern oder falls keine Fehlerausschriften generiert sind, kann aus den Variablen ERR_TEST1A_OFF und ERR_TEST1A_SEG die Adresse des Textes der Fehlerursache ermittelt und mit Hilfe des Monitorkommandos DISPLAY(D) gelesen werden.

9.2. Fortsetzung nach Fehler

Die Wiederholung des Tests erfolgt durch das Monitorkommando
G<CR>.

10. Verschiedenes

10.1. Programmabbruchmoeglichkeiten

Der Programmablauf kann nicht durch die BREAK-Taste unterbrochen werden. Das Setzen von Haltepunkten im Programm ist verboten, da waehrend des Ablaufes saemtliche Interruptvektoren definiert beschrieben sind. Fuer eine notwendige Fehlersuche kann dies durch Ueberschreiben des Aufrufes CALL FILL_TAB mit NOP-Befehlen unterbunden werden. Danach koennen aber eventuelle falsche Interruptvektoren zum undefinierten Programmabsturz fuehren. Ein geordneter Abbruch ist durch Eingabe von CTRL-C moeglich, damit erfolgt der Uebergang in den Monitorbedienzustand vor der naechsten Ein/Ausgabe.

10.2. Kontrolle der Programmdurchfuehrung

Die Durchfuehrung des Testprogramms wird durch die folgende LACS-Meldung auf dem Bildschirm bestaetigt (Standardfall):

(Testprogr.Nr.) (Benennung des Testprogr.) "PASSED"
Fehler- und Durchlaufzaehler werden in folgender Form ausgegeben:
ERRORS / CYCLES: xxxx / yyyy

Danach beginnt ein neuer Programmzyklus.

Nach Unterbrechung des Programms kann die Durchfuehrungszahl des Testprogramms auch aus der Zelle

CONST_LACS:0178

ermittelt werden (Monitorkommando DW).

Die Anzahl der aufgetretenen Fehlermeldungen kann man der Zelle
CONST_LACS:0176

entnehmen.

1. TEST1B, ASP SERIAL INTERFACE TEST2. Funktion des Programms

Der ASP SERIAL INTERFACE TEST prueft die seriellen Interfaces der Baugruppe ASP, sowie die dazugehoerige Interruptlogik. Der fuer die Realisierung dieser Interfaces verantwortliche Schaltkreis U 856 (SIO) wird dabei komplex getestet, d.h. die Tests laufen gleichzeitig in beiden SIO-Kanaelen. Die Rueckkopplung der Sendeleitungen auf die Empfangsleitungen erfolgt ueber Kurzschlussstecker bzw. ueber einen ASP-internen Kurzschluss. Der Test besteht aus funf Routinen, die in der nachfolgend angefuhrten Reihenfolge aufgerufen werden.

- PME2:

Es wird die Datenuebertragung in der Polling-Betriebsart mit externer Freigabe getestet. Nach der Initialisierung des SIO werden die Statuskanaele abgefragt, ob ein Empfangszeichen verfuegbar ist bzw. ob ein Zeichen gesendet wurde. Ist dies der Fall, erfolgt eine entsprechende Fehlerausschrift. Anschliessend werden Sender und Empfaenger freigegeben und das Empfangszeichen (Kanal A = 55H, Kanal B = AAH) auf Richtigkeit geprueft. Verlaeuft die Datenuebertragung fehlerhaft, wird dies durch eine Fehlerausschrift angezeigt.

- P01:

Es wird die Datenuebertragung in der Polling-Betriebsart getestet. Mit Ausnahme der externen Freigabe von Sender und Empfaenger erfolgt der Ablauf analog dem Programmteil PME2. Es werden ueber beide Testschleifen die Datenwerte von 00H bis FFH uebertragen und auf Richtigkeit kontrolliert (einschliesslich der Paritaetsrichtigkeit).

- INT3:

Dieser Programmteil testet die Interruptlogik des SIO. Es werden zuerst folgende Interrupts ausgeloezt:

- Empfaenger Kanal A
- Sender Kanal A
- Empfaenger Kanal B
- Sender Kanal B

Die richtige Reihenfolge der Interrupteintritte wird ueberprueft. Anschliessend erfolgt eine Neuinitialisierung des U 856 und eine Datenuebertragung mit externer Freigabe von Sender und Empfaenger. Dadurch werden saemtliche moeglichen Interrupts (acht) erzeugt und die Interrupteintritte getestet. Die Richtigkeit der Datenuebertragung wird ueberprueft. Beim Erkennen eines Fehlers erfolgt eine entsprechende Fehlerausschrift.

- PS4:

Es werden zehn Varianten der Interruptausloeung (externe Statusinterrupts) in den Kanaelen A und B getestet. Die Interrupts werden ausgeloezt durch Pegelwechsel auf den Anschluessen /DCD oder /CTS bzw. S2/107, S2/109 und S2/142.

- SYN:

Die Betriebsart monosynchrone Datenuebertragung des Interfaces S2 wird im Kurzschlussbetrieb (Voll-Duplex, interruptgesteuert) geprueft. Die Uebertragungsrates betraegt 19,2 kBit/s. Es erfolgt eine blockweise Datenuebertragung der Werte von 40H - FFH. Nach

Empfang des letzten Zeichens wird die Prüfsumme der Empfangsseite mit der der Senderseite verglichen. Bei Gleichheit wird die Markierung DIO (Datenblock in Ordnung) gesetzt. Abschliessend erfolgt der Vergleich der CRC-Zeichen. Fehler im Übertragungsablauf werden durch entsprechende Ausschriften angezeigt.

3. Voraussetzungen

Im Folgenden werden Speicheradressen in der Form Segment:Offset angegeben. Die Zuordnung von Segmentnamen und absoluten Adressen ist aus der MP2-Liste zu entnehmen.

3.1. Geraeteausrustung

Es ist die Grundausrüstung des Rechners A 7100 erforderlich. Die Ein/Ausgabe richtet sich nach der im Monitor generierten Geraete-konfiguration.

Die Steckverbinder IFSS und S2 (V.24) muessen mit den entsprechenden Kurzschlusssteckern versehen sein.

Die Schalter fuer den Uebertragungstakt des Interfaces S2 muessen auf "synchron" stehen.

3.2. Speicherbedarf

ca. 7 KB fuer TEST1B.OBJ

Die absoluten Adressen der durch den Locator zugewiesenen Speicherbereiche sind anhand der MP2-Liste zu berechnen.

3.3. Test zusaetzlicher Baugruppen

TEST1B prueft zusaetzlich die Interruptlogik der ZVE (busvektori-sierter Interrupt).

3.4. Nutzung anderer Programmodule

Der Testmodul TEST1B.OBJ ist mit dem Bibliotheksmodul LIBRAR.OBJ, dem Leitprogrammmodul LACS.OBJ, dem Modul TEST1A.OBJ und der Bibliothek LACS.LIB verbunden.

Aus LACS.OBJ werden genutzt:
 Anspruenge: - RETURN_FAIL
 - RETURN_PASS
 Routinen: - TDMASKEDMESSAGE
 - TDNEWLINE
 - THREESPACE
 - NEWLINE

Aus LIBRAR.OBJ werden genutzt:
 Routinen: - REST_INTR_MON
 - NMI_DISABLE

Aus TEST1A.OBJ wird genutzt:
- ASP_RESET

4. Ladeprozedur fuer autonomen Ablauf

Das Testprogramm wird im Verbund mit dem Leitprogramm LACS von Minidiskette gebootet (Kommandobeispiele siehe Punkt 4.1 - 4.3.).

4.1. Laden mit Angabe des Geraetenamens

.B :Fn:TEST<CR> (n = Laufwerksnr.)
Nach Erreichen des Monitorbedienzustandes:
.B :Fn:G1B<CR> (n = Laufwerksnr.)
Anschliessend verweilt das Programm im Generierhalt.

4.2. Laden ohne Angabe des Geraetenamens

.B TEST<CR>
Nach Erreichen des Monitorbedienzustandes:
.B G1B<CR>
Bei letzterem Kommando werden TEST bzw. G1B auf allen Laufwerken gesucht, beginnend mit F0. Anschliessend verweilt das Programm im Generierhalt.

4.3. Laden durch Kommandoeingabe im ACT (A 7100-Confidence-Test)

Nach Netzeinschalten oder nach Druecken der Taste RESET besteht die verkuerzte Lademoeglichkeit von LACS durch Eingabe des Zeichens "T" innerhalb des PIC - Tests von ACT. Nach Erreichen des Monitorbedienzustandes weiter wie oben.

5. Generierinformationen

Der Bediener hat die Moeglichkeit, die Ausgabe von Durchlauf- und Fehlermeldungen sowie allgemeine Mitteilungen ueber den Zustand des Prueflings durch Eingabe von Generierinformationen zu steuern (siehe Punkt 5.1.). Ausserdem besteht die Moeglichkeit, zur Fehlereingrenzung, Fehlerverdichtung oder fuer Messzwecke den normalen Ablauf des Testprogrammes zu veraendern (siehe Punkt 5.2.). Dazu muss der Programmablauf unterbrochen werden (z.B. durch CTRL-C). Mittels des Monitorkommandos SW kann dann eine Aenderung der Generierung vorgenommen werden.

5.1. Generierung von Meldungen und Mitteilungen

TEST1B laeuft unter der Regie des Leitprogrammmodules LACS. Deshalb werden Meldungen ueber den Programmablauf durch die Variablen TDERRONLY (Adr. DATA_LACS:0000) und TDDEBUG (Adr. DATA_LACS:0002) gesteuert (siehe Punkt 5.1. der Beschreibung des Leitprogrammes LACS).

5.2. Generierung spezieller Testablaeufer

Spezielle Testablaeufer werden gesteuert durch die Variablen ROUT_TEST1B (Adr. DATA_TEST1B:0000) - Routinesteuerwort
 Bit 0 = PME2, Bit 1 = PO1, Bit 2 = INT3, Bit 3 = PS4
 Bit 4 = SYN

6. Startprozedur

6.1. Start des Testprogramms

Das Testprogramm TEST1B laeuft unter der Regie von LACS. Nach dem Laden (siehe Punkt 4.) erfolgt der Start:

Aus dem Generierhalt durch das Monitorkommando

.G<CR>

Restart erfolgt durch das Monitorkommando

.G <CODE-LACS>:40<CR>.

6.2. Bedieneraktionen

Der Bediener hat folgende Eingriffsmoeglichkeiten waehrend des Programmablaufs:

- CTRL-S = Unterbrechung der Ausgabe, Fortsetzung mit CTRL-Q
- CTRL-Q = Fortsetzung einer unterbrochenen Ausgabe
- CTRL-P = Hardcopy der Bildschirmausgabe auf angeschlossener Drucker. Erneutes CTRL-P setzt die Hardcopy zurueck.
- CTRL-C = Uebergang in den Bedienzustand des Monitors

Die Wirkung dieser Kommandos erfolgt erst unmittelbar vor Ausgabe einer Meldung auf den Bildschirm. Der Uebergang in den Bedienzustand des Monitors ist nicht durch Druecken der Taste BREAK auf der Tastatur moeglich, da der entsprechende Interrupt waehrend des Tests maskiert ist.

7. Einstellungsmoeglichkeiten

Die Standardeinstellung fuer TEST1B lautet:

- TDERRONLY: 0 ; siehe Punkt 5.1. LACS-Beschreibung
 - TDDEBUG: 3 ; " " " " " "
 - ROUT_TEST1B: 1F ; 5 Routinen
- (siehe Punkt 5. Programmbeschreibung TEST1B)

8. Programmbeschreibung8.1. Programmablauf

Vor dem 1. Durchlauf erfolgt die Ausgabe der Programmnummern, Modulnamen und der Zeichnungs- und Ausgabennummern aller beteiligten Module. Sie hat fuer TEST1B folgendes Aussehen (siehe Punkt 8.2.1.):

```
TEST1B yy-mm-dd 1.56.703028.3/67-aa-56ACnnn
        (yy-mm-dd = Erstellungsdatum
         aa       = Ausgabennummer
         56ACnnn = Nr. der Aenderungsmittlung).
```

Anschließend beginnt die eigentliche Pruefung. Die Tests werden in der Reihenfolge PME2, PO1, INT3, PS4, SYN abgearbeitet. Die Rueckkehr zum Leitprogramm erfolgt im fehlerfreien Fall durch einen Ruecksprung

```
JMP RETURN PASS.
```

Die Laufzeit betraegt dabei ca. 3 s.

Im Fehlerfall informiert das Testprogramm den Bediener durch Fehlerausschriften (beachte Punkt 5. und 9.) und uebergibt die Regie an das Leitprogramm durch einen

```
JMP RETURN FAIL.
```

Abbruch des Testprogramms TEST1B siehe Punkt 10.

8.2. Anschlussbedingungen8.2.1. Struktur des Quellprogramms

Der Testmodul TEST1B ist folgendermassen aufgebaut:

1. TITLE(TEST1B Zeichnungsnr.-Ausg.Nr.-Aend.Mitt.Nr.)
2. NAME TEST1B
3. Anschließend folgt die Programmbeschreibung nach dem Muster der vorliegenden Beschreibung des Leitprogrammes.
4. Es folgen die EXTRN-Erklaerungen.
5. Es folgen die Vereinbarungen symbolischer Adressen.
6. Es folgt das Datensegment mit der Bezeichnung DATA TEST1B.
7. Es folgt das Codesegment mit der Bezeichnung CODE TEST1B.
8. PUBLIC-Erklaerungen stehen in dem Segment, in dem die betreffenden Symbole vorkommen, ganz am Anfang des Segments.

Das Datensegment DATA TEST1B enthaelt folgende wichtige Wortvariablen:

```
- ROUT TEST1B      (DATA TEST1B:0000)   Routinesteuerwort
- ERR TEST1B OFF   (DATA TEST1B:0002)   Fehlerzelle Offset
- ERR TEST1B SEG   (DATA TEST1B:0004)   Fehlerzelle Segment
```

Anschließend folgen Texte fuer Fehlerausschriften sowie Variablen fuer Ausgabe- und Erwartungswerte getrennt nach ihrer Zugehoerigkeit zu den einzelnen Testroutinen.

Im Segment CODE TEST1B stehen nach den PUBLIC-Erklarungen und den Symboldefinitionen folgende Zeilen:

```
VERSION_TEST1B DB ' TEST1B
                DB 'yy-mm-dd '
                DB 'Z.Nr.-Ausg.Nr.-Aend.Mitt.Nr.', OAH, ODH, O
                ORG 40H
```

dabei sind: yy-mm-dd Datum der letzten Aenderung
Dieser Text wird nach dem Start des Testmoduls in Abhaengigkeit von TDDEBUG (s.Pkt.5.1.) ausgegeben. Die naechste Zeile nach dem ORG-Befehl enthaelt die Ansprungstelle des Testmoduls mit der Marke TEST1B.

8.2.2. Stack- und Registernutzung

Vom Testmodul TEST1B wird der Stack des Leitprogrammes genutzt, d.h.beim Start von TEST1B wird kein SS sondern nur DS initialisiert.

8.2.3. Protokollsteuerung

Die Ausgaben sind je nach den Werten, die die Variablen TDERRONLY und TDDEBUG enthalten, abgestuft unterdrueckbar (s.Pkt.5.1.). Es gelten die Richtlinien gemaess Punkt 8.2.3. der LACS-Beschreibung.

8.2.4. Nutzbare TEST1B-Routinen

TEST1B enthaelt keine Routinen, die von anderen Programmen nutzbar sind.

8.2.5. Nutzbarkeit anderer Routinen

Siehe Punkt 3.4.

9. Fehler

9.1. Fehlerausgaben

Das Programm TEST1B gibt im Fehlerfall einen Text aus, welcher die Fehlerursache kennzeichnet. Danach wird das Leitprogramm angesprungen, welches durch INT 3 den Monitor-Bedienzustand erreicht. Falls undefiniertes Verhalten oder 'Abstuerze' des Rechners den normalen Programmablauf verhindern oder falls keine Fehlerausschriften generiert sind, kann aus den Variablen ERR_TEST1B_OFF und ERR_TEST1B_SEG die Adresse des Textes der Fehlerursache ermittelt und mit Hilfe des Monitorkommandos DISPLAY(D) gelesen werden.

9.2. Fortsetzung nach Fehler

Die Wiederholung des Tests erfolgt durch das Monitorkommando G<CR>.

10. Verschiedenes

10.1. Programmabbruchmoeglichkeiten

Der Programmablauf kann nicht durch die BREAK-Taste unterbrochen werden. Das Setzen von Haltepunkten im Programm ist verboten, da waehrend des Ablaufes saemtliche Interruptvektoren definiert beschrieben sind. Fuer eine notwendige Fehlersuche kann dies durch Ueberschreiben des Aufrufes CALL FILL_ITAB mit NOP-Befehlen unterbunden werden.

Danach koennen aber eventuelle falsche Interruptvektoren zum undefinierten Programmabsturz fuehren. Ein geordneter Abbruch ist durch Eingabe von CTRL-C moeglich, damit erfolgt der Uebergang in den Monitorbedienzustand vor der naechsten Ein/Ausgabe.

10.2. Kontrolle der Programmdurchfuehrung

Die Durchfuehrung des Testprogramms wird durch die folgende LACS-Meldung auf dem Bildschirm bestaetigt (Standardfall):

(Testprogr.Nr.) (Benennung des Testprogr.) "PASSED"
Fehler- und Durchlaufzaehler werden in folgender Form ausgegeben.

ERRORS / CYCLES: xxxx / yyyy

Danach beginnt ein neuer Programmzyklus. Nach Unterbrechung des Programms kann die Durchfuehrungszahl des Testprogramms TEST1B auch aus der Zelle

CONST_LACS:0186

ermittelt werden (Monitorkommando DW).

Die Anzahl der aufgetretenen Fehlermeldungen kann man der Zelle

CONST_LACS:0184

entnehmen.

TEST1B

Betriebsdokumentation A 7100, Bd.3

1. TEST1C, ASP PARALLEL INTERFACE TEST

2. Funktion des Programms

Der ASP PARALLEL INTERFACE TEST prueft das Interface IFSP der Baugruppe ASP, sowie die dazugehoerige Interruptlogik. Der Test besteht aus zwei Routinen.

- IFSP:

Der PIO U 855 der ASP wird programmiert (Kanal A Bit Ein-/Ausgabe, Kanal B Byte Ausgabe). Anschliessend wird jeweils eine Daten- bzw. Steuerleitung aktiviert und das Ergebnis der durch logische Verknuepfung, entsprechend der Belegung des Kurzschlusssteckers, entstandenen Bitkombination am PIO-Kanal eingelesen und mit einem Erwartungswert verglichen. Es gilt folgende Belegung des Kurzschlusssteckers:

A0 - S0 - D0 - D4

A1 - S1 - D1 - D5

A2 - S2 - D2 - D6

A3 - S3 - D3 - D7

A4 - S4 - SC - DPO (PARITAETSBIT)

AC - S5

Im Fehlerfall wird die entsprechend Daten- bzw. Steuerleitung ausgegeben, bei deren Aenderung der Fehler auftrat.

- PIO_IA:

Durch den Kanal A des PIO wird ein Interrupt ausgeloeset. Es erfolgt dabei die Pruefung der Interruptlogik des PIO, der ASP sowie der ZVE (Logik fuer busvektorierten Interrupt). Auftretende Fehler (Time out, falsche Uebertragung des Vektors) werden angezeigt.

3. Voraussetzungen

Im Folgenden werden Speicheradressen in der Form Segment:Offset angegeben. Die Zuordnung von Segmentnamen und absoluten Adressen ist aus der MP2-Liste zu entnehmen.

3.1. Geraeteausruestung

Es ist die Grundausruestung des Rechners A 7100 erforderlich. Die Ein/Ausgabe richtet sich nach der im Monitor generierten Geraete-konfiguration. Der IFSP - Steckverbinder (PARALLEL) muss mit dem entsprechenden Kurzschlussstecker versehen sein. Die Schalter fuer die Steuerleitungen des IFSP auf der Baugruppe muessen auf "asynchron" stehen.

3.2. Speicherbedarf

ca. 1 KB fuer TEST1C.OBJ

Die absoluten Adressen der durch den Locator zugewiesenen Speicherbereiche sind anhand der MP2-Liste zu berechnen.

3.3. Test zusaetzlicher Baugruppen

TEST1C prueft zusaetzlich die Interruptlogik der ZVE (busvektori-
sierter Interrupt).

3.4. Nutzung anderer Programmmodule

Der Testmodul TEST1C.OBJ ist mit dem Bibliotheksmodul LIBRAR.OBJ,
dem Leitprogrammmodul LACS.OBJ, dem Modul TEST1A.OBJ und der Bi-
bliothek LACS.LIB verbunden.

Aus LACS.OBJ werden genutzt:
Anspruege: - RETURN FAIL
Routinen: - TDMASKEDMESSAGE
- NEWLINE
- THREESPACE
- ASCII_CONV

Aus LIBRAR.OBJ werden genutzt:
Routinen: - REST_INTR_MON
- NMI_DISABLE

Aus TEST1A.OBJ wird genutzt:
- ASP_RESET

4. Ladeprozedur fuer autonomen Ablauf

Das Testprogramm wird im Verbund mit dem Leitprogramm LACS von
Minidiskette gebootet (Kommandobeispiele siehe Punkt 4.1. -
4.3.).

4.1. Laden mit Angabe des Geraetenamens

.B :Fn:TEST<CR> (n = Laufwerksnr.)
Nach Erreichen des Monitorbedienzustandes:
.B :Fn:G1C<CR> (n = Laufwerksnr.)
Anschliessend verweilt das Programm im Generierhalt.

4.2. Laden ohne Angabe des Geraetenamens

.B TEST<CR>
Nach Erreichen des Monitorbedienzustandes:
.B G1C<CR>
Bei letzterem Kommando werden TEST bzw. G1C auf allen Laufwerken
gesucht, beginnend mit FO. Anschliessend verweilt das Programm im
Generierhalt.

4.3. Laden durch Kommandoeingabe im ACT (A 7100-Confidence-Test)

Nach Netzeinschalten oder nach Druecken der Taste RESET besteht die verkuerzte Lademoeglichkeit von LACS durch Eingabe des Zeichens "T" innerhalb des PIC - Tests von ACT. Nach Erreichen des Monitorbedienzustandes weiter wie oben.

5. Generierinformationen

Der Bediener hat die Moeglichkeit, die Ausgabe von Durchlauf- und Fehlermeldungen sowie allgemeine Mitteilungen ueber den Zustand des Prueflings durch Eingabe von Generierinformationen zu steuern (siehe Punkt 5.1.). Ausserdem besteht die Moeglichkeit, zur Fehlereingrenzung, Fehlerverdichtung oder fuer Messzwecke den normalen Ablauf des Testprogrammes zu veraendern (siehe Punkt 5.2.). Dazu muss der Programmablauf unterbrochen werden (z.B. durch CTRL-C). Mittels des Monitorkommandos SW kann dann eine Aenderung der Generierung vorgenommen werden.

5.1. Generierung von Meldungen und Mitteilungen

TEST1C laeuft unter der Regie des Leitprogrammmodules LACS. Deshalb werden Meldungen ueber den Programmablauf durch die Variablen TDERRONLY (Adr. DATA_LACS:0000) und TDDEBUG (Adr. DATA_LACS:0002) gesteuert (siehe Punkt 5.1. der Beschreibung des Leitprogrammes LACS).

5.2. Generierung spezieller Testablaufe

Spezielle Testablaufe werden gesteuert durch die Variable ROUT_TEST1C (Adr. DATA_TEST1C:0000) - Routinesteuerwort Bit 0 = IFSP, Bit 1 = PIO_IA.

6. Startprozedur

6.1. Start des Testprogramms

Das Testprogramm TEST1C laeft unter der Regie von LACS. Nach dem Laden (siehe Punkt 4.) erfolgt der Start:

- Aus dem Generierhalt durch das Monitorkommando
.G<CR>
- Restart erfolgt durch das Monitorkommando
.G <CODE-LACS>:40<CR>.

6.2. Bedieneraktionen

Der Bediener hat folgende Eingriffsmoeglichkeiten waehrend des Programmablaufs:

- CTRL-S = Unterbrechung der Ausgabe, Fortsetzung mit CTRL-Q
- CTRL-Q = Fortsetzung einer unterbrochenen Ausgabe
- CTRL-P = Hardcopy der Bildschirmausgabe auf angeschlossenen Drucker. Erneutes CTRL-P setzt die Hardcopy zurueck.
- CTRL-C = Uebergang in den Bedienzustand des Monitors

Die Wirkung dieser Kommandos erfolgt erst unmittelbar vor Ausgabe einer Meldung auf den Bildschirm. Der Uebergang in den Bedienzustand des Monitors ist nicht durch Druucken der Taste BREAK auf der Tastatur moeglich, da der entsprechende Interrupt waehrend des Tests maskiert ist.

7. Einstellungsmoeglichkeiten

Die Standardeinstellung fuer TEST1C lautet:

- TDERRONLY: 0 ; siehe Punkt 5.1. LACS-Beschreibung
 - TDDEBUG: 3 ; " " " "
 - ROUT TEST1C: 3 ; 2 Routinen
- (siehe Punkt 5. Programmbeschreibung TEST1C)

8. Programmbeschreibung8.1. Programmablauf

Vor dem 1. Durchlauf erfolgt die Ausgabe der Programmnummern, Modulnamen und der Zeichnungs- und Ausgabennummern aller beteiligten Module. Sie hat fuer TEST1C folgendes Aussehen (siehe Punkt 8.2.1.):

```
TEST1C yy-mm-dd 1.56.703029.1/67-aa-56ACnnn
        (yy-mm-dd = Erstellungsdatum
          aa       = Ausgabennummer
          56ACnnn = Nr. der Aenderungsmittelung).
```

Anschliessend beginnt die eigentliche Pruefung. Die Tests werden in der Reihenfolge IFSP, PIO IA abgearbeitet. Die Rueckkehr zum Leitprogramm erfolgt im fehlerfreien Fall durch einen Ruecksprung (RET) mit der Uebergabe des Wertes 1 im Register AX. Die Laufzeit betraegt dabei ca. 0,3 s. Im Fehlerfall informiert das Testprogramm den Bediener durch Fehlerausschriften (beachte Punkt 5. und 9.) und uebergibt die Regie an das Leitprogramm durch einen

```
JMP RETURN FAIL.
```

Abbruch des Testprogramms TEST1C siehe Punkt 10.

8.2. Anschlussbedingungen8.2.1. Struktur des Quellprogramms

Der Testmodul TEST1C ist folgendermassen aufgebaut:

1. TITLE(TEST1C Zeichnungsnr.-Ausg.Nr.-Aend.Mitt.Nr.)
2. NAME TEST1C
3. Anschliessend folgt die Programmbeschreibung nach dem Muster der vorliegenden Beschreibung des LACS-Moduls.
4. Es folgen die EXTRN-Erklarungen.
5. Es folgen die Vereinbarungen symbolischer Adressen.
6. Es folgt das Datensegment mit der Bezeichnung DATA TEST1C.
7. Es folgt das Codesegment mit der Bezeichnung CODE TEST1C.
8. PUBLIC-Erklarungen stehen in dem Segment, in dem die betreffenden Symbole vorkommen, ganz am Anfang des Segments.

Das Datensegment DATA TEST1C enthaelt folgende wichtige Wortvariablen:

```
- ROUT TEST1C          (DATA TEST1C:0000)   Routinesteuerwort
- ERR TEST1C_OFF      (DATA TEST1C:0002)   Fehlerzelle Offset
- ERR TEST1C_SEG      (DATA TEST1C:0004)   Fehlerzelle Segment
```

Anschliessend folgen Texte fuer Fehlerausschriften sowie Tabellen fuer Ausgabe- und Erwartungswerte.

Im Segment CODE TEST1C stehen nach den PUBLIC-Erklarungen und den Symboldefinitionen folgende Zeilen:

```
VERSION_TEST1C DB      '          TEST1C '
                DB      'yy-mm-dd '
                DB      'Z.Nr.-Ausg.Nr.-Aend.Mitt.Nr.',OAH,ODH,O
                ORG      40"
```

dabei sind: yy-mm-dd Datum der letzten Aenderung
Dieser Text wird nach dem Start des Testmoduls in Abhaengigkeit von TDDEBUG (s.Pkt.5.1.) ausgegeben. Die naechste Zeile nach dem ORG-Befehl enthaelt die Anspringstelle des Testmoduls mit der Marke TEST1C.

8.2.2. Stack- und Registernutzung

Vom Testmodul TEST1C wird der Stack des Leitprogrammes genutzt, d.h. beim Start von TEST1C wird kein SS sondern nur DS initialisiert.

8.2.3. Protokollsteuerung

Die Ausgaben sind je nach den Werten, die die Variablen TDERRONLY und TDDEBUG enthalten, abgestuft unterdrueckbar (s.Pkt.5.1.). Es gelten die Richtlinien gemaess Punkt 8.2.3. der LACS-Beschreibung.

8.2.4. Nutzbare TEST1C-Routinen

TEST1C enthaelt keine Routinen, die von anderen Programmen nutzbar sind.

8.2.5. Nutzbarkeit anderer Routinen

Siehe Punkt 3.4.

9. Fehler

9.1. Fehlerausgaben

Das Programm TEST1C gibt im Fehlerfall einen Text aus, welcher die Fehlerursache kennzeichnet. Danach wird das Leitprogramm angesprungen, welches durch INT 3 den Monitor-Bedienzustand erreicht. Falls undefiniertes Verhalten oder 'Abstuerze' des Rechners den normalen Programmablauf verhindern oder falls keine Fehlerausschriften generiert sind, kann aus den Variablen ERR TEST1C_OFF und ERR TEST1C_SEG die Adresse des Textes der Fehlerursache ermittelt und mit Hilfe des Monitorbefehls DISPLAY(D) gelesen werden.

9.2. Fortsetzung nach Fehler

Die Wiederholung des Tests erfolgt durch das Monitorbefehl G<CR>.

10. Verschiedenes

10.1. Programmabbruchmoeglichkeiten

Der Programmablauf kann nicht durch die BREAK-Taste unterbrochen werden. Das Setzen von Haltepunkten im Programm ist verboten, da waehrend des Ablaufes saemtliche Interruptvektoren definiert beschrieben sind. Fuer eine notwendige Fehlersuche kann dies durch Ueberschreiben des Aufrufes CALL FILL_ITAB mit NOP-Befehlen unterbunden werden. Danach koennen aber eventuelle falsche Interruptvektoren zum undefinierten Programmabsturz fuehren. Ein geordnetes Abbruch ist durch Eingabe von CTRL-C moeglich, damit erfolgt der Uebergang in den Monitorbedienzustand vor der naechsten Ein/Ausgabe.

10.2. Kontrolle der Programmdurchfuehrung

Die Durchfuehrung des Testprogramms wird durch die folgende LACS-Meldung auf dem Bildschirm bestaetigt (Standardfall):
(Testprogr.Nr.) (Benennung des Testprogr.) "PASSED"
Fehler- und Durchlaufzaehler werden in folgender Form ausgegeben:
ERRORS / CYCLES: xxxx / yyyy

Danach beginnt ein neuer Programmzyklus. Nach Unterbrechung des Programms kann die Durchfuehrungszahl des Testprogramms auch aus der Zelle

CONST_LACS:0194

ermittelt werden (Monitorkommando DW).

Die Anzahl der aufgetretenen Fehlermeldungen, kann man der Zelle

CONST_LACS:0192

entnehmen.

TEST10

Betriebsdokumentation A 7100, Bd.3

1. ABS/ABG-Testprogramm

2. Funktion des Programms

TEST1D ist ein nur unter Steuerung des Leitprogramms LACS arbeitendes Programm zum Test der Baugruppen ABS und KGS/ABG. Das Programm ist zur besseren Fehlerlokalisierung in verschiedene Teile untergliedert, die folgerichtig aufeinander aufbauen und fuer den Gesamttest eine geschlossene Einheit bilden. Die Struktur der zu testenden Baugruppen macht es erforderlich, dass neben den vom Rechner kontrollierbaren Funktionen es auch solche gibt, die nur einer visuellen Kontrollmoeglichkeit unterliegen (Videostrecke). Zu diesem Zweck erscheint auf dem angeschlossenen Bildschirm ein entsprechend gut uebersehbares Testmuster sowohl fuer die Alphanumerik als auch fuer die Grafik, welche einige Sekunden zur Betrachtung dargeboten werden. Ein Bedieneringriff ist nicht erforderlich. Der Alphanumerikteil wird gleichartig bei den beiden Baugruppen durchlaufen. Die Verzweigung zum Test der grafischen Funktionen geschieht automatisch in Abhaengigkeit von der gesteckten Baugruppe. Bei der Abarbeitung des Initialtests innerhalb der Baugruppen, der einen grundlegenden Teil des Gesamttests bildet, erfolgt ein Loeschen des Bildinhaltes, was bei der Wahl der Reihenfolge im Leitprogrammverbund zu beachten ist.

3. Voraussetzungen

3.1. Geraetaeusruetzung

Es ist die Grundausruestung des Rechners A 7100 erforderlich. Insbesondere ist der Bildschirm an die zu testenden Baugruppen ABS bzw.KGS/ABG anzuschliessen.

3.2. Speicherbedarf

5 KB

Die absoluten Adressen der durch den Locator zugewiesenen Speicherbereiche sind anhand der MP2-Liste zu berechnen.

3.3. Test* zusaetzlicher Baugruppen

Ein Test des angeschlossenen Bildschirms ist zwangsweise mit eingeschlossen.

3.4. Nutzung anderer Module

Der Bibliotheksmodul SAVE1D ist zur Einbindung in das LACS unbedingt erforderlich.

4. Ladeprozedur fuer autonomen Ablauf

Das Testprogramm wird im Verbund mit dem LACS von Minidiskette gebootet.

4.1. Laden mit Angabe des Geraetenamens

„B :Fn:TEST<CR> (n = Laufwerksnr.)
 Nach Erreichen des Monitorbedienzustandes:
 „B :Fn:G1D<CR> (n = Laufwerksnr.)
 Anschliessend verweilt das Programm im Generierhalt.

4.2. Laden ohne Angabe des Geraetenamens

„B TEST<CR>
 Nach Erreichen des Monitorbedienzustandes:
 „B G1D<CR>
 Bei letzterem Kommando werden TEST bzw. G1D auf allen Laufwerken gesucht, beginnend mit FO. Anschliessend verweilt das Programm im Generierhalt.

4.3. Laden durch Kommandoeingabe im ACT (A 7100-Confidence-Test)

Nach Netzeinschalten oder nach Druucken der Taste RESET besteht die verkuerzte Lademoeglichkeit von LACS durch Eingabe des Zeichens "T" innerhalb des PIC - Tests von ACT. Nach Erreichen des Monitorbedienzustandes weiter wie oben.

5. Generierinformationen

Der Bediener hat die Moeglichkeit, die Ausgabe von Durchlauf- und Fehlermeldungen durch Eingabe von Generierinformationen zu steuern.

5.1. Generierung von Meldungen und Mitteilungen

TEST1D laeuft unter der Regie des Leitprogrammes LACS. Deshalb werden Meldungen ueber den Programmablauf durch die Variablen TDERRONLY (Adr. DATA_LACS:0000) und TDDEBUG (Adr. DATA_LACS:0002) gesteuert. (Siehe Punkt 5.1. der Beschreibung des Leitprogrammes LACS).

6. Startprozedur

Das Testprogramm TEST1D laeuft unter der Regie von LACS Nach dem Laden (siehe Punkt 4.) erfolgt der Start:
 - Aus dem Generierhalt durch das Monitorkommando

- .G<CR>
- Restart erfolgt durch das Monitorkommando
- .G <CODE LACS>.40<CR>.

7. Einstellungsmöglichkeiten

Gemaess Standardeinstellung wird der Test zyklisch wiederholt. Pro Zyklus wird das Testprogramm einmal abgearbeitet.

8. Programmbeschreibung

Das Leitprogramm springt TEST1D ueber den als Verbindungsprogramm dienenden Modul SAVE1D an. Von da aus erfolgt ein CALL zum Eintrittspunkt G01D. Nach der Ausschrift ABS TEST bzw. ABG TEST in invertierter Form, der Ausgabe der Firmware Version der zu testenden Baugruppe (nur bei ABG-Test) werden vor dem 1. Durchlauf die Programmnummer, der Modulname und die Zeichnungs- und Ausgabennummer mit folgendem Aussehen ausgegeben:

```
TEST1D yy-mm-dd 1.56.703030.6/67-aa-56ACnnn
```

```
(yy-mm-dd = Erstellungsdatum
aa         = Ausgabennummer
56ACnnn   = Nr. der Aenderungsmittelung).
```

Danach erscheint ein vierzeiliges Testmuster ueber die gesamte Zeilenlaenge. Die 3. Zeile wird mittels des 2. Zeichensatzes als Quasigrafik realisiert. Die vorhergehende Dunkelstzeit (Loeschen des Bildschirms) von 3 sec beim ABS-Test und 11 sec beim ABG-Test laesst den Ablauf des Initialtests erkennen. Bei letzterem erfolgt zusaetzlich der Test der Grafikfunktionen mit einem ueber den gesamten Bildschirm gehenden Testbild, welches durch automatisches Verschieben der Splitgrenze (Grenze zwischen alphanumerischer und grafischer Darstellung) wieder aufgeloeset wird, so dass das alphanumerische Bild wieder erscheint. Beim ABG-Test muessen dabei die beiden Kurzschluss-Stecker (KGS TABLE/PRINT) aufgesteckt sein. Ueber den Modul SAVE1D wird die Steuerung mit

```
JMP RETURN_PASS
```

wieder dem Leitprogramm uebergeben. Das Quellprogramm ist in PL/M geschrieben.

9. Fehler

9.1. Fehlerausgaben

Wird im TEST1D ein Fehler erkannt, so erfolgt der Aufbau einer entsprechenden Fehlermeldung im Fehlerpuffer BUFFER1D bzw. DIAGFILE1D. Ueber einen Sprung zum Modul SAVE1D (FM01D oder FM11D) wird die Uebergabe der Pufferadresse ueber den Stack realisiert und nach einem CALL TDMASKEDMESSAGE ein Ruecksprung JMP RETURN_FAIL zum Leitprogramm ausgeloeset, welches die Fehler-

ausschrift abhaengig von bestimmten Bedingungen ausfuehrt. Wird ein Fehler nur visuell erkannt, so liegt es in der Hand des Bedieners, entsprechend zu reagieren.

Fehlertabelle:

Fehlerausschriften	Fehler	
INTERFACE TEST FAILED,ERROR	F1:<STA>	IBF oder OBF nicht im Grundzustand
"	F2:<STA>	Status nicht geloescht
"	F3:<STA>	INT-Bit nicht gesetzt
"	F4:<STA>	wie F2
"	F5:<STA>	wie F3
"	F6:<STA>	Status ungleich 0
"	F7:<STA>	Status ungleich 84H (INT=1;ERR=1)
"	F8:<STA>	wie F6
DEV. ATTR. FAILED, INITIAL TEST FAILED, FIRST MESSAGE FAILED, COMMAND CANCEL FAILED, CHAR-GEN-LOADING FAILED, WRITING ON SCREEN FAILED, ABG DIAGNOSIS FAILED, GRAPHIC TEST FAILED, STATE ILLEGAL	ERROR	F9:<STA> IBF ungleich 0
"	"	"
"	"	"
"	"	"
"	"	"
"	"	"
"	"	"
"	ERROR FO	"
DEV. ATTR. FAILED,	ERROR	FA:<STA> OBF nicht gesetzt (1.Byte erwartet)
"	FB:<DAT>	1.Byte fehlerhaft (1BH erwartet)
"	FC:<STA>	Status ungleich 5 (OBF=1;INT=1)
"	FD:<DAT>	2.,3.,4. oder 5.Byte falsch (5BH,3FH,32H,3BH erwartet)
"	FE:<STA>	Status ungleich 5 (OBF=1;INT=1)
"	FF:<DAT>	6.Byte nicht im Bereich 0...9
"	FG:<STA>	Status ungleich 5 (OBF=1;INT=1)
"	FH:<DAT>	7.Byte fehlerhaft (63H erwartet)

Fehlerrauschrift	Fehler
INITIAL TEST FAILED, ERROR	FI:<STA> OBF nicht gesetzt (1.Byte erwartet)
"	FJ:<DAT> 1.Byte fehlerhaft (1BH erwartet)
"	FK:<STA> OBF nicht gesetzt (2.Byte erwartet)
"	FL:<DAT> 2.Byte fehlerhaft (5BH erwartet)
"	FM:<STA> OBF nicht gesetzt (3.Byte erwartet)
"	FN:<DAT> 3.Byte fehlerhaft (30H oder 33H erwartet)
"	FO:<STA> OBF nicht gesetzt (4.Byte erwartet)
"	FP:<DAT> 4.Byte fehlerhaft (6BH erwartet)
"	FQ:00 Initialtest mit NO-GO gemeldet
COMMAND CANCEL FAILED, ERROR	FR:<STA> ERR-Bit nicht gesetzt
INTERFACE TEST FAILED, ERROR	FS:<STA> INT7 nicht gesetzt
"	FT:<STA> INT7 ungleich 0
ABG DIAGNOSIS FAILED, ERROR	FU:<STA> OBF nicht gesetzt (1.Diagnosebyte erwartet)
"	FV:<STA> OBF nicht gesetzt (DIAGFILE nicht vollst.)
GRAPHIC TEST FAILED, ERROR	FW:<STA> OBF nicht gesetzt (Fehlerbyte erwartet)
"	FX:<F-BYTE> TABLE/PRINT-Interface fehlerh.
	TABLE:<F-BYTE> Fehler
	01 CTS,DCD bei DTR u. RTS aktiv
	02 CTS "
	03 DCD "
	04 Paritaetsfehler
	05 DCD bei DTRu.RTS inaktiv
	06 CTS "
	07 CTS,DCD "
	08 Daten
	09 Daten

Fehlerausschrift	Fehler
GRAPHIC TEST FAILED, ERROR	PRINT:<F-BYTE> OA Daten OB Daten
GRAPHIC TEST FAILED: "Ausschrift des DIAGFILES"	siehe Betriebsdok.

9.2. Fortsetzung nach Fehler

Bei Auftreten eines Fehlers wird die Steuerung dem Leitprogramm uebergeben.

10. Verschiedenes

10.1. Programmabbruchmoeglichkeiten

Ein Abbruch innerhalb TEST1D ist nicht vorgesehen. Durch Eingabe von CTRL-C kann der Test abgebrochen werden, wenn das Leitprogramm die Steuerung innehat.

10.2. Kontrolle der Programmdurchfuehrung

Die Kontrolle ist durch die unter Pkt.8 beschriebenen Ausgaben auf den Bildschirm (Bildschirmloeschen, Ueberschrift, Testmuster, Testbild) in jedem Falle gegeben.

1. TEST1E, KES Test

2. Funktion des Programms

Der TEST1E prueft umfassend die Funktionsfaehigkeit des KES, wobei alle definierten Funktionskommandos, die den KES unabhangig von einem physischen Modul (AFS/AFP) betreiben, variantenreich getestet werden.

3. Voraussetzungen

Im Folgenden werden Speicheradressen in der Form Segment:Offset angegeben. Die Zuordnung von Segmentnamen und absoluten Adressen ist aus der MP2-Liste zu entnehmen.

3.1. Geraetaeusruistung

Es ist die Grundausruestung des Rechners A 7100 erforderlich mit einem Speicherbereich bis mindestens 40000H. Die Bildschirm-Ausgabe bzw. die Tastatur-Eingabe richtet sich nach der im Monitor generierten Geraetekonfiguration.

3.2. Speicherbedarf

ca. 17 KB fuer TEST1E.OBJ

Die absoluten Adressen werden durch den Locator festgelegt und sind mit Hilfe der MP2-Liste zu ermitteln.

3.3. Test zusaetzlicher Baugruppen

- Die Interruptebene 5 des PIC wird in der KES-Interrupt-Routine ueberprueft.
- Ein installierter ZPS wird einerseits von der Systembusseite (X1-BUS) und andererseits von der ZVE-Seite (X2-BUS) dadurch uebergeprueft, dass bei einer DMA-Uebertragung zwischen KES und ZPS gleichzeitig ein ZVE-Stringbefehl abgearbeitet wird.
- Die Funktionsueberpruefung der EA-Register auf der ZVE K2771 erfolgt waehrend eines DMA-Betriebes mit dem KES.

3.4. Nutzung anderer Programmmodule

Der Testmodul TEST1E.OBJ ist mit dem Bibliotheksmodul LIBRAR.OBJ, dem Leitprogrammmodul LACS.OBJ und der Bibliothek SERVIC.LIB verbunden.

Aus LACS.OBJ werden genutzt:

Routinen: - ASCII CONV
 - TDMASKEDMESSAGE
 - TDDISPLAY

Aus LIBRAR.OBJ werden genutzt:

Routinen: - REST_INTR MON
 - FILL_WRONGSWI
 - RESET_59A

4. Ladeprozedur fuer autonomen Ablauf

Das Testprogramm wird im Verbund mit dem Leitprogramm LACS von Minidiskette gebootet. (Kommandobeispiele siehe Punkt 4.1. - 4.3.).

4.1. Laden mit Angabe des Geraetenamens

.B :Fn:TEST<CR> (n = Laufwerknr.)
 Nach Erreichen des Monitorbedienzustandes:
 .B :Fn:G1E<CR> (n = Laufwerknr.)
 Anschliessend verweilt das Programm im Generierhalt.

4.2. Laden ohne Angabe des Geraetenamens

.B TEST<CR>
 Nach Erreichen des Monitorbedienzustandes:
 .B G1E<CR>
 Bei letzterem Kommando werden TEST bzw. G1E auf allen Laufwerken gesucht, beginnend mit FO.
 Anschliessend verweilt das Programm im Generierhalt.

4.3. Laden durch Kommandoeingabe im ACT (A7100 - Confidence - Test)

Nach Netzeinschalten oder nach Druecken der Taste RESET besteht die verkuerzte Lademoeglichkeit von LACS durch Eingabe des Zeichens "T" innerhalb des PIC - Tests von ACT.
 Nach Erreichen des Monitorbedienzustandes weiter wie oben.

5. Generierinformationen

Der Bediener hat die Moeglichkeit, die Ausgabe von Durchlauf- und Fehlermeldungen sowie allgemeine Mitteilungen ueber den Zustand des Pruefblings durch Eingabe von Generierinformationen zu steuern. (siehe Punkt 5.1).
 Ausserdem besteht die Moeglichkeit zur Fehlereingrenzung, Fehlerverdichtung oder fuer Messzwecke den normalen Ablauf des Testprogrammes zu veraendern (siehe Punkt 5.2).
 Dazu muss der Programmablauf unterbrochen werden (z.B. durch

CTRL-C). Mittels des Monitorkommandos SW bzw. S kann dann eine Aenderung der Generierung vorgenommen werden.

5.1. Generierung von Meldungen und Mitteilungen

TEST1E laeuft unter der Regie des Leitprogrammmodules LACS. Deshalb werden Meldungen ueber den Programmablauf durch die Variablen TDERRONLY (Adr. DATA_LACS:0000) und TDDEBUG (Adr. DATA_LACS:0002) gesteuert. (siehe Punkt 5.1 der Beschreibung des Leitprogrammes LACS)

5.2. Generierung spezieller Testablaeufer

Die Hauptroutine-Steuerung erfolgt mit den Bits der Variablen:

```
- ROUT-TEST1E <Adr. DATA_TEST1E:0> - Routine-Steuerwort
  BIT0 = 1 : KES RESET-Routine
  BIT1 = 1 : DIAGNOSE-Routine
  BIT2 = 1 : KES-PUFFER-EIN/AUS-Routine
  BIT3 = 1 : START-UA880-PROGR.-Routine
  BIT4 = 1 : SUBPORT-DMA-Routine
  BIT5 = 1 : INTERRUPT-TEST-Routine
  BIT6 = 1 : DMA_SPEZIAL-Routine
```

Zu jeder Hauptroutine gehoert ein Subroutine-Steuerbyte, dessen Bits die Subroutinen der betreffenden Hauptroutine steuern. Diese Subroutinen-Steuerbytes folgen unmittelbar auf das Steuerwort ROUT_TEST1E, ihre Bedeutung ist an Hand der Programmliste zu erkennen.

```
- SB_KES_RESET <Adr.DATA_TEST1E:2> - Subroutine-Steuerbyte
- SB_DIAGNOSE <Adr.DATA_TEST1E:3> - " - " -
- SB_PUFFER_EA <Adr.DATA_TEST1E:4> - " - " -
- SB_START-UA880 <Adr.DATA_TEST1E:5> - " - " -
- SB_SUBPORT_DMA <Adr.DATA_TEST1E:6> - " - " -
- SB_IRQST_TEST <Adr.DATA_TEST1E:7> - " - " -
- SB_DMA_SPEZIAL <Adr.DATA_TEST1E:8> - " - " -
```

6. Startprozedur

6.1. Start des Testprogramms

Das Testprogramm TEST1E laeuft unter der Regie von LACS. Nach dem Laden (siehe Punkt 4.) erfolgt der Start:

```
- Aus dem Generierhalt durch das Monitorkommando
  G<CR>
- Restart erfolgt durch das Monitorkommando
  .G <CODE_LACS>:40<CR>
```

6.2. Bedieneraktionen

Der Bediener hat folgende Eingriffsmoeglichkeiten waehrend des Programmablaufs:

- CTRL-S = Unterbrechung der Ausgabe, Fortsetzung mit CTRL-Q.
- CTRL-Q = Fortsetzung einer unterbrochenen Ausgabe.
- CTRL-P = Hardcopy der Bildschirmausgabe auf angeschlossenen Drucker.
Erneutes CTRL-P setzt die Hardcopy zurueck.
- CTRL-C = Uebergang in den Bedienzustand des Monitors.

Die Wirkung dieser Kommandos erfolgt erst unmittelbar vor Ausgabe einer Meldung auf den Bildschirm.

Der Uebergang in den Bedienzustand des Monitors durch Druecken der Taste BREAK auf der Tastatur, ist nicht in jedem Testzeitpunkt moeglich.

7. Einstellungsmoeglichkeiten

Die Standardeinstellung fuer TEST1E lautet:

- TDERRONLY: 0 ; siehe Punkt5.1 LACS-Beschreibung
- TDDEBUG: 3 ; " " " " "
- ROUT TEST1E: FFH ; alle Hauptroutinen generiert
- SB_DIAGNOSE FFH ; \
- SB_PUFFER EA FFH ; |
- SB_START-UA880 FFH ; | Subroutine-Steuerbytes
- SB_SUPPORT DMA FFH ; |
- SB_IRQST TEST FFH ; |
- SB_DMA_SPEZIAL FFH ; /

8. Programmbeschreibung8.1. Programmablauf

Vor dem 1. Durchlauf erfolgt die Ausgabe der Programmnummern, Modulnamen und der Zeichnungs- und Ausgabennummern aller beteiligten Module.

Sie hat fuer TEST1E folgendes Aussehen (siehe Punkt 8.2.1.):

```
TEST1E yy-mm-dd 1.56.703031.4/67-aa-56ACnnn
```

```
(yy-mm-dd = Erstellungsdatum
aa = Ausgabennummer
56ACnnn = Nr. der Aenderungsmittelung).
```

- Danach wird die vom TEST1E benutzte KES-Portadresse auf dem Bildschirm angezeigt (nur im ersten Durchlauf).
- Eine RESET-Kommandofolge zum KES setzt den KES (und die AFS, die AFP mit den angeschlossenen Laufwerken) zurueck. Im System- speicher werden die KES-Kommunikationsbloেকে generiert
- Die Versionen der installierten Firmwarekomponenten (KES, AFS, AFP) werden komplex ermittelt (durch ein modifiziertes Initia-

lisierkommando) und im ersten Testdurchlauf angezeigt.

Es folgen die Hauptroutinen mit ihren Subroutinen:

8.1.1. Hauptroutine KES RESET

Mit speziellen RESET-Kommandofolgen wird das Ruecksetzen des KES gepueft.

8.1.2. Hauptroutine DIAGNOSE

Tests mit dem Funktionskommando "Diagnose" (OFH). Zuerst Pruefung der Teilfunktionen des Funktionskommandos durch Modifizierung, danach zusaetzliche Kontrolle mit dem Funktionskommando "Status-transfer" (O1H).

1. Subroutine: Ueberpruefung aller verfuegbaren Firmware-PROM's mit einem CRC-Verfahren (Kontrollsummentest).
2. Subroutine: Pruefung des gesamten SRAM des KES mit einem Marchingtest.
3. Subroutine: Pruefung des CTC-Schaltkreises mit 3 speziellen Tests:
 - Test1: Zaehler gestaffelt, Reihenfolge 0,1,2,3, mit Vorteiler 16
 - Test2: Zaehler gleichzeitig, Prioritaet 0,1,2,3, mit Vorteiler 16
 - Test3: Zaehler gestaffelt, Reihenfolge 3,2,1,0, mit Vorteiler 256.
4. Subroutine: Pruefung des DMA-Schaltkreises durch DMA-Uebertragung innerhalb des KES mit 4 Tests:
 - Test1: 4KB-Transfer von den Firmware-PROM's in den SRAM
 - Test2: 2KB-Umspeicherung innerhalb des SRAM
 - Test3: 2-Byte-Transfer vom PROM zum E/A-Port (CTC-Kanal3)
 - Test4: 96-Byte-Transfer vom E/A-Port (CTC-Kanal3) zum SRAM.
5. Subroutine: Durch Kommandomodifizierung automatisierte Aufeinanderfolge der in den 4 Subroutinen gepueften Teilfunktionen des Diagnosekommandos.
6. Subroutine: Kontrollfunktion durch Uebertragung des Statuspuffers aus dem KES in den Systemspeicher (Funktionskommando O1H) und Test auf Transfermenge und Pufferinhalt.

8.1.3. Hauptroutine PUFFER EA

Tests mit dem Funktionskommando "KES-Puffer-Ein/Ausgabe" (OEH), d.h. Datentransfers zwischen KES und Systemspeicher. Systematische Erzeugung bestimmter Grenzfaelle u.a. zur Ueberpruefung der Firmware-Handlungen: Variationen der Transfermenge und -richtung mit automatisierter Transferunterbrechung wegen MAP-Aktualisierung (16K Speichergrenze), verbunden mit automatisiertem Uebergang von CPU-Datentransfer zu DMA-Datentransfer der Teilmengen und mit ausgewaehlten Positionsvarianten der Transferdaten im KES-Puffer.

6.2. Bedieneraktionen

Der Bediener hat folgende Eingriffsmoeglichkeiten waehrend des Programmablaufs:

- CTRL-S = Unterbrechung der Ausgabe, Fortsetzung mit CTRL-Q.
- CTRL-Q = Fortsetzung einer unterbrochenen Ausgabe.
- CTRL-P = Harcopy der Bildschirmausgabe auf angeschlossenen Drucker.

Erneutes CTRL-P setzt die Harcopy zurueck.

- CTRL-C = Uebergang in den Bedienzustand des Monitors.

Die Wirkung dieser Kommandos erfolgt erst unmittelbar vor Ausgabe einer Meldung auf den Bildschirm.

Der Uebergang in den Bedienzustand des Monitors durch Druecken der Taste BREAK auf der Tastatur, ist nicht in jedem Testzeitpunkt moeglich.

7. Einstellungsmoeglichkeiten

Die Standardeinstellung fuer TEST1E lautet:

- TDERRONLY: 0 ; siehe Punkt5.1 LACS-Beschreibung
- TDDEBUG: 3 ; " " " " "
- ROUT TEST1E: FFH ; alle Hauptroutrinen generiert
- SB_DIAGNOSE FFH ; \
- SB_PUFFER BA FFH ; |
- SB_START-UA880 FFH ; | Subroutine-Steuerbytes
- SB_SUPPORT DMA FFH ; |
- SB_IRQST TEST FFH ; |
- SB_DMA_SPEZIAL FFH ; /

8. Programmbeschreibung8.1. Programmablauf

Vor dem 1. Durchlauf erfolgt die Ausgabe der Programmnummern, Modulnamen und der Zeichnungs- und Ausgabennummern aller beteiligten Module.

Sie hat fuer TEST1E folgendes Aussehen (siehe Punkt 8.2.1.):

```
TEST1E yy-mm-dd 1.56.703031.4/67-aa-56ACnnn
```

(yy-mm-dd = Erstellungsdatum

aa = Ausgabennummer

56ACnnn = Nr. der Aenderungsmittelung).

- Danach wird die vom TEST1E benutzte KES-Portadresse auf dem Bildschirm angezeigt (nur im ersten Durchlauf).
- Eine RESET-Kommandofolge zum KES setzt den KES (und die AFS, die AFP mit den angeschlossenen Laufwerken) zurueck. Im System-speicher werden die KES-Kommunikationsbloেকে generiert
- Die Versionen der installierten Firmwarekomponenten (KES, AFS, AFP) werden komplex ermittelt (durch ein modifiziertes Initia-

lisierkommando) und im ersten Testdurchlauf angezeigt.

Es folgen die Hauptroutinen mit ihren Subroutinen:

8.1.1. Hauptroutine KES RESET

Mit speziellen RESET-Kommandofolgen wird das Ruecksetzen des KES geprueft.

8.1.2. Hauptroutine DIAGNOSE

Tests mit dem Funktionskommando "Diagnose" (OFH). Zuerst Pruefung der Teilfunktionen des Funktionskommandos durch Modifizierung, danach zusaetzliche Kontrolle mit dem Funktionskommando "Status-transfer" (O1H).

1. Subroutine: Ueberpruefung aller verfuegbaren Firmware-PROM's mit einem CRC-Verfahren (Kontrollsummentest).
2. Subroutine: Pruefung des gesamten SRAM des KES mit einem Marchingtest.
3. Subroutine: Pruefung des CTC-Schaltkreises mit 3 speziellen Tests:
 - Test1: Zaehler gestaffelt, Reihenfolge 0,1,2,3, mit Vorteiler 16
 - Test2: Zaehler gleichzeitig, Prioritaet 0,1,2,3, mit Vorteiler 16
 - Test3: Zaehler gestaffelt, Reihenfolge 3,2,1,0, mit Vorteiler 256.
4. Subroutine: Pruefung des DMA-Schaltkreises durch DMA-Uebertragung innerhalb des KES mit 4 Tests:
 - Test1: 4KB-Transfer von den Firmware-PROM's in den SRAM
 - Test2: 2KB-Umspeicherung innerhalb des SRAM
 - Test3: 2-Byte-Transfer vom PROM zum E/A-Port (CTC-Kanal3)
 - Test4: 96-Byte-Transfer vom E/A-Port (CTC-Kanal3) zum SRAM.
5. Subroutine: Durch Kommandomodifizierung automatisierte Aufeinanderfolge der in den 4 Subroutinen geprueften Teilfunktionen des Diagnosekommandos.
6. Subroutine: Kontrollfunktion durch Uebertragung des Statuspuffers aus dem KES in den Systemspeicher (Funktionskommando O1H) und Test auf Transfermenge und Pufferinhalt.

8.1.3. Hauptroutine PUFFER EA

Tests mit dem Funktionskommando "KES-Puffer-Ein/Ausgabe" (OEH), d.h. Datentransfers zwischen KES und Systemspeicher. Systematische Erzeugung bestimmter Grenzfaelle u.a. zur Ueberpruefung der Firmware-Handlungen: Variationen der Transfermenge und -richtung mit automatisierter Transferunterbrechung wegen MAP-Aktualisierung (16K Speichergrenze), verbunden mit automatisiertem Uebergang von CPU-Datentransfer zu DMA-Datentransfer der Teilmengen und mit ausgewaehlten Positionsvarianten der Transferdaten im KES-Puffer.

1. Subroutine: CPU-/CPU-Datentransfers an einer 16K-Grenze
2. Subroutine: CPU-/DMA- " - " - " - " - "
3. Subroutine: DMA-/DMA- " - " - " - " - "
4. Subroutine: max. zulaessiger DMA-Datentransfertest 1 (4,5KB)
5. Subroutine: max. zulaessiger DMA-Datentransfertest 2 (4,5KB)
6. Subroutine: Fehlerprovokationen bezueglich Puffergrenzen und Datenmenge.

8.1.4. Hauptroutine START UA880

Tests mit dem Funktionskommando "Start-UA880-E/A-Progr." (OCH). Nach den Prinziptests des Funktionskommandos, die eine Testung der UA880-Befehle NOP, RET und HALT beinhalten, folgt ein umfangreicher Befelstest fuer die KES-ZVE.

1. Subroutine: NOP-Programm mit RET-Abschluss in den KES-RAM laden und starten.
2. Subroutine: HALT-Befehl im KES-RAM testen.
3. Subroutine: NOP-Programm mit RET-Abschluss in den Systemspeicher an eine 16K-Adressgrenze laden und als KES-externes UA880-(Window-)Programm starten.
4. Subroutine: UA880-HALT-Befehl im Systemspeicher testen.
5. Subroutine: UA880-Befelstest fuer die KES-ZVE (9,5kB) als KES-externes (Window-)Programm (mit vorherigem Laden an eine 16K-Adressgrenze im Systemspeicher).
 Programmliste des UA880-Befelstests siehe Z80INS 1.56.703056.4/67-00.
 Beschreibung (und Fehlertabellen) des Befelstests siehe Z80DOC 1.56.703057.2/67-00.

8.1.5. Hauptroutine SUB PORT DMA

Tests mit dem Funktionskommando ODH; d.h. Datentransfers zwischen dem Systemspeicher und einem I/O-Port am KES-lokalen BUS (hier I/O-Port des KES-CTC-Schaltkreises).

1. Subroutine: Zaehlkonstante in den CTC (Kanal 0) eintragen und wieder lesen; 1-Byte-Transfers.
2. Subroutine: wie 1. Subroutine aber mit 3-Byte-Transfer.

8.1.6. Hauptroutine IRQST TEST

KES-Interrupt-Tests

1. Subroutine: Interrupt-Test (KES-Kanal 1) mit Ueberpruefung der PIC-Interruptebene 5.

8.1.7. Hauptroutine DMA SPEZIAL

Spezielle Tests, die waehrend der Zeit von KES-DMA-Uebertragungen durchgefuehrt werden.

1. Subroutine: Stringoperationen mit der ZVE K2771 im ZPS-Bereich waehrend DMA-Uebertragungen zwischen KES und ZPS.

2. Subroutine: Zugriffstest zu den EA-Registern der ZVE K2771 während der DMA-Uebertragung mit dem KES.

Das Verlassen des TEST1E erfolgt im fehlerfreien Fall durch den RET-FAR-Befehl mit AX = 1
 Im Fehlerfall wird der Bediener mit Fehlerausschriften informiert und das Testprogramm wird durch einen RET-FAR-Befehl mit AX = 0 verlassen.
 Abbruchmöglichkeit des Testprogramms TEST1E siehe Punkt 10.

8.2. Anschlussbedingungen

8.2.1. Struktur des Quellprogramms

Der Testmodul TEST1E ist folgendermassen aufgebaut:

1. #TITLE(TEST1E Zeichnungnr.-Ausgabenr.-Aend.Mitt.Nr.)
2. Die Programmbeschreibung nach dem Muster der vorliegenden Beschreibung des Leitprogramms LACS.
3. NAME TEST1E
4. Es folgen die EXTRN-Erklärungen.
5. Es folgt das Datensegment mit der Bezeichnung DATA TEST1E.
6. Es folgt das Codesegment mit der Bezeichnung CODE TEST1E.
7. PUBLIC-Erklärungen von Symbolen stehen am Anfang von dem Segment, in dem die betreffenden Symbole definiert sind.

Das Datensegment DATA TEST1E beginnt mit dem Hauptroutine-Steuerwort ROUT TEST1E, es folgen die Subroutine-Steuerbytes; nach weiteren Merkwerten folgen die Anfangsdaten fuer die Kommunikationsblöcke, der Puffer fuer die aktuellen Kommunikationsblöcke, der Statuspuffer, die Puffer fuer Initialisierungsdaten, fuer CTC-Testdaten und fuer Ausgabertexte. Es folgen die UA880-Programme (Programmliste fuer UA880-ZVE-Test siehe Z80INS 1.56.703056.4/67 und Programmbeschreibung mit Fehlertabellen siehe Z80DOC 1.56.703057.2/67).
 Anschliessend folgen die Texte fuer Fehlerausschriften und fuer Bedienerhinweise.

Im Codesegment CODE TEST1E stehen nach den PUBLIC-Erklärungen und den Symboldefinitionen folgende Daten:

```
VERSION_TEST1E DB 'TEST1E '
                DB 'yy-mm-dd '
                DB 'Zeich.Nr.-Ausgabenr.-Aend.Nr. ',OAH,ODH,0
                ORG 40H
```

dabei sind:

yy-mm-dd Datum der letzten Aenderung
 Die naechste Zeile nach dem ORG-Befehl enthaelt die Ansprungstelle des Testmoduls mit der Marke TEST1E.

Das Programm gliedert sich anfangs in den Aufruf von zwei Hauptkomplexen: VORBEREITUNG und eigentlicher KES_TEST. Diese werden mit nachfolgenden Uebersichtsroutinen feiner gegliedert.

8.2.2. Stack- und Registernutzung

Vom Testmodul TEST1E wird der Stack des Leitprogramms genutzt, d.h. beim Start von TEST1E wird nur DS initialisiert.

8.2.3. Protokollsteuerung

Die Ausgaben sind je nach den Werten, die die Variablen TDERRONLY und TDDEBUG enthalten, abgestuft unterdrueckbar (s.Pkt.5.1.). Es gelten die Richtlinien gemaess Punkt 8.2.3 der LACS-Beschreibung.

8.2.4. Nutzbare TEST1E-Routinen

TEST1E.OBJ enthaelt keine Routinen die fuer andere Nutzer verfuegbar sind.

8.2.5. Nutzbarkeit anderer Routinen

Siehe Punkt 3.4.

9. Fehler

9.1. Fehlerausgaben

Das Programm TEST1E gibt im Falle eines erkannten Fehlers einen Text aus und dazu den Instructionpointer, der auf die fehlerfindende Programmstelle hinweist. Im allgemeinen folgen weitere Fehleranalysen mit fehlereingrenzenden Textausgaben bzw. mit der Ausgabe der Kommunikationsbloecke, evtl. mit nachfolgender Ausgabe des Statuspuffers (Bedienanalyse erforderlich).

9.2. Fortsetzung nach Fehler

Nach der Fehlerausgabe erfolgt der Uebergang zum Leitprogramm. Dieses erhoehrt seinen Fehlerzaehler und geht zum Monitor. Die Wiederholung des Tests erfolgt durch das Monitorkommando *G<CR>.

10. Verschiedenes

10.1. Programmabbruchmoeglichkeiten

Der Programmablauf kann durch die BREAK-Taste bedingt unterbrochen werden (siehe Punkt 6.3.). Weiterstart mit G<CR> ist erlaubt. Restart auf Adresse CODE LACS:40 ist moeglich. Ein geordneter Abbruch kann durch Eingabe von CTRL-C erfolgen, damit erfolgt der Uebergang in den Monitorbedienzustand vor der naechsten Ausgabe.

10.2. Kontrolle der Programmdurchfuehrung

Die Durchfuehrung des Testprogramms wird durch die folgende LACS-Meldung auf dem Bildschirm bestaetigt (Standardfall):

```
TEST1E: KES TEST          "PASSED"
```

Fehler- und Durchlaufzaehler werden in folgender Form ausgegeben:

```
*****  ERRORS / CYCLES :  xxxx / yyyy
```

Danach beginnt ein neuer Programmzyklus.

Nach Unterbrechung des Programms kann die Durchfuehrungszahl des Testprogramms TEST1E auch aus der Zelle CONST_LACS:1B0 ermittelt werden (Monitorkommando DW).

Die Anzahl der aufgetretenen Fehlermeldungen kann man der Zelle CONST_LACS:1AE entnehmen.

TEST1E.

Betriebsdokumentation A 7100, Bd.3