IBV11,IB11 IBV11-A DIAGNOSTIC AH-T619B-MC OCT 1983
CVIBABO FICHE 1 OF 1

IDENTIFICATION

Product Code: AC-A880B-MC

Product Name: CVIBAB0 IBV11-A DIAG

Date : FEB 1983

Maintainer: Ray Shoop

TABLE OF CONTENTS

The "B" revision allow the operator to indicate CPU type.
The type of CPU is needed because signal timing is performed.
The "B" revision includes provisions for the "ER1-INHIBIT" (S1-8) switch to be tested.

1.0 ABSTRACT

This program allows the user to check-out a greater portion of logic on this option if a second IBV11-A is available. See section 2.1. When a second IBV11-A can be obtained, the user must inform this diagnostic to exercise the logic on one IBV11-A that requires a KGM (Known Good Module). Please note that the second IBV11-A should be known good. When a KGM is available, the "ER1-INHIBIT SWITCH" on the module under test should be CLEARED (OPEN-OFF). No attempt is made to checkout the KGM and no conclusion that if good passes are made through this diagnostic that the KGM is also good. Signals "SRQ", "ER1", "BIAKI" and "ER1IHB" are not tested on the IBV11-A if a KGM is not used.

If the user is unfamilar with an LSI-11 he should review sections 8.4 and 8.5. A software switch egister is included with this program.

Every effort was made to make this program conform to LSI-11 programming restrictions. However, the user should read sections 7.1 and 7.2.

2.0 REQUIREMENTS

2.1 Equipment

1. PDP-11 Family Computer with 4K of memory (or more) and console I/O facilities (i.e., TTY).

2. IBV11-A under test.

3. (Optional) Second IBV11-A "KGM" (known good module). The "KGM" must have an intrumentation Bus Cable between it and the first IBV-11. Its base address should be 760160 and vector address of 660 (see section 3.2 if different).

NOTE

While it is generally recommended that a "KGM" is used, if one is available, deposit a "000001" into location "$CDW1". No test will be performed that requires the "KGM" if $CDW1 is zero.

2.2 Storage

This program occupies and uses the lower 4K of memory.

3.0 LOADING PROCEDURE

3.1 Method

Standard procedure for normal binary tapes should be followed.

The program can also be loaded by XXDP, ACT or APT.

3.2 Non-Standard Address, Vector, or Use of Software Switch Register

This program is set to test a IBV11-A with a standard address and vector. If any of these are different on the IBV11-A you are testing, change the corresponding location in memory before starting this test.

TAG	ADDRESS	CURRENT CONTENTS	COMMENTS
$BASE:	1244	760150	;;BASE ADDRESS OF EQUIPMENT ;; UNDER TEST
$VECT1:	1240	000420	;;INTERRUPT VECTOR #1
SWREG:	176	000000	;;MANUAL SWR.
IBS2:	1402	760160	; ADDRESS OF SECOND IBV11-A.
VECTA2:	1406	000660	; VECTOR ADDRESS OF SECOND IBV11-A.
$CDW1:	1250	000000	;;DEVICE DESCRIPTOR WORD #1 (if = 000001 to use 'KGM'' in testing 1st IBV-11) (Default = 00000 to disable use of KGM in tests.
$CPUOP:	1216	000010	;;TYPE OF CPU CODE (DEFAULT = 00010 INDICATE PDP-11/03) (REF. SECTION 5.2 FOR LIST OF CODES)

4.0 STARTING PROCEDURE

4.1 Control Switch Setting

Before starting the diagnostic, set all switch register bits as desired. See section 5.1.

4.2 Starting Addresses

200 Start of Logic Tests

4.3 Program and/or Operator Action

1. Load program into memory.
2. Enter keyboard "ODT".
3. Alter location "SWREG" to reflect desired options of a switch register - See section 5.1.
4. Alter location "$CPUOP" to reflect current cpu type code - See section 5.2.
5. Type starting address, followed by "G" to start program.

5.0 OPERATING PROCEDURE

5.1 Switch Register Function

SWR BIT	OCTAL	FUNCTION WHEN SET
15	100000	HALT ON ERROR
14	040000	LOOP ON TEST
13	020000	INHIBIT ERROR TYPEOUT
11	004000	INHIBIT ITERATIONS (SHORT PASS)
10	002000	INDICATE "ER1-INHIBIT" SWITCH (S1-8) IS SET (CLOSED-ON)
09	001000	LOOP ON ERROR
08	000400	LOOP ON TEST IN SWR <7:0>

NOTE

The Switch Register may be changed at any time while the diagnostic is running by typing "G".

5.2 Cpu Type Code Function

01	PDP-11/04
02	PDP-11/05
03	PDP-11/20
04	PDP-11/40
05	PDP-11/45
06	PDP-11/70
07	PDP-11/34
10	PDP-11/03 <DEFAULT>
11	PDP-11/
12	PDP-11/
13	PDP-11/
14	PDP-11/
15	PDP-11/
16	PDP-11/
17	PDP-11/

5.3 Scope Loops

If an error occurs and the user wishes to scope the error, "SWREG" should be altered to "100000" at the start of the test to halt on error, then when the program halts on error and the CPU enters "ODT", "$SWREG" should be altered to "060000" to loop on current test and inhibit error typeout, then type "P" to continue program execution.

5.4 Program and/or Operator Action

1. When the program is initially started it will type:

   CVIBA-B

   SWR=000000 NEW=

2. Program now waits for the operator to enter a switch register setting (see section 8.6). To change the switch register setting, see section 8.6.

3. Program executes first pass of logic tests, subtest iterations inhibited.

4. Program reports any errors it detects.

5. Program reports 'END PASS 1".

6. Program executes second pass of logic tests, only this time it will loop on each test 2000 times.

7. Program the reports 'END PASS 2".

8. Program will continue executing steps 6 and 7 until stoped.

6.0 ERRORS

6.1 Error Printout

Printout varies with the error detected. The error PC typed out is the actual location of the error call.

6.2 Non-Standard Error Halt

Bus errors will cause a Halt in the routine "IOTRD". The address that caused this trap will be in "TRTO".

7.0 RESTRICTIONS

7.1 Starting Restriction

If a free-running clock, such as 60Hz from the power supply, is attached to the "BEVNT" bus line on both REV level C/D and E systems, an interrupt to location 100 will occur when using the "G" and "L" commands prior to executing the first instruction. Therefore this program can not disable the BEVNT bus line by inhibiting interrupts.

User systems requiring a free-running clock attached to the BEVNT bus line can temporarily avoid this situation by setting the PSW(RS) to 200, loading the PC with the starting address instead of using the "G" command, and then using the "P" command. Before using the "L" command, the PSW(RS) can be set to 200, thereby inhibiting interrupts, to avoid receiving the event interrupt after loading the ABS loader.

7.2 Possible Program "BOMBS"

The first two tests of this program check to see if the IBV11-A responds to the address the program thinks its at. If the IBV11-A does not respond, a bus error occurs.

For more information on the next subject, see JAN. 1976 LSI-11 engineering bulletin issued by the Digital Components Group.

Bus errors may alter the preset contents of locaton 4 before the trap is executed, thereby transferring program control to area in the program that was not set up to handle the trap. If this happens, the program will "BOMB" and possibly rewrite parts of itself.

7.3 Customer Devices

No customer devices should be connected to the unit under test while executing the program.

8.0 MISCELLANEOUS

8.1 Power Fail

After a power failure occurs, the program execution will continue at the point where the power occurred. The program will type "POWER".

8.2 XXDP, ACT, APT

The program is chainable under XXDP, ACT, or APT. Although "APT HOOKS" have been installed, they have not been tested.

8.3 Execution Time

PDP-11/03 CPU WITH ONLY 1 BOARD

0.1 Minutes (6 sec) Iteration Inhibited - No Errors
0.5 Minutes (30 sec) With Iterations - No Errors

PDP-11/03 CPU WITH KGM AND 1 BOARD

1.4 Minutes (80 sec) With Interations - No Errors

8.4 LSI-11 "ODT" Commands

FORMAT	DESCRIPTION
<CR> RETURN	Close opened location and accept next command.
<LF> LINE FEED	Close current location; open next sequential location.
^(UPARROW)	Open previous location.
< (LEFT ARROW)	Take contents of opened location, indexed by contents of PC, and open that location.
@	Take contents of opened location as absolute address and open that location.
R/	Open the word at location R.
/	Reopen the last location.
$N/ or RN/	Open general register N(0-7) or S(PS register).
R:G or RG	GOTO location R and start program.
NL	Execute Bootstrap loader using N as device CSR. Console device is 177560.
:P or P	Proceed with program execution.
RUBOUT	Erases previous numeric character. Response is a backslash ().

8.5 Entering LSI-11 "ODT"

The halt or ODT microcode state of the KD11F (LSI-11 module) can be entered in five different ways (others are a subset of these) from the run state:

1. Execution of a LSI-11 halt instruction.
2. A double bus error.
3. As a power up option.
4. ASCII break with DLV11 framing error asserting the B halt line (enabled by jumper of DLV11).

Upon entering the halt state, the KD11F responds through the set of command listed in section 8.4.

8.6 Use of Program Software SWR

The software switch register may be changed by typing ^G (control and letter G keys typed simultaneously). When ^G is typed, the program responds by typing "SWR=XXXXXX" where XXXXXX equals the former contents of the switch register.

If you wish to keep the current value, type <CR>. If you wish to change the value, type the new value followed by a <CR>.

It is important to note that the diagnostic is not running after the ^G until a <CR> is typed.

8.7 Trap Catcher

The Trap Catcher in this diagnostic employs a new concept. This concept will enable the user of this diagnostic to gain more knowledge of the events that lead the program to this area.

The Trap Catch consists of PC+2 and JSR PC,R0. (i.e., Location 300 would contain 302 and location 302 would contain 4700).

When a device interrupts to the Trap Catcher, it would pick up the PC+2 of the trap as an address of the interrupt service routine.

The program would then pick up "4700" as the new PSW. Bit 7 of the new PSW having been set, would cause further interrupts from happening. When the CPU attempts to execute "4700" (JSR PC,R0), a Bus-time-out trap will occur to location 4. Location 4 contains a pointer to "IOTRD", a routine that will report the trap as an error.

To guard against "Real" Bus errors routing us through location 4 to "IOTRD", we check to see if the trap that brought us to location 4 really came from the Trap Catcher area. If not we'll halt and leave the Trap Address in "TRTO".

More about the interrupt error can be found in the description of the error in the program listing in the routine "IOTRD".

CVIBA-B MACY11 30G(1063) 25-FEB-83 08:23
CVIBAB.P11   13-DEC-82 09:29 TABLE OF CONTENTS SEQ 0013

```
                                        .NLIST  MC,MD,CND
                                        .LIST   ME
                                        .ENABL  ABS
                                        .ENABL  AMA
000000  165400                          $SWR=165400

                                .TITLE  CVIBA-B
                                ;*COPYRIGHT (C) 1983
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY EDWARD C. BADGER MOD BY R. SHOOP
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.
                                ;*
000001  000001                  $TN=1

                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            INDICATE 'EH1-INHIBIT" SWITCH IS SET (CLOSED-ON)
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>

                                        .SBTTL  TRAP CATCHER

000000  000000                          .=0
                                        ;*ALL UNUSED LOCATIONS FROM 4-776 CONTAIN A ".+2"
                                        ;*AND "JSR PC,R0" SEQUENCE TO CATCH ILLEGAL INTERRUPTS,
                                        ;*AND INTERRUPTS TO THE WRONG VECTOR.
                                        ;*LOCATION 0 CONTAINS A 0 TO CATCH IMPROPERLY LOADED
                                        ;*VECTORS
000004  000004                          .=4
000004  013126  000200                  .WORD   IOTRD,200               ;HANDLE BUSS ERROR.
000140  000140                          .=140
000140  170000  000300                  170000,300                      ;FALCON CPU FIX
000174  000174                          .=174
000174  000000                  DISPREG:.WORD   0               ;;SOFTWARE DISPLAY REGISTER.
000176  000000                  SWREG:  .WORD   0               ;;SOFTWARE SWITCH REGISTER.
000100  000100                          .=100
000100  000104  000200  000002          .WORD   104,200,2       ;IF "B EVENT" ON Q-BUS IS
                                                                ;CONNECTED, WE NEED A WAY OF
000200  000200                          .=200                   ;IGNORING ITS INTERRUPTS.
000200  000137  001460                  JMP     START

                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100  001100                  STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
```

```
CVIBA-B MACY11 30G(1063)  25-FEB-83  08:23  PAGE 1-1                                                    SEQ 0015
CVIBAB.P11    13-DEC-82 09:29          BASIC DEFINITIONS

(1)                                    .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
(1)
(1)                                    ;*MISCELLANEOUS DEFINITIONS
(1)             000011                 HT=     11              ;;CODE FOR HORIZONTAL TAB
(1)             000012                 LF=     12              ;;CODE FOR LINE FEED
(1)             000015                 CR=     15              ;;CODE FOR CARRIAGE RETURN
(1)             000200                 CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
(1)             177776                 PS=     177776          ;;PROCESSOR STATUS WORD
(1)                                    .EQUIV  PS,PSW
(1)             177774                 STKLMT= 177774          ;;STACK LIMIT REGISTER
(1)             177772                 PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
(1)             177570                 DSWR=   177570          ;;HARDWARE SWITCH REGISTER
(1)             177570                 DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
(1)
(1)                                    ;*GENERAL PURPOSE REGISTER DEFINITIONS
(1)             000000                 R0=     %0              ;;GENERAL REGISTER
(1)             000001                 R1=     %1              ;;GENERAL REGISTER
(1)             000002                 R2=     %2              ;;GENERAL REGISTER
(1)             000003                 R3=     %3              ;;GENERAL REGISTER
(1)             000004                 R4=     %4              ;;GENERAL REGISTER
(1)             000005                 R5=     %5              ;;GENERAL REGISTER
(1)             000006                 R6=     %6              ;;GENERAL REGISTER
(1)             000007                 R7=     %7              ;;GENERAL REGISTER
(1)             000006                 SP=     %6              ;;STACK POINTER
(1)             000007                 PC=     %7              ;;PROGRAM COUNTER
(1)
(1)                                    ;*PRIORITY LEVEL DEFINITIONS
(1)             000000                 PR0=    0               ;;PRIORITY LEVEL 0
(1)             000040                 PR1=    40              ;;PRIORITY LEVEL 1
(1)             000100                 PR2=    100             ;;PRIORITY LEVEL 2
(1)             000140                 PR3=    140             ;;PRIORITY LEVEL 3
(1)             000200                 PR4=    200             ;;PRIORITY LEVEL 4
(1)             000240                 PR5=    240             ;;PRIORITY LEVEL 5
(1)             000300                 PR6=    300             ;;PRIORITY LEVEL 6
(1)             000340                 PR7=    340             ;;PRIORITY LEVEL 7
(1)
(1)                                    ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
(1)             100000                 SW15=   100000
(1)             040000                 SW14=   40000
(1)             020000                 SW13=   20000
(1)             010000                 SW12=   10000
(1)             004000                 SW11=   4000
(1)             002000                 SW10=   2000
(1)             001000                 SW09=   1000
(1)             000400                 SW08=   400
(1)             000200                 SW07=   200
(1)             000100                 SW06=   100
(1)             000040                 SW05=   40
(1)             000020                 SW04=   20
(1)             000010                 SW03=   10
(1)             000004                 SW02=   4
(1)             000002                 SW01=   2
(1)             000001                 SW00=   1
(1)                                    .EQUIV  SW09,SW9
(1)                                    .EQUIV  SW08,SW8
(1)                                    .EQUIV  SW07,SW7
```

```
(1)                                     .EQUIV  SW06,SW6
(1)                                     .EQUIV  SW05,SW5
(1)                                     .EQUIV  SW04,SW4
(1)                                     .EQUIV  SW03,SW3
(1)                                     .EQUIV  SW02,SW2
(1)                                     .EQUIV  SW01,SW1
(1)                                     .EQUIV  SW00,SW0
(1)
(1)                                     ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)             100000                  BIT15=  100000
(1)             040000                  BIT14=  40000
(1)             020000                  BIT13=  20000
(1)             010000                  BIT12=  10000
(1)             004000                  BIT11=  4000
(1)             002000                  BIT10=  2000
(1)             001000                  BIT09=  1000
(1)             000400                  BIT08=  400
(1)             000200                  BIT07=  200
(1)             000100                  BIT06=  100
(1)             000040                  BIT05=  40
(1)             000020                  BIT04=  20
(1)             000010                  BIT03=  10
(1)             000004                  BIT02=  4
(1)             000002                  BIT01=  2
(1)             000001                  BIT00=  1
(1)                                     .EQUIV  BIT09,BIT9
(1)                                     .EQUIV  BIT08,BIT8
(1)                                     .EQUIV  BIT07,BIT7
(1)                                     .EQUIV  BIT06,BIT6
(1)                                     .EQUIV  BIT05,BIT5
(1)                                     .EQUIV  BIT04,BIT4
(1)                                     .EQUIV  BIT03,BIT3
(1)                                     .EQUIV  BIT02,BIT2
(1)                                     .EQUIV  BIT01,BIT1
(1)                                     .EQUIV  BIT00,BIT0
(1)
(1)                                     ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1)             000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
(1)             000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1)             000014                  TBITVEC=14              ;;"T" BIT
(1)             000014                  TRTVEC= 14              ;;TRACE TRAP
(1)             000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
(1)             000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)             000024                  PWRVEC= 24              ;;POWER FAIL
(1)             000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
(1)             000034                  TRAPVEC=34              ;;'TRAP' TRAP
(1)             000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
(1)             000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
(1)             000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
 47
 48             160150                          ABASE=  160150
 49             000420                          AVECT1= 420
 50             000200                          APRIOR= 200
 51             000010                          ACPUOP= 10
 52             000001                          $TN=1
 53
```

```
     54                                 .SBTTL  ACT11 HOOKS
    (1)
    (2)                                 ;;******************************************************************
    (1)                                 ;HOOKS REQUIRED BY ACT11
    (1)           000204                        $SVPC=.                 ;SAVE PC
    (1)           000046                        .=46
    (1)  000046   010044                        $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
    (1)           000052                        .=52
    (1)  000052   000000                        .WORD   0               ;;2)SET LOC.52 TO ZERO
    (1)           000204                        .=$SVPC                 ;; RESTORE PC
     55
     56           001000                        .=1000
     57
     58                                 .SBTTL  APT PARAMETER BLOCK
    (1)
    (2)                                 ;;******************************************************************
    (1)                                 ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
    (2)                                 ;;******************************************************************
    (1)           001000                        .$X=.   ;;SAVE CURRENT LOCATION
    (1)           000024                        .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
    (1)  000024   000200                        200     ;;FOR APT START UP
    (1)           000044                        .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
    (1)  000044   001000                        $APTHDR ;;POINT TO APT HEADER BLOCK
    (1)           001000                        .=.$X   ;;RESET LOCATION COUNTER
    (2)                                 ;;******************************************************************
    (1)                                 ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
    (1)                                 ;INTERFACE SPEC.
    (1)
    (1)  001000                         $APTHD:
    (1)  001000   000000                $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
    (1)  001002   001170                $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
    (1)  001004   000074                $TSTM:  .WORD   60.     ;;RUN TIM OF LONGEST TEST
    (1)  001006   000170                $PASTM: .WORD   120.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
    (1)  001010   000170                $UNITM: .WORD   120.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
    (1)  001012   000031                        .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
     59
```

```
 60
(1)
(2)
(1)
(1)
(1)
(1)          001100
(1)  001100
(1)  001100  000000
(1)  001102     000
(1)  001103     000
(1)  001104  000000
(1)  001106  000000
(1)  001110  000000
(1)  001112  000000
(1)  001114     000
(1)  001115     001
(1)  001116  000000
(1)  001120  000000
(1)  001122  000000
(1)  001124  000000
(1)  001126  000000
(1)  001130  000000
(1)  001132  000000
(1)  001134     000
(1)  001135     000
(1)  001136  000000
(1)  001140  177570
(1)  001142  177570
(1)  001144  177560
(1)  001146  177562
(1)  001150  177564
(1)  001152  177566
(1)  001154     000
(1)  001155     002
(1)  001156     012
(1)  001157     000
(1)  001160  000000
(1)  001162  000000
(1)  001164     077
(1)  001165     015
(1)  001166  000012
(2)
(2)
(2)
(3)
(2)
(2)  001170
(2)  001170  000000
(2)  001172  000000
(2)  001174  000000
(2)  001176  000000
(2)  001200  000000
(2)  001202  000000
(2)  001204  000000
(2)  001206  000000
```

```
.SBTTL  COMMON TAGS

;;*****************************************************************
;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
;*USED IN THE PROGRAM.

        .=1100
$CMTAG:                         ;;START OF COMMON TAGS
        .WORD   0
$TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
$ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
$ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
$LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
$LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
$ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
$ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
$_RMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
$ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
$GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
$BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
$GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
$BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
        .WORD   0               ;;RESERVED--NOT TO BE USED
        .WORD   0
$AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
$INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
        .WORD   0
SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
$TKS:   177560                  ;;TTY KBD STATUS
$TKB:   177562                  ;;TTY KBD BUFFER
$TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
$TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
$NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
$FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
$FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
$TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
$TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
$ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
$QUES:  .ASCII  /?/             ;;QUESTION MARK
$CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
$LF:    .ASCIZ  <12>            ;;LINE FEED
;;*****************************************************************
.SBTTL  APT MAILBOX-ETABLE

;;*****************************************************************
.EVEN
$MAIL:                          ;;APT MAILBOX
$MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
$FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
$TESTN: .WORD   ATESTN          ;;TEST NUMBER
$PASS:  .WORD   APASS           ;;PASS COUNT
$DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
$UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
$MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
$MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
```

```
(2)  001210                          $ETABLE:                ;;APT ENVIRONMENT TABLE
(2)  001210      000                 $ENV:    .BYTE  AENV    ;;ENVIRONMENT BYTE
(2)  001211      000                 $ENVM:   .BYTE  AENVM   ;;ENVIRONMENT MODE BITS
(2)  001212  000000                  $SWREG:  .WORD  ASWREG  ;;APT SWITCH REGISTER
(2)  001214  000000                  $USWR:   .WORD  AUSWR   ;;USER SWITCHES
(2)  001216  000010                  $CPUOP:  .WORD  ACPUOP  ;;CPU TYPE,OPTIONS
(2)                                  :*                      BITS 15-11=CPU TYPE
(2)                                  :*                             11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(2)                                  :*                             11/70=06,PDQ=07,Q=10
(2)                                  :*                      BIT 10=REAL TIME CLOCK
(2)                                  :*                      BIT  9=FLOATING POINT PROCESSOR
(2)                                  :*                      BIT  8=MEMORY MANAGEMENT
(2)  001220      000                 $MAMS1:  .BYTE  AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
(2)  001221      000                 $MTYP1:  .BYTE  AMTYP1  ;;MEM. TYPE,BLK#1
(2)                                  :*                      MEM.TYPE BYTE   --  (HIGH BYTE)
(2)                                  :*                             900 NSEC CORE=001
(2)                                  :*                             300 NSEC BIPOLAR=002
(2)                                  :*                             500 NSEC MOS=003
(2)  001222  000000                  $MADR1:  .WORD  AMADR1  ;;HIGH ADDRESS,BLK#1
(2)                                  :*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
(2)  001224      000                 $MAMS2:  .BYTE  AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
(2)  001225      000                 $MTYP2:  .BYTE  AMTYP2  ;;MEM.TYPE,BLK#2
(2)  001226  000000                  $MADR2:  .WORD  AMADR2  ;;MEM.LAST ADDRESS,BLK#2
(2)  001230      000                 $MAMS3:  .BYTE  AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
(2)  001231      000                 $MTYP3:  .BYTE  AMTYP3  ;;MEM.TYPE,BLK#3
(2)  001232  000000                  $MADR3:  .WORD  AMADR3  ;;MEM.LAST ADDRESS,BLK#3
(2)  001234      000                 $MAMS4:  .BYTE  AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
(2)  001235      000                 $MTYP4:  .BYTE  AMTYP4  ;;MEM.TYPE,BLK#4
(2)  001236  000000                  $MADR4:  .WORD  AMADR4  ;;MEM.LAST ADDRESS,BLK#4
(2)  0012[illegible]  000420         $VECT1:  .WORD  AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
(2)  001242  000000                  $VECT2:  .WORD  AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
(2)  0012[illegible]4  160150        $BASE:   .WORD  ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
(2)  001246  000000                  $DEVM:   .WORD  A[illegible]VM  ;;DEVICE MAP
(2)  001250  000000                  $CDW1:   .WORD  ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
(2)  001252                          $ETEND:
(2)                                  .MEXIT
```

SEQ 0020

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT


(1) 001252                      $ERRTB:
 64                                     ;ITEM   1
 65 001252  013256                      EM1                     ;IBS FUNCTION ERROR
 66 001254  014401                      DH1                     ;TEST    ERRPC   IB ADDR
 67 001256  014602                      DT1                     ;$TESTN,$ERRPC,IBS
 68 001260  014652                      DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.
 69                                     ;ITEM   2
 70 001262  013304                      EM2                     ;IBD FUNCTION ERROR
 71 001264  014401                      DH1                     ;TEST    ERRPC   IB ADDR
 72 001266  014602                      DT1                     ;$TESTN,$ERRPC,IBS
 73 001270  014652                      DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.
 74                                     ;ITEM   3
 75 001272  013332                      EM3                     ;IBS DATA ERROR
 76 001274  014447                      DH3                     ;TEST    ERRPC   GOOD    BAD
 77 001276  014616                      DT3                     ;$TESTN,$ERRPC,$GDDAT,$BDDAT
 78 001300  014652                      DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.
 79                                     ;ITEM   4
 80 001302  013354                      EM4                     ;IBD DATA ERROR
 81 001304  014447                      DH3                     ;TEST    ,ERRPC   GOOD    BAD
 82 001306  014616                      DT3                     ;$TESTN,$ERRPC,$GDDAT,$BDDAT
 83 001310  014652                      DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.
 84                                     ;ITEM   5
 85 001312  013376                      EM5                     ;IBS/IBD ADDRESS ERROR
 86 001314  014504                      DH5                     ;TEST   ERROR PC   ADDRESS
 87 001316  014630                      DT5                     ;$TSTNM,$ERRPC,IBS
 88 001320  014652                      DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.
 89                                     ;ITEM   6
 90 001322  013426                      EM6                     ;IBWC/IBCA DATA ERROR
 91 001324  014447                      DH3                     ;TEST    ERRPC   GOOD    BAD
 92 001326  014616                      DT3                     ;$TESTN,$ERRPC,$GDDAT,$BDDAT
 93 001330  014652                      DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.
 94                                     ;ITEM   7
 95 001332  013455                      EM7                     ;INTERRUPT ERROR
 96 001334  014535                      DH7                     ;TEST    ERRPC   TO      FROM ADDR.
 97 001336  014640                      DT7                     ;TSTNM,#ERRPC,TRTO,TRFRO
 98 001340  014652                      DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.
 99                                     ;ITEM   10
100 001342  013477  014447  014616      EM10,DH3,DT3,DF0        ;POSSIBLE -YA VARIATION ERROR
     001350  014652
101                                     ;ITEM   11
102 001352  014023  014447  014616      EM11,DH3,DT3,DF0        ;POSSIBLE 'ER1-INH' SWITCH ERROR
     001360  014652
```

```
                                        .SBTTL  REG ADDRESS AND COMMON TAGS
                                        ;WARNING IF DEVICE # IS AT DIFFERENT ADDRESS OR VECTOR
                                        ;DO NOT PATCH THESE LOCATIONS - SEE PROGRAM DOCUMENTATION.
001362  160150                IBS:    .WORD   ABASE   ;>NO            <;CONTROL AND STATUS REGISTER.
001364  160152                IBD:    .WORD   ABASE+2 ;>PATCHES       <;DATA REGISTER.
001366  160154                IBWC:   .WORD   ABASE+4         ;ADDRESS RESERVED FOR
001370  160156                IBCA:   .WORD   ABASE+6         ;FUTURE USE
001372  000420                VECTA:  .WORD   AVECT1  ;>ALLOWED       <;VECTOR ADDRESS.
001374  000424                VECTB:  .WORD   AVECT1+4 ;>HERE!        <;VECTOR ADDR. +4.
001376  000430                VECTC:  .WORD   AVECT1+10
001400  000434                VECTD:  .WORD   AVECT1+14
001402  160160                IBS2:   .WORD   ABASE+10
001404  160162                IBD2:   .WORD   ABASE+12
001406  000660                VECTA2: .WORD   660
001410  000664                VECTB2: .WORD   664
001412  000670                VECTC2: .WORD   670
001414  000674                VECTD2: .WORD   674

                                        ;VECTOR ADDRESSES +2 LOCATIONS.
001416  000422                PRA:    .WORD   AVECT1+2        ;NOTE:  DO NOT ATTEMPT TO PATCH
001420  000426                PRB:    .WORD   AVECT1+6        ;       THESE LOCATIONS IF A VECTOR
001422  000432                PRC:    .WORD   AVECT1+12       ;       VARYIES .   ALTER LOCATION
001424  000436                PRD:    .WORD   AVECT1+16       ;       '$VECT1:'.
001426  000442                PRA2:   .WORD   AVECT1+22       ;       IF TEST MODULE VECTOR IS
001430  000446                PRB2:   .WORD   AVECT1+26       ;       DIFFERENT,YOU MUST CHANGE
001432  000452                PRC2:   .WORD   AVECT1+32       ;       LOCATION 'VECTA2:'.
001434  000456                PRD2:   .WORD   AVECT1+36

000002                  CPUDLY: 2                       ;25 USEC LOOP FACTOR
    002                 CPUSPD: .BYTE   2               ;0 (PDP-11)
    005                         .BYTE   5               ;1 (PDP-11/04)
    004                         .BYTE   4               ;2 (PDP-11/05)
    002                         .BYTE   2               ;3 (PDP-11/20)
    011                         .BYTE   9.              ;4 (PDP-11/40)
    014                         .BYTE   12.             ;5 (PDP-11/45)
    014                         .BYTE   12.             ;6 (PDP-11/70)
    002                         .BYTE   2               ;7 (PDP-)
    002                         .BYTE   2               ;10 (PDP-11/03) ** DEFAULT VALUE **
    006                         .BYTE   6               ;11 (PDP-11/23)
    006                         .BYTE   6               ;12 (PDP-11/34)
    002                         .BYTE   2               ;13 (PDP-11/)
    002                         .BYTE   2               ;14 (PDP-11/)
    002                         .BYTE   2               ;15 (PDP-11/)
    002                         .BYTE   2               ;16 (PDP-11/)
    002                         .BYTE   2               ;17 (PDP-11/)
```

```
                                        .SBTTL  PROGRAM START


                                START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001460  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001464  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001466  022706  001140                  CMP     #SWR,R6 ;;DONE?
001472  001374                          BNE     .-6             ;;LOOP BACK IF NO
001474  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
001500  012737  010426  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001506  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001514  012737  010120  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001522  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
001530  012737  013176  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001536  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
001544  012737  012750  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
001552  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
001560  005037  001160                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
001564  005037  001162                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
001570  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
001576  012737  001576  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
001604  012737  001604  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
001612  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
001616  012737  001652  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
001624  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
001632  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
001640  022777  177777  177272          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
001646  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
001650  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
001652  012716  001660          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
001656  000002                          RTI
001660  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
001666  012737  000174  001142          MOV     #DISPREG,DISPLAY
001674  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

001700  005037  001176                  CLR     $PASS           ;;CLEAR PASS COUNT
001704  132737  000200  001211          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
001712  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
001714  012737  001212  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
001722                          67$:
001722  012737  000300  000022          MOV     #300,IOTVEC+2   ;LOWER PS TO LEVEL 6 FOR FALCON CPU
001730  012737  000300  000032          MOV     #300,EMTVEC+2   ;       ''      ''      ''
001736  012737  000300  000036          MOV     #300,TRAPVEC+2  ;       ''      ''      ''
001744  012737  000300  000026          MOV     #300,PWRVEC+2   ;       ''      ''      ''
001752  012737  013126  000004          MOV     #IOTRD,ERRVEC   ;SET TO HANDLE BUS ERRORS.
001760  012737  000200  000006          MOV     #200,ERRVEC+2

001766  013737  001244  001362          MOV     $BASE,IBS       ;GET BASE ADDR.
001774  013737  001362  001364          MOV     IBS,IBD         ;FIX DATA BUFFER=
002002  062737  000002  001364          ADD     #2,IBD          ;CSR+2
```

```
   223  002010  013737  001364  001366          MOV     IBD,IBWC
   224  002016  062737  000002  001366          ADD     #2,IBWC
   225  002024  013737  001366  001370          MOV     IBWC,IBCA
   226  002032  062737  000002  001370          ADD     #2,IBCA
   227  002040  013737  001240  001372          MOV     $VECT1,VECTA    ;GET VECTOR ADDR.
   228  002046  042737  170000  001372          BIC     #170000,VECTA   ;STRIP JUNK
   229  002054  013737  001362  014610          MOV     IBS,IBSA
   230  002062  013737  001364  014612          MOV     IBD,IBDA
   231  002070  013737  001402  001404          MOV     IBS2,IBD2
   232  002076  062737  000002  001404          ADD     #2,IBD2
   233  002104  013737  001372  001374          MOV     VECTA,VECTB
   234  002112  062737  000004  001374          ADD     #4,VECTB
   235  002120  013737  001374  001376          MOV     VECTB,VECTC
   236  002126  062737  000004  001376          ADD     #4,VECTC
   237  002134  013737  001376  001400          MOV     VECTC,VECTD
   238  002142  062737  000004  001400          ADD     #4,VECTD
   239  002150  013737  001406  001410          MOV     VECTA2,VECTB2
   240  002156  062737  000004  001410          ADD     #4,VECTB2
   241  002164  013737  001410  001412          MOV     VECTB2,VECTC2
   242  002172  062737  000004  001412          ADD     #4,VECTC2
   243  002200  013737  001412  001414          MOV     VECTC2,VECTD2
   244  002206  062737  000004  001414          ADD     #4,VECTD2
   245
   246  002214  013737  001372  001416          MOV     VECTA,PRA       ;SET UP VECTOR+2 ADDRESSES.
   247  002222  062737  000002  001416          ADD     #2,PRA
   248  002230  013737  001416  001420          MOV     PRA,PRB
   249  002236  062737  000004  001420          ADD     #4,PRB
   250  002244  013737  001420  001422          MOV     PRB,PRC
   251  002252  062737  000004  001422          ADD     #4,PRC
   252  002260  013737  001422  001424          MOV     PRC,PRD
   253  002266  062737  000004  001424          ADD     #4,PRD
   254  002274  013737  001406  001426          MOV     VECTA2,PRA2
   255  002302  062737  000002  001426          ADD     #2,PRA2
   256  002310  013737  001426  001430          MOV     PRA2,PRB2
   257  002316  062737  000004  001430          ADD     #4,PRB2
   258  002324  013737  001430  001432          MOV     PRB2,PRC2
   259  002332  062737  000004  001432          ADD     #4,PRC2
   260  002340  013737  001432  001434          MOV     PRC2,PRD2
   261  002346  062737  000004  001434          ADD     #4,PRD2
   262  002354  013700  001216          RSTART: MOV     $CPUOP,R0               ;GET CPU TYPE
   263  002360  042700  177760                  BIC     #177760,R0              ;MASK OFF ANOTHER BITS
   264  002364  116037  001440  001436          MOVB    CPUSPD(R0),CPUDLY       ;GET DELAY FACTOR
   265
   266                                                                  ;/-RESV-
(1)  002372  013777  001416  176772          MOV     PRA,@VECTA ;/RESTORE VECTOR FOR
(1)  002400  012777  004700  177010          MOV     #4700,@PRA      ;/ILLEGAL INTRO.
   267                                                               ;/-RESV-
(1)  002406  013777  001420  176760          MOV     PRB,@VECTB ;/RESTORE VECTOR FOR
(1)  002414  012777  004700  176776          MOV     #4700,@PRB      ;/ILLEGAL INTRO.
   268                                                               ;/-RESV-
(1)  002422  013777  001422  176746          MOV     PRC,@VECTC ;/RESTORE VECTOR FOR
(1)  002430  012777  004700  176764          MOV     #4700,@PRC      ;/ILLEGAL INTRO.
   269                                                               ;/-RESV-
(1)  002436  013777  001424  176734          MOV     PRD,@VECTD ;/RESTORE VECTOR FOR
(1)  002444  012777  004700  176752          MOV     #4700,@PRD      ;/ILLEGAL INTRO.
```

```
271                                     .SBTTL  TYPE PROGRAM NAME
(1)                                     ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
(1)  002452  005227  177777                    INC     #-1             ;;FIRST TIME?
(1)  002456  001041                            BNE     64$             ;;BRANCH IF NO
(1)  002460  104401  002526                    TYPE    ,65$            ;;TYPE ASCIZ STRING
(2)                                     .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
(2)  002464  005737  000042                    TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
(2)  002470  001012                            BNE     66$             ;;BRANCH IF YES
(2)  002472  123727  001210  000001            CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
(2)  002500  001406                            BEQ     66$             ;;BRANCH IF YES
(2)  002502  023727  001140  000176            CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
(2)  002510  001005                            BNE     67$             ;;BRANCH IF NO
(2)  002512  104406                            GTSWR                   ;;GET SOFT-SWR SETTINGS
(2)  002514  000403                            BR      67$
(2)  002516  112737  000001  001134  66$:      MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
(2)  002524                             67$:
(1)  002524  000416                            BR      64$             ;;GET OVER THE ASCIZ
(1)                                     ;;65$:  .ASCIZ  <CRLF>#CVIBAB IBV11A DIAGNOSTIC#<CRLF>
(1)  002562                             64$:
272  002562  000005                            RESET
273  002564  004537  010100                    JSR     R5,DEL50        ;DELAY FOR RESET PULSE TO FINISH
274  002570  001000                            1000            ; <SO THAT ANY ERROR MESSAGES ARE NOT TRASHED>
275
```

```
277 -
(3)
(3)
(2)  002572  000240                        ;;*******************************************************************
(1)  002574  012737  000050  001160        ;*TEST 1        *TEST THE ADDRESSABILITY OF THE IBS, IBD REGISTERS
(1)  002602  012737  002632  001106        ;*******************************************************************
278                                 TST1:   NOP
279  002610  012737  000001  001102        MOV     #50,$TIMES      ;;DO 50 ITERATIONS
280  002616  012737  000001  001174        MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
281  002624  012737  002632  001110
282                                         MOV     #1,$TSTNM       ;SET TEST #1.
283  002632  013746  000004                 MOV     #1,$TESTN       ;DON'T FORGET APT!
284  002636  012737  002664  000004 1$:     MOV     #1$,$LPERR
285                                         MOV     ERRVEC,-(SP)    ;SAVE CONTENTS OF ADDR. 4
286                                         MOV     #2$,ERRVEC      ;SET TIME-OUT TRAP VECTOR TO HANDLE
287                                                                 ;IN CASE WE TIME OUT WHEN
288  002644  005777  176512                                         ;WE ADDR. THE IBV-11
289                                         TST     @IBS            ;ADDR THE IBS, IF NO RESPONSE,
290  002650  012737  002672  000004                                 ;WILL TRAP TO 2$ FROM HERE
291                                         MOV     #3$,ERRVEC      ;CHANGE FOR ADDRESSING THE IBD REG.
292  002656  005777  176502
293                                         TST     @IBD            ;ADDR THE IBD REG.
294  002662  000406                                                 ;WE'LL TRAP TO 3$ FROM HERE IF BAD.
295  002664                                 BR      4$
(1)  002664  062706  000004        2$:      ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
296

                                    ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$
(1)
(1)  002670  104005                         ERROR   5               ;/MODULE FAULT DETECTED:
297                                                                 ;IBS REGISTER COULD NOT BE
298                                                                 ;ADDRESSED

                                    ;;$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$
300  002672                         3$:
(1)  002672  062706  000004                 ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
301

                                    ;;$$$$$$$$$>>> ERROR <<<$$$$$$$$$

(1)
(1)  002676  104005                         ERROR   5               ;/MODULE FAULT DETECTED:
302                                                                 ;ADDRESSED

                                    ;;$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$
304  002700  012637  000004        4$:      MOV     (SP)+,ERRVEC    ;RESTORE CONTENTS OF LOC 4.
305                                                                 ;/PR
(1)  002704  012746  000000                 MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
(1)  002710  012746  002716                 MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
(1)  002714  000002                         RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
(1)  002716                         64$:                            ;/IN STATUS REG.)
306
313
```

```
                                    ;;*****************************************************************
                                    ;*TEST 2         *TEST THAT BASE ADDRESSES +4,+6 RESPOND WHEN ADDRESSED
                                    ;*
                                    ;*EVEN THOUGH THE BASE ADDRESS +4 AND +6 ARE NOT USED,
                                    ;*THE IBV11A SHOULD RESPOND TO THEM
                                    ;*
                                    ;;*****************************************************************
002716  000004                    TST2:   SCOPE
002720  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS

002726  013746  000004                    MOV     ERRVEC,-(SP)    ;SAVE CONTENTS OF ADDR 4.
002732  012737  002760  000004            MOV     #1$,ERRVEC      ;SET TIME OUT TRAP VECTOR TO HANDLE
                                                                  ;IN CASE WE TIME OUT WHEN WE
                                                                  ;ADDRESS THE IBV-11 ADDRESSES +4,+6.
002740  005777  176422                    TST     @IBWC           ;TEST BASE ADDRESS +4, IF NO RESPONSE
                                                                  ;WILL TRAP TO 1$ FROM HERE

002744  012737  002770  000004            MOV     #2$,ERRVEC      ;CHANGE FOR ADDRESSING +6 ADDR.

002752  005777  176412                    TST     @IBCA           ;ADDR THE +6 ADDR. - TRAP IF BAD.
002756  000407                            BR      3$              ;CONTINUE IF GOOD.

002760                              1$:
002760  062706  000004                    ADD     #4,SP           ;/ADD #4 TO STACK POINTER.

                                    ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
002764  104005                            ERROR   5               ;/MODULE FAULT DETECTED:
                                                                  ;BASE ADDR+4 COULD NOT
                                                                  ;BE ADDRESSED.

                                    ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
002766  000403                            BR      3$

002770                              2$:
002770  062706  000004                    ADD     #4,SP           ;/ADD #4 TO STACK POINTER.

                                    ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
002774  104005                            ERROR   5               ;/MODULE FAULT DETECTED:
                                                                  ;BASE ADDR+6 COULD NOT
                                                                  ;BE ADDRESSED.

                                    ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

002776  012637  000004              3$:   MOV     (SP)+,ERRVEC    ;RESTORE CONTENTS OF LOC 4.
```

```
                                ;;*****************************************************************
                                ;*TEST 3         *TEST THAT IBS IS CLEAR AT INIT OF TESTING
                                ;;*****************************************************************
                        TST3:   SCOPE
003002  000004
003004  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION

003012  000005                          RESET                   ;ISSUE SYSTEM INIT.

003014  005037  001124                  CLR     $GDDAT          ;EXPECT ZERO CSR.
003020  017737  176336  001126          MOV     @IBS,$BDDAT     ;READ CSR.
003026  001401                          BEQ     TST4            ;;

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

                                        ERROR   3               ;/MODULE FAULT DETECTED:
003030  104003                                                  ;IBS NOT CLEAR ON INT.

                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

                                ;;*****************************************************************
                                ;*TEST 4         *TEST THAT IBD IS CLEAR AT INIT OF TESTING
                                ;;*****************************************************************
                        TST4:   SCOPE
003032  000004
003034  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION

003042  000005                          RESET                   ;ISSUE SYSTEM INITIALIZE.

003044  005037  001124                  CLR     $GDDAT          ;EXPECT ZERO CSR.
003050  117737  176310  001126          MOVB    @IBD,$BDDAT     ;READ DBR
003056  001401                          BEQ     TST5            ;;

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

                                        ERROR   4               ;/MODULE FAULT DETECTED:
003060  104004                                                  ;IBD NOT CLEAR ON INIT.

                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
```

```
 367
 374
 375                                    ;;*********************************************************************
 (3)                                    ;*TEST 5         *TEST THAT BASE ADDRESSES +4,+6 RETURN ZERO WHEN READ
 (4)                                    ;*
 (4)                                    ;*BASE ADDRESS +4 AND +6 SHOULD RETURN A ZERO WHEN
 (4)                                    ;*READ, IN THIS TEST WE WILL TRY THAT.
 (4)                                    ;*
 (3)                                    ;*********************************************************************
 (2)  003062  000004                    TST5:   SCOPE
 (1)  003064  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
 376
 377  003072  005037  001124                    CLR     $GDDAT          ;EXPECT ZERO RETURN
 378  003076  017737  176264  001126            MOV     @IBWC,$BDDAT    ;READ BASE ADDRESS+4
 379  003104  001402                            BEQ     1$              ;IF ZERO - GOOD.
 380

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

 (1)
 (1)  003106  104006                    ERROR   6               ;/MODULE FAULT DETECTED:
 381                                            ;SHOULD HAVE READ BACK ZERO FROM
 382                                            ;THIS ADDR.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
 384  003110  000405                    BR      TST6            ;;
 385
 386  003112  017737  176252  001126  1$:       MOV     @IBCA,$BDDAT    ;READ BASE ADDR+6, SHOULD BE ZERO
 387  003120  001401                            BEQ     TST6            ;;
 388

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

 (1)
 (1)  003122  104006                    ERROR   6               ;/MODULE FAULT DETECTED:
 389                                            ;SHOULD HAVE READ BACK ZERO FROM
 390                                            ;BASE ADDR+6.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
 392
```

```
                        ;;*****************************************************************
                        ;*TEST 6         *TEST THAT WE CAN SET TCS (BIT00), TCS SETS CMD (BIT10)
                        ;;*****************************************************************
003124  000004          TST6:   SCOPE

003126  005077  176230          CLR     @IBS            ;CLEAR CLR
003132  052777  000001  176222  BIS     #BIT0,@IBS      ;SET TCS.
003140  012737  002001  001124  MOV     #BIT0!BIT10,$GDDAT ;EXPECT ONLY TCS AND CMD TO SET
003146  017737  176210  001126  MOV     @IBS,$BDDAT     ;READ THE IBS.
003154  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;DID TCS AND CMD SET?
003162  000402                  BR      1$              ;YES - CONTINUE

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1) 003164  104003              ERROR   3               ;/MODULE FAULT DETECTED:
                                                        ;TCS AND/OR CMD FAILED TO SET

                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
003166  000412                  BR      TST7            ;;

003170  042777  000001  176164  1$:     BIC     #BIT0,@IBS      ;CLEAR TCS.
003176  005037  001124          CLR     $GDDAT          ;EXPECT TCS AND CMD TO CLEAR
003202  017737  176154  001126  MOV     @IBS,$BDDAT     ;READ IBS, DID THEY CLEAR?
003210  001401                  BEQ     TST7            ;;

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1) 003212  104003              ERROR   3               ;/MODULE FAULT DETECTED:
                                                        ;TCS AND/OR CMD FAILED TO CLEAR.

                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$


                        ;;*****************************************************************
                        ;*TEST 7         *TEST THAT EOP (BIT01) WILL SET
                        ;;*****************************************************************
003214  000004          TST7:   SCOPE

003216  005077  176140          CLR     @IBS            ;CLEAR CSR.
003222  052777  000002  176132  BIS     #BIT1,@IBS      ;SET EOP
003230  012737  000002  001124  MOV     #BIT1,$GDDAT    ;EXPECT ONLY EOP TO SET.
003236  017737  176120  001126  MOV     @IBS,$BDDAT     ;READ IBS
003244  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;DID EOP SET?
003252  001402                  BEQ     1$              ;YES - CONTINUE

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1) 003254  104003              ERROR   3               ;/MODULE FAULT DETECTED:
                                                        ;EOP BIT SETTING ERROR.

                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
```

```
                                T7      *TEST THAT EOP (BIT01) WILL SET

   426  003256  000412                  BR      TST10           ;;
   427
   428  003260  042777  000002  176074  1$:     BIC     #BIT1,@IBS      ;CLEAR EOP
   429  003266  005037  001124                  CLR     $GDDAT          ;EXPECT A ZERO CSR.
   430  003272  017737  176064  001126          MOV     @IBS,$BDDAT     ;READ IBS, IS IT CLEAR?
   431  003300  001401                  BEQ     TST10           ;;
   432

                                        .;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

   (1)
   (1)  003302  104003                  ERROR   3               ;/MODULE FAULT DETECTED:
   433                                                          ;IBS FAILED TO CLEAR

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

   435
   436                          ;;**************************************************************
   (3)                          ;*TEST 10       *TEST THAT REM (BIT02) WILL SET + CLEAR
   (3)                          ;;**************************************************************
   (2)  003304  000004          TST10:  SCOPE
   437
   438  003306  005077  176050          CLR     @IBS            ;CLEAR CSR.
   439  003312  052777  000004  176042  BIS     #BIT02,@IBS     ;SET REM
   440  003320  012737  000004  001124  MOV     #BIT02,$GDDAT   ;EXPECT ONLY REM TO SET.
   441  003326  017737  176030  001126  MOV     @IBS,$BDDAT     ;READ IBS.
   442  003334  023737  001126  001124  CMP     $BDDAT,$GDDAT   ;DID REM AND ONLY REM SET?
   443  003342  001402                  BEQ     1$              ;YES - CONTINUE
   444

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

   (1)
   (1)  003344  104003                  ERROR   3               ;/MODULE FAULT DETECTED:
   445                                                          ;REM BIT SETTING ERROR.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
   447  003346  000412                  BR      TST11           ;;
   448
   449  003350  042777  000004  176004  1$:     BIC     #BIT02,@IBS     ;CLEAR REM BIT.
   450  003356  005037  001124                  CLR     $GDDAT          ;EXPECT ZERO CSR.
   451  003362  017737  175774  001126          MOV     @IBS,$BDDAT     ;READ IBS - IS IT CLEAR?
   452  003370  001401                  BEQ     TST11           ;;
   453

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

   (1)
   (1)  003372  104003                  ERROR   3               ;/MODULE FAULT DETECTED:
   454                                                          ;IBS FAILED TO CLEAR.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

   456
```

```
458                                     ;;*****************************************************************
(3)                                     ;*TEST 11       *TEST THAT IBC (BIT03) WILL SET AND CLEAR
(3)                                     ;;*****************************************************************
(2)  003374  000004                     TST11:  SCOPE
459  003376  005077  175760                        CLR     @IBS            ;CLEAR CSR.
460  003402  052777  000010  175752               BIS     #BIT03,@IBS     ;SET IBC
461  003410  012737  000010  001124               MOV     #BIT03,$GDDAT   ;EXPECT ONLY IBC TO BE SET
462  003416  017737  175740  001126               MOV     @IBS,$BDDAT     ;READ IBS.
463  003424  023737  001124  001126               CMP     $GDDAT,$BDDAT   ;DID IBS SET?
464  003432  001414                              BEQ     1$              ;YES CONTINUE
465  003434  012777  000010  175720               MOV     #BIT03,@IBS     ;TRY SETTING IBC AGAIN.
466  003442  017737  175714  001126               MOV     @IBS,$BDDAT     ;MEMORY REFRESH MIGHT HAVE
467  003450  023737  001124  001126               CMP     $GDDAT,$BDDAT   ;GOT IN THE WAY.
468  003456  001402                              BEQ     1$
469

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  003460  104003                             ERROR   3               ;/MODULE FAULT DETECTED:
470                                                                     ;IBS BIT SETTING ERROR.
471  003462  000416                             BR      TST12           ;;
472  003464  004537  010100          1$:        JSR     R5,DEL50        ;DELAY 150 US.
473  003470  000006                             .WORD   6
474  003472  012737  002001  001124               MOV     #BIT10!BIT0,$GDDAT      ;EXP CMD AND TCS.
475  003500  017737  175656  001126               MOV     @IBS,$BDDAT     ;READ IBS - IS IT CLEAR?
476  003506  023737  001124  001126               CMP     $GDDAT,$BDDAT
477  003514  001401                              BEQ     TST12           ;;
478

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  003516  104010                             ERROR   10              ;/MODULE FAULT DETECTED:
479                                                                     ;IBS NOT CLEAR AFTER IBC
480                                             ;DOES "$CPUOP" INDICATE THE CORRECT CODE FOR THIS CPU ? OR
481                                     ;;*****************************  *************************************
(3)                                     ;*TEST 12       *TEST THAT TON (BIT05) AND TKR (BIT09) SET AND CLEAR
(3)                                     ;;*****************************************************************
(2)  003520  000004                     TST12:  SCOPE
482
483  003522  005077  175634                        CLR     @IBS            ;CLEAR THE CSR.
484  003526  052777  000040  175626               BIS     #BIT5,@IBS      ;SET TON.
485  003534  012737  001040  001124               MOV     #BIT5!BIT9,$GDDAT ;EXPECT ONLY TON AND TKR TO SET.
486  003542  017737  175614  001126               MOV     @IBS,$BDDAT     ;READ CSR.
487  003550  023737  001124  001126               CMP     $GDDAT,$BDDAT   ;DID THEY BOTH SET?
488  003556  001402                              BEQ     1$              ;YES - CONTINUE.
489
490

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  003560  104003                             ERROR   3               ;/MODULE FAULT DETECTED:
491                                                                     ;ERROR IN SETTING TON BIT OR TKR BIT.
```

```
CVIBA-B MACY11 30G(1063)  25-FEB-83  08:23  PAGE 10-1
CVIBAB.P11    13-DEC-82 09:29          T12     *TEST THAT TON (BIT05) AND TKR (BIT09) SET AND CLEAR                    SEQ 0032

                                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
   493  003562  000412                          BR      TST13           ;;
   494
   495  003564  042777  000040  175570  1$:     BIC     #BIT5,@IBS      ;WHEN TON CLEARED, TKR SHOULD CLEAR.
   496  003572  005037  001124                  CLR     $GDDAT          ;EXPECT ZERO CSR.
   497  003576  017737  175560  001126          MOV     @IBS,$BDDAT     ;DID IT CLEAR?
   498  003604  001401                          BEQ     TST13           ;;
   499

                                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

   (1)
   (1)  003606  104003                          ERROR   3               ;/MODULE FAULT DETECTED:
   500                                                                  ;CSR FAILED TO CLEAR.

                                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

   502
```

```
                                ;;*****************************************************************
                                ;*TEST 13       *TEST THAT LON (BIT04) WILL SET AND CLEAR
                                ;;*****************************************************************
                        TST13:  SCOPE
003610  000004                 
003612  005077  175544         CLR     @IBS            ;CLEAR THE CSR.
003616  052777  000020  175536 BIS     #BIT4,@IBS      ;SET LON
003624  012737  000020  001124 MOV     #BIT4,$GDDAT    ;EXPECT ONLY LON TO BE SET.
003632  017737  175524  001126 MOV     @IBS,$BDDAT     ;READ CSR
003640  023737  001124  001126 CMP     $GDDAT,$BDDAT   ;DID THEY BOTH SET ?
003646  001402                 BEQ     1$              ;;YES - CONTINUE

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  003650  104003             ERROR   3               ;/MODULE FAULT DETECTED:
                                                        ;ERROR IN SETTING LON BIT

                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
003652  000412                 BR      TST14           ;;
003654  042777  000020  175500  1$:     BIC     #BIT4,@IBS      ;WHEN LON CLEARED, LNK SHOULD CLEAR
003662  005037  0011
003666  017737  1754 )  001126 MOV     @IBS,$BDDAT     ;DID IT CLEAR ?
003674  001401                 BEQ     TST14           ;;YES - CONTINUE

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  003676  104003             ERROR   3               ;/MODULE FAULT DETECTED:
                                                        ;CSR FAILED TO CLEAR LON BIT

                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
```

```
523                                     ;;*********************************************************************
(3)                                     ;*TEST 14       *TEST THAT IE (BIT06) SET AND CLEAR
(3)                                     ;;*********************************************************************
(2) 003700  000004                      TST14:  SCOPE
524
525                                                             ;/PR
(1) 003702  012746  000300                      MOV     #300,-(SP)      ;/SET CPU PRIORITY ON RETURN
(1) 003706  012746  003714                      MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
(1) 003712  000002                              RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
(1) 003714                              64$:                            ;/IN STATUS REG.)
526
527 003714  005077  175442                      CLR     @IBS            ;CLEAR CSR.
528
529 003720  052777  000100  175434              BIS     #BIT6,@IBS      ;SET IE.
530 003726  012737  000100  001124              MOV     #BIT6,$GDDAT    ;EXPECT ONLY BIT 6 TO SET.
531 003734  017737  175422  001126              MOV     @IBS,$BDDAT     ;READ IBS.
532 003742  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;DID IE SET?
533 003750  001402                              BEQ     1$              ;YES - CONTINUE.
534

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1) 003752  104003                              ERROR   3               ;/MODULE FAULT DETECTED:
535                                                                     ;ERROR IN SETTING IE BIT.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
537 003754  000412                              BR      TST15           ;;
538
539 003756  005037  001124          1$:         CLR     $GDDAT          ;EXPECT ZERO CSR AFTER.
540 003762  042777  000100  175372              BIC     #BIT6,@IBS      ;IE IS CLEARED.
541 003770  017737  175366  001126              MOV     @IBS,$BDDAT     ;READ CSR - IS IT CLEAR?
542 003776  001401                              BEQ     TST15           ;;
543

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1) 004000  104003                              ERROR   3               ;/MODULE FAULT DETECTED:
544                                                                     ;FAILED TO CLEAR CSR.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

546
```

```
548                                     ;;*******************************************************************
(3)                                     ;*TEST 15       *TEST THAT ACC (BIT07) CAN BE SET AND CLEARED
(3)                                     ;;*******************************************************************
(2)  004002  000004                     TST15:  SCOPE
549
550  004004  005077  175352                    CLR     @IBS            ;CLEAR CSR.
551  004010  052777  000200  175344            BIS     #BIT7,@IBS      ;SET ACC.
552  004016  012737  000200  001124            MOV     #BIT7,$GDDAT    ;EXPECT ONLY ACC TO SET.
553  004024  017737  175332  001126            MOV     @IBS,$BDDAT     ;READ IBS.
554  004032  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;DID ACC SET?
555  004040  001402                            BEQ     1$              ;YES - CONTINUE.
556

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  004042  104003                            ERROR   3               ;/MODULE FAULT DETECTED:
                                                                        ;FAILURE IN SETTING BIT 7 (ACC).
557

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
559  004044  000412                            BR      TST16           ;;
560
561  004046  042777  000200  175306  1$:      BIC     #BIT7,@IBS      ;TRY CLEARING ACC.
562  004054  005037  001124                    CLR     $GDDAT          ;EXPECT ZERO CSR.
563  004060  017737  175276  001126            MOV     @IBS,$BDDAT     ;READ IBS, IS IT CLEAR?
564  004066  001401                            BEQ     TST16           ;;
565

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  004070  104003                            ERROR   3               ;/MODULE FAULT DETECTED:
                                                                        ;IBS FAILED TO CLEAR.
566

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

568
```

```
570                                     ;;*********************************************************************
(3)                                     ;*TEST 16       * TEST THAT SRQ (BIT15) CAN BE SET AND CLEARED (IF SWR10=1)
(3)                                     ;;*********************************************************************
(2)  004072  000004                     TST16:  SCOPE
571  004074  005077  175262                    CLR     @IBS            ;CLEAR CSR
572  004100  052777  100000  175254           BIS     #BIT15,@IBS     ;SET SRQ
573  004106  012737  100000  001124           MOV     #BIT15,$GDDAT   ;LOAD EXPECTED VALUE
574  004114  032777  002000  175016           BIT     #BIT10,@SWR     ;TEST IF 'ER1-INHIBIT" SWITCH IS SET (CLOSED-ON)
575  004122  001002                           BNE     1$              ;BR IF SWITCH IS SET
576  004124  005037  001124                   CLR     $GDDAT          ;RELOAD EXPECTED
577  004130  017737  175226  001126  1$:      MOV     @IBS,$BDDAT     ;READ IBS
578  004136  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;DID SRQ SET?
579  004144  001402                           BEQ     2$              ;;YES - CONTINUE
580
                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  004146  104011                     ERROR   11              ;/MODULE FAULT DETECTED:
                                                                ;"SRQ" BIT IN THE IBS REGISTER IS IN ERROR
581                                                             ;IS SWR10 ("ER1-INHIBIT SWITCH") CORRECT ?
582
                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
584  004150  000412                     BR      TST17           ;;
585  004152  042777  100000  175202  2$:      BIC     #BIT15,@IBS     ;CLEAR SRQ
586  004160  005037  001124                   CLR     $GDDAT          ;CLEAR EXPECTED
587  004164  017737  175172  001126           MOV     @IBS,$BDDAT     ;READ IBS BITS
588  004172  100001                           BPL     TST17           ;;BR IF CLEARED
589
                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  004174  104003                     ERROR   3               ;/MODULE FAULT DETECTED:
590                                                             ;FAILED TO CLEAR IBS "SRQ" BIT
                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
618
```

```
620
(5)                                     ;;*****************************************************************
(4)                                     ;*TEST 17        *TEST THAT IBD BIT 0 CAN BE SET + CLEARED
(4)                                     ;;*****************************************************************
(3)  004176  000004                     TST17:  SCOPE
(1)
(1)                                                                     ;/MACRO BDT
(1)  004200  012777  000060  175154             MOV     #BIT4!BIT5,@IBS ;/SET TCN AND LON.
(1)  004206  012737  000001  001124             MOV     #BIT0,$GDDAT    ;/WE'RE GONNA TEST BIT 0.
(1)  004214  013777  001124  175142             MOV     $GDDAT,@IBD     ;/SET THE BIT.
(1)
(1)  004222  117737  175136  001126             MOVB    @IBD,$BDDAT     ;/READ THE IBD.
(1)  004230  123737  001124  001126             CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
(1)  004236  001402                             BEQ     1$              ;/YES - CONTINUE.
(1)
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004240  104004                             ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/ERROR IN SETTING IBD BIT 0.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(3)  004242  000412                             BR      TST20           ;;
(1)  004244  005037  001124     1$:    CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
(1)  004250  042777  000001  175106     BIC     #BIT0,@IBD      ;/BIT 0 IS CLEARED.
(1)  004256  117737  175102  001126     MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
(3)  004264  001401                     BEQ     TST20           ;;
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004266  104004                     ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                             ;/FAILED TO CLEAR IBD.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(1)
```

```
622
(5)                                     ;;*****************************************************************
(4)                                     ;*TEST 20        *TEST THAT IBD BIT 1 CAN BE SET + CLEARED
(4)                                     ;;*****************************************************************
(3)  004270  000004                     TST20:  SCOPE
(1)
(1)                                                                     ;/MACRO BDT
(1)  004272  012777  000060  175062             MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
(1)  004300  012737  000002  001124             MOV     #BIT1,$GDDAT    ;/WE'RE GONNA TEST BIT 1.
(1)  004306  013777  001124  175050             MOV     $GDDAT,@IBD     ;/SET THE BIT.
(1)
(1)  004314  117737  175044  001126             MOVB    @IBD,$BDDAT     ;/READ THE IBD.
(1)  004322  123737  001124  001126             CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
(1)  004330  001402                             BEQ     1$              ;/YES - CONTINUE.
(1)
(2)

                                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004332  104004                             ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/ERROR IN SETTING IBD BIT 1.

                                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(3)  004334  000412                             BR      TST21           ;;
(1)  004336  005037  001124             1$:     CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
(1)  004342  042777  000002  175014             BIC     #BIT1,@IBD      ;/BIT 1 IS CLEARED.
(1)  004350  117737  175010  001126             MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
(3)  004356  001401                             BEQ     TST21           ;;
(2)

                                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004360  104004                             ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                                     ;FAILED TO CLEAR IBD.

                                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

(1)
```

```
624
(5)                                     ;;*****************************************************************
(4)                                     ;*TEST 21       *TEST THAT IBD BIT 2 CAN BE SET + CLEARED
(4)                                     ;;*****************************************************************
(3)  004362  000004                     TST21:  SCOPE
(1)
(1)                                                                     ;/MACRO BDT
(1)  004364  012777  000060  174770             MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
(1)  004372  012737  000004  001124             MOV     #BIT2,$GDDAT    ;/WE'RE GONNA TEST BIT 2.
(1)  004400  013777  001124  174756             MOV     $GDDAT,@IBD     ;/SET THE BIT.
(1)
(1)  004406  117737  174752  001126             MOVB    @IBD,$BDDAT     ;/READ THE IBD.
(1)  004414  123737  001124  001126             CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
(1)  004422  001402                             BEQ     1$              ;/YES - CONTINUE.
(1)
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004424  104004                             ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/ERROR IN SETTING IBD BIT 2.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(3)  004426  000412                             BR      TST22           ;;
(1)  004430  005037  001124             1$:     CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
(1)  004434  042777  000004  174722             BIC     #BIT2,@IBD      ;/BIT 2 IS CLEARED.
(1)  004442  117737  174716  001126             MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
(3)  004450  001401                             BEQ     TST22           ;;
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004452  104004                             ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/FAILED TO CLEAR IBD.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(1)
```

```
626
(5)                                     ;;*****************************************************************
(4)                                     ;*TEST 22        *TEST THAT IBD BIT 3 CAN BE SET + CLEARED
(4)                                     ;;*****************************************************************
(3)  004454  000004                     TST22:  SCOPE
(1)
(1)                                                                     ;/MACRO BDT
(1)  004456  012777  000060  174676             MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
(1)  004464  012737  000010  001124             MOV     #BIT3,$GDDAT    ;/WE'RE GONNA TEST BIT 3.
(1)  004472  013777  001124  174664             MOV     $GDDAT,@IBD     ;/SET THE BIT.
(1)
(1)  004500  117737  174660  001126             MOVB    @IBD,$BDDAT     ;/READ THE IBD.
(1)  004506  123737  001124  001126             CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
(1)  004514  001402                             BEQ     1$              ;/YES - CONTINUE.
(1)
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004516  104004                             ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/ERROR IN SETTING IBD BIT 3.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(3)  004520  000412                             BR      TST23           ;;
(1)  004522  005037  001124             1$:     CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
(1)  004526  042777  000010  174630             BIC     #BIT3,@IBD      ;/BIT 3 IS CLEARED.
(1)  004534  117737  174624  001126             MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
(3)  004542  001401                             BEQ     TST23           ;;
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004544  104004                             ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/FAILED TO CLEAR IBD.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(1)
```

```
628
(5)                             ;;*****************************************************************
(4)                             ;*TEST 23       *TEST THAT IBD BIT 4 CAN BE SET + CLEARED
(4)                             ;;*****************************************************************
(3)  004546  000004             TST23:  SCOPE
(1)
(1)                                                             ;/MACRO BDT
(1)  004550  012777  000060  174604     MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
(1)  004556  012737  000020  001124     MOV     #BIT4,$GDDAT    ;/WE'RE GONNA TEST BIT 4.
(1)  004564  013777  001124  174572     MOV     $GDDAT,@IBD     ;/SET THE BIT.
(1)
(1)  004572  117737  174566  001126     MOVB    @IBD,$BDDAT     ;/READ THE IBD.
(1)  004600  123737  001124  001126     CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
(1)  004606  001402                     BEQ     1$              ;/YES - CONTINUE.
(1)
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004610  104004                     ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                             ;/ERROR IN SETTING IBD BIT 4.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(3)  004612  000412                     BR      TST24           ;;
(1)  004614  005037  001124     1$:     CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
(1)  004620  042777  000020  174536     BIC     #BIT4,@IBD      ;/BIT 4 IS CLEARED.
(1)  004626  117737  174532  001126     MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
(3)  004634  001401                     BEQ     TST24           ;;
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004636  104004                     ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                             ;FAILED TO CLEAR IBD.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

(1)
```

```
630
(5)                                     ;;*********************************************************************
(4)                                     ;*TEST 24        *TEST THAT IBD BIT 5 CAN BE SET + CLEARED
(4)                                     ;;*********************************************************************
(3)  004640  000004                     TST24:  SCOPE
(1)
(1)                                                                     ;/MACRO BDT
(1)  004642  012777  000060  174512             MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
(1)  004650  012737  000040  001124             MOV     #BIT5,$GDDAT    ;/WE'RE GONNA TEST BIT 5.
(1)  004656  013777  001124  174500             MOV     $GDDAT,@IBD     ;/SET THE BIT.
(1)
(1)  004664  117737  174474  001126             MOVB    @IBD,$BDDAT     ;/READ THE IBD.
(1)  004672  123737  001124  001126             CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
(1)  004700  001402                             BEQ     1$              ;/YES - CONTINUE.
(1)
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004702  104004                             ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/ERROR IN SETTING IBD BIT 5.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(3)  004704  000412                             BR      TST25           ;;
(1)  004706  005037  001124             1$:     CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
(1)  004712  042777  000040  174444             BIC     #BIT5,@IBD      ;/BIT 5 IS CLEARED.
(1)  004720  117737  174440  001126             MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
(3)  004726  001401                             BEQ     TST25           ;;
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004730  104004                     ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                             ;FAILED TO CLEAR IBD.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

(1)
```

```
632
(5)                                     ;;*********************************************************************
(4)                                     ;*TEST 25       *TEST THAT IBD BIT 6 CAN BE SET + CLEARED
(4)                                     ;;*********************************************************************
(3)  004732  000004                     TST25:  SCOPE
(1)
(1)                                                                     ;/MACRO BDT
(1)  004734  012777  000060  174420             MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
(1)  004742  012737  000100  001124             MOV     #BIT6,$GDDAT    ;/WE'RE GONNA TEST BIT 6.
(1)  004750  013777  001124  174406             MOV     $GDDAT,@IBD     ;/SET THE BIT.
(1)
(1)  004756  117737  174402  001126             MOVB    @IBD,$BDDAT     ;/READ THE IBD.
(1)  004764  123737  001124  001126             CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
(1)  004772  001402                             BEQ     1$              ;/YES - CONTINUE.
(1)
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  004774  104004                             ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/ERROR IN SETTING IBD BIT 6.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(3)  004776  000412                             BR      TST26           ;;
(1)  005000  005037  001124     1$:     CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
(1)  005004  042777  000100  174352     BIC     #BIT6,@IBD      ;/BIT 6 IS CLEARED.
(1)  005012  117737  174346  001126     MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
(3)  005020  001401                     BEQ     TST26           ;;
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  005022  104004                     ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                             ;FAILED TO CLEAR IBD.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

(1)
```

```
634
(5)                                     ;;*********************************************************************
(4)                                     ;*TEST 26        *TEST THAT IBD BIT 7 CAN BE SET + CLEARED
(4)                                     ;;*********************************************************************
(3)  005024  000004                     TST26:  SCOPE
(1)
(1)                                                                     ;/MACRO BDT
(1)  005026  012777  000060  174326             MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
(1)  005034  012737  000200  001124             MOV     #BIT7,$GDDAT    ;/WE'RE GONNA TEST BIT 7.
(1)  005042  013777  001124  174314             MOV     $GDDAT,@IBD     ;/SET THE BIT.
(1)
(1)  005050  117737  174310  001126             MOVB    @IBD,$BDDAT     ;/READ THE IBD.
(1)  005056  123737  001124  001126             CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
(1)  005064  001402                             BEQ     1$              ;/YES - CONTINUE.
(1)
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  005066  104004                             ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/ERROR IN SETTING IBD BIT 7.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(3)  005070  000412                             BR      TST27           ;;
(1)  005072  005037  001124             1$:     CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
(1)  005076  042777  000200  174260             BIC     #BIT7,@IBD      ;/BIT 7 IS CLEARED.
(1)  005104  117737  174254  001126             MOVB    @IBD,$BDDAT     ;/READ IBD. IS IT CLEAR?
(3)  005112  001401                             BEQ     TST27           ;;
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  005114  104004                             ERROR   4               ;/MODULE FAULT DETECTED:
(1)                                                                     ;FAILED TO CLEAR IBD.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

(1)
635
```

```
CVIBA-B MACY11 30G(1063)  25-FEB-83  08:23  PAGE 23                                                    SEQ 0045
CVIBAB.P11    13-DEC-82 09:29           T27     *TEST THAT NO DATA GETS XFERRED, IF NOT ENABLED

   637                                  ;;*************************************************************************
   (3)                                  ;*TEST 27       *TEST THAT NO DATA GETS XFERRED, IF NOT ENABLED
   (3)                                  ;;*************************************************************************
   (2)  005116  000004                  TST27:  SCOPE
   638
   639  005120  005077  174236                   CLR     @IBS            ;CLEAR CSR
   640  005124  112777  000252  174232           MOVB    #252,@IBD       ;TRY XFERRING DATA
   641  005132  005037  001124                   CLR     $GDDAT          ;NO DATA SHOULD XFERR
   642  005136  117737  174222  001126           MOVB    @IBD,$BDDAT     ;READ BUFFER REG.
   643  005144  001401                           BEQ     TST30           ;;
   644

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

   (1)
   (1)  005146  104002                  ERROR   2                       ;/MODULE FAULT DETECTED:
   645                                                                  ;DATA WAS XFERRED THROUGH IBD
   646                                                                  ;EVEN THOUGH TON AND LON CLEARED.
   647                                                                  ;SIGNAL 'ENB XFER L" PROBABLY
   648                                                                  ;STUCK LOW.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

   650
```

```
                                ;;*****************************************************************
                                ;*TEST 30       *TEST IBD-DAC (BIT08) AND IBD-DAV (BIT09) CAN GE SET + CLEARED
                                ;;*****************************************************************
005150  000004                  TST30:  SCOPE
                                ;               <<*ALSO TEST IBS-LNR (BIT08)>>
005152  005077  174204                  CLR     @IBS            ;CLEAR CSR.
005156  005077  174202                  CLR     @IBD            ;CLEAR DATA REG.
005162  032777  000400  174174          BIT     #BIT8,@IBD      ;IS DAC SET?
005170  001002                          BNE     1$              ;YES (GOOD)

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  005172  104002                     ERROR   2               ;/MODULE FAULT DETECTED:
                                                                ;IBD-DAC NOT SET.
005174  000503                          BR      TST31           ;;
005176  005037  001124          1$:     CLR     $GDDAT          ;CLEAR EXPECTED
005202  017737  174154  001126          MOV     @IBS,$BDDAT     ;READ STATUS
005210  032737  000400  001126          BIT     #BIT8,$BDDAT    ;IS LNR CLEARED?
005216  001402                          BEQ     2$              ;;BR IF LNR IS CLEARED

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  005220  104003                     ERROR   3               ;/MODULE FAULT DETECTED:
                                                                ;IBS-LNR SET IN ERROR
005222  000470                          BR      TST31           ;;
005224  052777  000260  174130  2$:     BIS     #BIT5!BIT4!BIT7,@IBS    ;SET TON, LON AND ACC BITS
005232  012777  000252  174124          MOV     #252,@IBD       ;PUT DATA IN IBD.
005240  017737  174120  001126          MOV     @IBD,$BDDAT     ;READ IBD.
005246  032737  001000  001126          BIT     #BIT9,$BDDAT    ;DID DAV SET?
005254  001002                          BNE     3$              ;YES - CONTINUE.

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  005256  104002                     ERROR   2               ;/MODULE FAULT DETECTED:
                                                                ;DAV FAILED TO SET.
005260  000451                          BR      TST31           ;;
005262  012737  000660  001124  3$:     MOV     #660,$GDDAT     ;LOAD EXPECTED
005270  017737  174066  001126          MOV     @IBS,$BDDAT     ;READ CSR
005276  032737  000400  001126          BIT     #BIT8,$BDDAT    ;TEST IF LNR SET
005304  001002                          BNE     4$              ;BR IF IBS-LNR WAS SET

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  005306  104003                     ERROR   3               ;/MODULE FAULT DETECTED:
                                                                ;IBS-LNR FAILED TO SET
005310  000435                          BR      TST31           ;;
005312  012777  000060  174042  4$:     MOV     #BIT4!BIT5,@IBS ;CLEAR ACC BIT
005320  105777  174040                  TSTB    @IBD            ;READ LOW BYTE OF IBD.
```

```
CVIBA-B MACY11 30G(1063)  25-FEB-83  08:23  PAGE 24-1
CVIBAB.P11    13-DEC-82 09:29          T30     *TEST IBD-DAC (BIT08) AND IBD-DAV (BIT09) CAN GE SET + CLEARED          SEQ 0047

   685  005324  032777  000400  174032          BIT     #BIT8,@IBD      ;DID DAC CLEAR?
   686  005332  001402                          BEQ     5$              ;YES - CONTINUE.
   687

                                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

   (1)
   (1)  005334  104002                          ERROR   2               ;/MODULE FAULT DETECTED:
   688                                                                  ;DAC FAILED TO CLEAR.
   689  005336  000422                          BR      TST31           ;;
   690  005340  032777  001000  174016  5$:     BIT     #BIT9,@IBD      ;DID DAV CLEAR?
   691  005346  001402                          BEQ     6$
   692

                                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

   (1)
   (1)  005350  104002                          ERROR   2               ;/MODULE FAULT DETECTED:
   693                                                                  ;IBD-DAV FAILED TO CLEAR.
   694  005352  000414                          BR      TST31           ;;
   695  005354  005077  174002          6$:     CLR     @IBS            ;CLEAR STATUS
   696  005360  005037  001124                  CLR     $GDDAT          ;CLEAR EXPECTED
   697  005364  017737  173772  001126          MOV     @IBS,$BDDAT     ;READ CSR
   698  005372  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE
   699  005400  001401                          BEQ     TST31           ;;
   700

                                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

   (1)
   (1)  005402  104001                          ERROR   1               ;/MODULE FAULT DETECTED:
   701                                                                  ;CSR FAILED TO CLEAR
   744
   745                                          ;/MACRO -SIGC-
   (1)
   (5)                              ;;*********************************************************************
   (4)                              ;*TEST 31        *TEST THAT IBD - REN SETS WHEN IBS - REM SETS, ALSO TEST CLEAR
   (4)                              ;;*********************************************************************
   (3)  005404  000004              TST31:  SCOPE
   (1)
   (1)  005406  005077  173750              CLR     @IBS            ;/CLEAR CSR.
   (1)  005412  052777  000004  173742      BIS     #BIT2,@IBS      ;/SET REM, SHOULD SET REN.
   (1)  005420  032777  010000  173736      BIT     #BIT12,@IBD     ;/DID REN SET?
   (1)  005426  001011                      BNE     1$              ;/YES - LETS TRY CLEARING IT.
   (1)  005430  052777  000004  173724      BIS     #BIT2,@IBS      ;/SET REM, MEMORY
   (1)                                                              ;/REFRESH COULD HAVE
   (1)                                                              ;/INTERRUPTED US.
   (1)  005436  032777  010000  173720      BIT     #BIT12,@IBD     ;/DID REN SET THIS TIME?
   (1)  005444  001002                      BNE     1$
   (2)

                                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

   (2)
   (2)  005446  104002                          ERROR   2               ;/MODULE FAULT DETECTED:
   (1)                                                                  ;/REN FAILED TO SET WHEN REM SET.
```

```
                                                *TEST THAT IBD - REN SETS WHEN IBS - REM SETS, ALSO TEST CLEAR                SEQ 0048


                                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(3)  005450  000410                             BR      TST32           ;;
(1)  005452  042777  000004  173702  1$:       BIC     #BIT2,@IBS      ;/CLEAR REM, SHOULD CLEAR REN.
(1)  005460  032777  010000  173676             BIT     #BIT12,@IBD     ;/DID REN CLEAR?
(3)  005466  001401                             BEQ     TST32           ;;
(2)

                                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  005470  104002                             ERROR   2               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/REN FAILED TO CLEAR.

                                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

(1)
746
```

```
748
(1)
(5)                                     ;;*****************************************************************
(4)                                     ;*TEST 32       *TEST THAT IBD - IFC SETS WHEN IBS - IBC SETS, ALSO TEST CLEAR
(4)                                     ;;*****************************************************************
(3)  005472  000004                     TST32:  SCOPE
(1)
(1)  005474  005077  173662                     CLR     @IBS            ;/CLEAR CSR.
(1)  005500  052777  000010  173654             BIS     #BIT3,@IBS      ;/SET IBC, SHOULD SET IFC.
(1)  005506  032777  020000  173650             BIT     #BIT13,@IBD     ;/DID IFC SET?
(1)  005514  001011                             BNE     1$              ;/YES - LETS TRY CLEARING IT.
(1)  005516  052777  000010  173636             BIS     #BIT3,@IBS      ;/SET IBC, MEMORY
(1)                                                                     ;/REFRESH COULD HAVE
(1)                                                                     ;/INTERRUPTED US.
(1)  005524  032777  020000  173632             BIT     #BIT13,@IBD     ;/DID IFC SET THIS TIME?
(1)  005532  001002                             BNE     1$
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  005534  104002                             ERROR   2               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/IFC FAILED TO SET WHEN IBC SET.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(3)  005536  000410                             BR      TST33           ;;
(1)
(1)  005540                             1$:
(1)  005540  004537  010100             2$:    JSR     R5,DEL50        ;DELAY SOME TIME
(1)  005544  000006                             .WORD   6
(1)
(1)  005546  032777  020000  173610             BIT     #BIT13,@IBD     ;/IBC CLEAR, DID IFC CLEAR?
(3)  005554  001401                             BEQ     TST33           ;;
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  005556  104002                             ERROR   2               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/IFC FAILED TO CLEAR.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

(1)
749
```

```
751
(1)
(5)                                     ;;*****************************************************************
(4)                                     ;*TEST 33       *TEST THAT IBD - ATN SETS WHEN IBS - TCS SETS, ALSO TEST CLEAR
(4)                                     ;;*****************************************************************
(3)  005560  000004                     TST33:  SCOPE
(1)
(1)  005562  005077  173574                     CLR     @IBS            ;/CLEAR CSR.
(1)  005566  052777  000001  173566             BIS     #BIT0,@IBS      ;/SET TCS, SHOULD SET ATN.
(1)  005574  032777  040000  173562             BIT     #BIT14,@IBD     ;/DID ATN SET?
(1)  005602  001011                             BNE     1$              ;/YES - LETS TRY CLEARING IT.
(1)  005604  052777  000001  173550             BIS     #BIT0,@IBS      ;/SET TCS, MEMORY
(1)                                                                     ;/REFRESH COULD HAVE
(1)                                                                     ;/INTERRUPTED US.
(1)  005612  032777  040000  173544             BIT     #BIT14,@IBD     ;/DID ATN SET THIS TIME?
(1)  005620  001002                             BNE     1$
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  005622  104002                             ERROR   2               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/ATN FAILED TO SET WHEN TCS SET.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(3)  005624  000410                             BR      TST34           ;;
(1)  005626  042777  000001  173526  1$:        BIC     #BIT0,@IBS      ;/CLEAR TCS, SHOULD CLEAR ATN.
(1)  005634  032777  040000  173522             BIT     #BIT14,@IBD     ;/DID ATN CLEAR?
(3)  005642  001401                             BEQ     TST34           ;;
(2)

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  005644  104002                             ERROR   2               ;/MODULE FAULT DETECTED:
(1)                                                                     ;/ATN FAILED TO CLEAR.

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

(1)
752
```

```
754
(1)
(5)                                 ;;*****************************************************************
(4)                                 ;*TEST 34       *TEST THAT IBD - EOI SETS WHEN IBS - EOP SETS, ALSO TEST CLEAR
(4)                                 ;;*****************************************************************
(3)  005646  000004                 TST34:  SCOPE
(1)
(1)  005650  005077  173506                 CLR     @IBS            ;/CLEAR CSR.
(1)  005654  052777  000002  173500         BIS     #BIT1,@IBS      ;/SET EOP, SHOULD SET EOI.
(1)  005662  032777  100000  173474         BIT     #BIT15,@IBD     ;/DID EOI SET?
(1)  005670  001011                         BNE     1$              ;/YES - LETS TRY CLEARING IT.
(1)  005672  052777  000002  173462         BIS     #BIT1,@IBS      ;/SET EOP, MEMORY
(1)                                                                 ;/REFRESH COULD HAVE
(1)                                                                 ;/INTERRUPTED US.
(1)  005700  032777  100000  173456         BIT     #BIT15,@IBD     ;/DID EOI SET THIS TIME?
(1)  005706  001002                         BNE     1$
(2)

                                    ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  005710  104002                         ERROR   2               ;/MODULE FAULT DETECTED:
(1)                                                                 ;/EOI FAILED TO SET WHEN EOP SET.

                                    ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
(3)  005712  000410                 BR      TST35           ;;
(1)  005714  042777  000002  173440  1$:     BIC     #BIT1,@IBS      ;/CLEAR EOP, SHOULD CLEAR EOI.
(1)  005722  032777  100000  173434         BIT     #BIT15,@IBD     ;/DID EOI CLEAR?
(3)  005730  001401                         BEQ     TST35           ;;
(2)

                                    ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(2)
(2)  005732  104002                         ERROR   2               ;/MODULE FAULT DETECTED:
(1)                                                                 ;/EOI FAILED TO CLEAR.

                                    ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

(1)
755
```

```
                                        ;;*****************************************************************
                                        ;*TEST 35        *TEST THAT IBD-SRQ SETS WHEN IBS-SRQ SETS, ALSO TEST CLEAR (IF SWR10=1)
                                        ;;*****************************************************************
005734  000004                  TST35:  SCOPE
005736  005077  173420                  CLR     @IBS            ;CLEAR CSR
005742  052777  100000  173412          BIS     #BIT15,@IBS     ;SET SRQ IN IBS SHOULD SET SRQ IN IBD
005750  032777  002000  173162          BIT     #BIT10,@SWR     ;TEST IF 'ER1-INHIBIT" SWITCH IS SET <CLOSED-ON>
005756  001406                          BEQ     1$              ;;BR IF SWITCH IS CLEARED
005760  032777  004000  173376          BIT     #BIT11,@IBD     ;DID SRQ IN IBD SET THIS TIME ?
005766  001002                          BNE     1$              ;;BR IF YES

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  005770  104002                    ERROR   2               ;/MODULE FAULT DETECTED:
                                                                ;SRQ IN IBD DID NOT SET WHEN SRQ IN IBS SET
                                                                ; CHECK ER1-INHIBIT SWITCH

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
005772  000410                          BR      2$              ;;
005774  042777  100000  173360  1$:     BIC     #BIT15,@IBS     ;REMOVE IBS-SRQ BIT
006002  032777  004000  173354          BIT     #BIT11,@IBD     ;DID SRQ IN IBD STAY CLEARED ?
006010  001401                          BEQ     2$              ;;BR IF STILL CLEARED

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  006012  104002                    ERROR   2               ;/MODULE FAULT DETECTED:
                                                                ;SRQ IN IBD WAS SET WHEN SRQ IN IBS CLEARED
                                                                ; CHECK ER-1 INHIBIT SWITCH

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
006014  005077  173342          2$:     CLR     @IBS            ;ENSURE CLEARED STATUS
```

```
                                ;;*****************************************************************
                                ;*TEST 36        *TEST THAT RFD SET WHEN CSR CLEAR, CLEAR WHEN ACC SET
                                ;;*****************************************************************
(2)  006020  000004             TST36:  SCOPE

006022  005077  173334                  CLR     @IBS            ;CLEAR CSR.
006026  032777  002000  173330          BIT     #BIT10,@IBD     ;DID RFD SET?
006034  001002                          BNE     1$              ;YES CONTINUE.

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  006036  104002                     ERROR   2               ;/MODULE FAULT DETECTED:
                                                                ;RFD FAILED TO SET.

                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
006040  000410                          BR      TST37           ;;

006042  052777  000200  173312  1$:     BIS     #BIT7,@IBS      ;NOW SET ACC,RFD SHOULD CLEAR.
006050  032777  002000  173306          BIT     #BIT10,@IBD     ;DID IT CLEAR?
006056  001401                          BEQ     TST37           ;;

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  006060  104002                     ERROR   2               ;/MODULE FAULT DETECTED:
                                                                ;RFD FAILED TO CLEAR.

                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$


                                ;;*****************************************************************
                                ;*TEST 37        *TEST THAT WE CAN GENERATE AN ER2
                                ;;*****************************************************************
(2)  006062  000004             TST37:  SCOPE

006064  005077  173272                  CLR     @IBS            ;CLEAR THE STATUS REG.
006070  052777  000041  173264          BIS     #BIT5!BIT0,@IBS ;SET TON; THIS SHOULD CAUSE AN
                                                                ;ERROR SENCE NO LISTENERS ARE ON
006076  105077  173262                  CLRB    @IBD            ;AND WE SENT DATA TO THE BUS.
                                                                ;BUS.
006102  032777  040000  173252          BIT     #BIT14,@IBS     ;DID ER2 SET?
006110  001001                          BNE     TST40           ;;

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  006112  104001                     ERROR   1               ;/MODULE FAULT DETECTED:
                                                                ;ER2 FAILED TO SET.

                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
```

```
                                ;;*****************************************************************
                                ;*TEST 40        *TEST THAT BUS INIT CLEARS ACC,TON,LON,REM,EIP,TCS
                                ;;*****************************************************************
006114  000004                  TST40:  SCOPE
006116  012737  000005  001160          MOV     #5,$TIMES       ;;DO 5 ITERATIONS

006124  012777  000367  173230          MOV     #367,@IBS       ;SET ACC,TON,LON,REM,EOP, AND TCS.
006132  000005                          RESET                   ;ISSUE SYS INIT.
006134  105777  173222                  TSTB    @IBS            ;DID THEY ALL CLEAR?
006140  001401                          BEQ     TST41           ;;

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
006142  104001                          ERROR   1               ;/MODULE FAULT DETECTED:
                                                                ;BUS INIT FAILED TO CLEAR CSR.

                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$

                                ;;*****************************************************************
                                ;*TEST 41        *TEST IBC CLEARS ACC,TON,LON,REM AND EOP
                                ;;*****************************************************************
006144  000004                  TST41:  SCOPE
006146  012737  000005  001160          MOV     #5,$TIMES       ;;DO 5 ITERATIONS

006154  012777  000266  173200          MOV     #266,@IBS       ;SET ACC,TON,LON,REM, AND EOP.
006162  052777  000010  173172          BIS     #BIT3,@IBS      ;SET IBC, THIS SHOULD CLEAR ABOVE BITS.
006170  004537  010100                  JSR     R5,DEL50        ;DELAY 125 USEC
006174  000006                          .WORD   6
006176  032777  000266  173156          BIT     #266,@IBS       ;DID THEY CLEAR?
006204  001401                          BEQ     TST42           ;;

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
006206  104001                          ERROR   1               ;/MODULE FAULT DETECTED:
                                                                ;ACC,TON,LON,REM, AND/OR EOP
                                                                ;FAILED TO CLEAR ON IBC

                                ;;*****************************************************************
                                ;*TEST 42        *TEST THAT BUS INIT INDIRECTLY CLEARS IBD
                                ;;*****************************************************************
006210  000004                  TST42:  SCOPE
006212  012737  000005  001160          MOV     #5,$TIMES       ;;DO 5 ITERATIONS

006220  012777  000260  173134          MOV     #BIT7!BIT5!BIT4,@IBS ;SET ACC,TON, AND LON.
006226  012777  000377  173130          MOV     #377,@IBD       ;LOAD IBD
006234  000005                          RESET                   ;ISSUE SYS INIT.
006236  105777  173122                  TSTB    @IBD            ;DID IT CLEAR?
006242  001401                          BEQ     TST43           ;;

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$
```

```
(1)
(1)  006244  104002                        ERROR   2               ;/MODULE FAULT DETECTED:
842                                                                 ;FAILED TO CLEAR LOW BYTE OF IBD ON
843                                                                 ;SYSTEM INIT.

                                            ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
845
846                                 .SBTTL
847                                 .SBTTL          INTERRUPT TESTS
848                                 .SBTTL
849
850                                 ;;*****************************************************************
(3)                                 ;*TEST 43        *TEST THAT CMD CAN GENERATE AN INTERRUPT
(3)                                 ;;*****************************************************************
(2)  006246  000004                 TST43:  SCOPE
851
852  006250  005077  173106                 CLR     @IBS            ;CLEAR THE CSR.
853  006254  012777  000200  173140         MOV     #200,@PRC
854  006262  012777  006340  173106         MOV     #1$,@VECTC      ;SET UP INTERRUPT VECTOR
855  006270  052777  000101  173064         BIS     #BIT0!BIT6,@IBS ;SET TCS, SHOULD CAUSE
856                                                                 ;/PR
(1)  006276  012746  000000                 MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
(1)  006302  012746  006310                 MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
(1)  006306  000002                         RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
(1)  006310                         64$:                            ;/IN STATUS REG.)
857  006310  000240                         NOP                     ;CMD TO SET AND GIVE US AN
858  006312  000240                         NOP                     ;INTERRUPT.
859  006314  000240                         NOP
860  006316  000240                         NOP
861  006320  000240                         NOP
862  006322  000240                         NOP
863  006324  000240                         NOP
864  006326  000240                         NOP
865  006330  000240                         NOP
866  006332  000240                         NOP
867

                                            ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  006334  104001                        ERROR   1               ;/MODULE FAULT DETECTED:
868                                                                 ;CMD FAILED TO GENERATE AN INTERRUPT.

                                            ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
870  006336  000402                         BR      2$
871  006340                      1$:
(1)  006340  062706  000004                 ADD     #4,SP           ;/ADD #4 TO ST. K POINTER.
872  006344  005077  173012         2$:     CLR     @IBS            ;CLEAR INTERRUPT
873                                                                 ;/-RESV-
(1)  006350  013777  001422  173020         MOV     PRC,@VECTC ;/RESTORE VECTOR FOR
(1)  006356  012777  004700  173036         MOV     #4700,@PRC      ;/ILLEGAL INTRO.
874
875                                 ;;*****************************************************************
(3)                                 ;*TEST 44        *TEST THAT TKR AND LNR CAN GENERATE INTERRUPTS
(3)                                 ;;*****************************************************************
```

```
    (2)  006364  000004                    TST44:  SCOPE
   876   006366  012777  000200  173026            MOV     #200,@PRC
   877   006374  012777  006500  172774            MOV     #1$,@VECTC      ;SET UP INTERRUPT VECTOR FOR TKR INTERRUPT
   878   006402  012777  000060  172752            MOV     #BIT4!BIT5,@IBS ;SET TON AND LON
   879   006410  052777  000100  172744            BIS     #BIT6,@IBS      ;ALLOW INTERRUPT
   880                                                                       ;/PR
   (1)   006416  012746  000000                    MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
   (1)   006422  012746  006430                    MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
   (1)   006426  000002                            RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
   (1)   006430                            64$:                            ;/IN STATUS REG.)
   881   006430  000240                            NOP
   882   006432  000240                            NOP
   883   006434  000240                            NOP
   884   006436  000240                            NOP
   885   006440  000240                            NOP
   886   006442  000240                            NOP
   887   006444  000240                            NOP
   888   006446  000240                            NOP
   889   006450  000240                            NOP
   890   006452  000240                            NOP
   891

                                                   ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

   (1)
   (1)   006454  104001                            ERROR   1               ;/MODULE FAULT DETECTED:
   892                                                                       ;FAILED TO GENERATE A TKR INTERRUPT.

                                                   ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
   894   006456  005077  172700                    CLR     @IBS            ;CLR CSR
   895                                                                       ;/-RESV-
   (1)   006462  013777  001422  172706            MOV     PRC,@VECTC ;/RESTORE VECTOR FOR
   (1)   006470  012777  004700  172724            MOV     #4700,@PRC      ;/ILLEGAL INTRO.
   896   006476  000453                            BR      TST45           ;;
   897
   898   006500                            1$:
   (1)   006500  062706  000004                    ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
   899                                                                       ;/-RESV-
   (1)   006504  013777  001422  172664            MOV     PRC,@VECTC ;/RESTORE VECTOR FOR
   (1)   006512  012777  004700  172702            MOV     #4700,@PRC      ;/ILLEGAL INTRO.
   900   006520  012777  000200  172676            MOV     #200,@PRD
   901   006526  012777  006602  172644            MOV     #2$,@VECTD      ;SET UP FOR LNR INTERRUPT.
   902   006534  105277  172624                    INCB    @IBD            ;SEND DATA - CLRS TKR SETS LNR
   903                                                                       ;FOR INTERRUPT
   904                                                                       ;/PR
   (1)   006540  012746  000000                    MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
   (1)   006544  012746  006552                    MOV     #65$,-(SP)      ;/SHOW RETURN ADDRESS
   (1)   006550  000002                            RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
   (1)   006552                            65$:                            ;/IN STATUS REG.)
   905   006552  000240                            NOP
   906   006554  000240                            NOP
   907   006556  000240                            NOP
   908   006560  000240                            NOP
   909   006562  000240                            NOP
   910   006564  000240                            NOP
   911   006566  000240                            NOP
```

```
CVIBA-B MACY11 30G(1063)  25-FEB-83  08:23  PAGE 30-3
CVIBAB.P11    13-DEC-82 09:29          T44     *TEST THAT TKR AND LNR CAN GENERATE INTERRUPTS                  SEQ 0057

   912  006570  000240                         NOP
   913  006572  000240                         NOP
   914  006574  000240                         NOP
   915

                                               ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

   (1)
   (1)  006576  104001                         ERROR   1               ;/MODULE FAULT DETECTED:
   916                                                                  ;FAILED TO GENERATE LNR INTERRUPT

                                               ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
   918  006600  000402                         BR      3$
   919
   920  006602                          2$:
   (1)  006602  062706  000004                 ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
   921  006606  005077  172550          3$:    CLR     @IBS            ;CLEAR THE STATUS REG.
   922                                                                  ;/-RESV-
   (1)  006612  013777  001424  172560         MOV     PRD,@VECTD ;/RESTORE VECTOR FOR
   (1)  006620  012777  004700  172576         MOV     #4700,@PRD      ;/ILLEGAL INTRO.
   923
```

```
925                                     ;;*****************************************************************
(3)                                     ;*TEST 45       *TEST THAT ER2 CAN GENERATE AN INTERRUPT
(3)                                     ;*****************************************************************
(2)  006626  000004                     TST45:  SCOPE
926
927  006630  005077  172526                    CLR     @IBS            ;START WITH CSR CLEAR
928  006634  012777  000200  172554            MOV     #200,@PRA
929  006642  012777  006734  172522            MOV     #1$,@VECTA      ;SET UP INTERRUPT VECTOR
930                                                                     ;/PR
(1)  006650  012746  000200                    MOV     #200,-(SP)      ;/SET CPU PRIORITY ON RETURN
(1)  006654  012746  006662                    MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
(1)  006660  000002                            RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
(1)  006662                             64$:                            ;/IN STATUS REG.)
931  006662  052777  000140  172472            BIS     #BIT5!BIT6,@IBS ;SET TON - NO LISTNERS ON
932  006670  105077  172470                    CLRB    @IBD            ;BUS BUT DATA PUT ON
933  006674  000240                            NOP                     ;BUS - THEREFORE AN INTERRUPT
934  006676  000240                            NOP                     ;SHOULD BE POSTED.
935                                                                     ;/PR
(1)  006700  012746  000000                    MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
(1)  006704  012746  006712                    MOV     #65$,-(SP)      ;/SHOW RETURN ADDRESS
(1)  006710  000002                            RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
(1)  006712                             65$:                            ;/IN STATUS REG.)
936  006712  000240                            NOP
937  006714  000240                            NOP
938  006716  000240                            NOP
939  006720  000240                            NOP
940  006722  000240                            NOP
941  006724  000240                            NOP
942  006726  000240                            NOP
943
944

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  006730  104001                            ERROR   1               ;/MODULE FAULT DETECTED:
945                                                                     ;FAILED TO INTERRUPT ON ERROR2

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
947  006732  000402                            BR      2$
948  006734                          1$:
(1)  006734  062706  000004                    ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
949  006740  005077  172416          2$:       CLR     @IBS            ;CLEAR CSR
950                                                                     ;/-RESV-
(1)  006744  013777  001416  172420            MOV     PRA,@VECTA ;/RESTORE VECTOR FOR
(1)  006752  012777  004700  172436            MOV     #4700,@PRA      ;/ILLEGAL INTRO.
951
```

```
953                                     ;;*****************************************************************
(3)                                     ;*TEST 46       *TEST THAT SRQ CAN GENERATE AN INTERRPUT (SWR10=1)
(3)                                     ;;*****************************************************************
(2)  006760  000004                     TST46:  SCOPE
954  006762  005077  172374                       CLR     @IBS                    ;START WITH CSR CLEAR
955  006766  032777  002000  172144               BIT     #BIT10,@SWR             ;TEST IF 'ER1 INH" SWITCH IS SET
956  006774  001450                              BEQ     TST47                   ;;BR IF OPERATOR SAID YES
957                                                                              ;/PR
(1)  006776  012746  000200                       MOV     #200,-(SP)              ;/SET CPU PRIORITY ON RETURN
(1)  007002  012746  007010                       MOV     #64$,-(SP)              ;/SHOW RETURN ADDRESS
(1)  007006  000002                               RTI                             ;/CAUSE A RETURN (PUTS NEW STATUS
(1)  007010                             64$:                                     ;/IN STATUS REG.)
958  007010  012777  000200  172402               MOV     #200,@PRB
959  007016  012777  007072  172350               MOV     #1$,@VECTB              ;SET UP SRQ INTERRPUT VECTOR
960  007024  052777  100100  172330               BIS     #BIT6!BIT15,@IBS        ;SET SRQ AND IE
961  007032  000240                               NOP
962  007034  000240                               NOP
963  007036  000240                               NOP                             ;AN INTERRUPT SHOULD BE POSTED
964                                                                              ;/PR
(1)  007040  012746  000000                       MOV     #0,-(SP)                ;/SET CPU PRIORITY ON RETURN
(1)  007044  012746  007052                       MOV     #65$,-(SP)              ;/SHOW RETURN ADDRESS
(1)  007050  000002                               RTI                             ;/CAUSE A RETURN (PUTS NEW STATUS
(1)  007052                             65$:                                     ;/IN STATUS REG.)
965  007052  000240                               NOP
966  007054  000240                               NOP
967  007056  000240                               NOP
968  007060  000240                               NOP
969  007062  000240                               NOP
970  007064  000240                               NOP
971

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  007066  104001                               ERROR   1                       ;/MODULE FAULT DETECTED:
972                                                                              ;FAILED TO INTERRUPT ON SRQ
973                                                                              ;IS SWR10 CORRECT ?

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
975  007070  000402                               BR      2$
976  007072                             1$:
(1)  007072  062706  000004                       ADD     #4,SP                   ;/ADD #4 TO STACK POINTER.
977  007076  005077  172260             2$:       CLR     @IBS                    ;CLEAR CSR
978                                                                              ;/-RESV-
(1)  007102  013777  001420  172264               MOV     PRB,@VECTB ;/RESTORE VECTOR FOR
(1)  007110  012777  004700  172302               MOV     #4700,@PRB              ;/ILLEGAL INTRO.
979
980                                     .SBTTL
981                                     .SBTTL  SECOND MODULE TESTS
982                                     .SBTTL
```

```
                                ;;*****************************************************************
                                ;*TEST 47       *TEST THAT MODULE PASSES 'BIAKI"
                                ;;*****************************************************************
(2)  007116  000004             TST47:  SCOPE

                                        ;*WARNING! THIS TEST IS DESIGNED TO BE EXERCISED WITH A
                                        ;*SECOUND MODULE (IBV-11) WITH SWITCH 'ER1 INH" SET (CLOSED/ON).
                                        ;*ADDRESS OF THE SECOUND MODULE IS IN LOCATION "IBS2" VECTOR
                                        ;*ADDRESS IS IN LOCATION 'VECTA2".
                                        ;* THE SECOUND IBV-11 SHOULD BE ELECTRICALLY SECOUND ON Q BUSS.
                                        ;*TO INHIBIT THE USE OF TESTING WITH A SECOUND MODULE, MAKE
                                        ;*LOCATION "$CDW1" ZERO.
                                        ;*THE SWITCH 'ER1 INH" SHOULD BE CLEARED (OPEN/OFF) FOR THE
                                        ;*MODULE UNDER TEST
                                        ;*
 999  007120  005737  001250           TST     $CDW1           ;TESTING WITH
1000  007124  001002                   BNE     3$              ;SECOUND IBV11?
1001  007126  000137  007756           JMP     EOP             ;NO-END PASS.
1002  007132  005077  172224     3$:   CLR     @IBS            ;CLEAR CSR.
1003  007136  005077  172240           CLR     @IBS2           ;CLEAR SECOUND MODULE.
1004  007142  012777  000200  172262   MOV     #200,@PRC2
1005  007150  012777  007206  172234   MOV     #1$,@VECTC2     ;SET UP VECTOR ADDR.
1006                                                           ;/PR
 (1)  007156  012746  000000           MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
 (1)  007162  012746  007170           MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
 (1)  007166  000002                   RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
 (1)  007170                     64$:                          ;/IN STATUS REG.)
1007  007170  012777  000140  172204   MOV     #BIT6!BIT5,@IBS2 ;SET INTR ENABLE AND TON ON SECOUND
1008  007176  000240                   NOP                     ;IBV - SHOULD CAUSE A TKR  INTERRUPT.
1009  007200  000240                   NOP
1010

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

 (1)
 (1)  007202  104001                   ERROR   1               ;/MODULE FAULT DETECTED:
1011                                                           ;ASSUMING SECOUND MODULE IS GOOD,
1012                                                           ;MODULE (IBV-11) UNDER TEST FAILED
1013                                                           ;TO PASS Q BUSS SIGNAL 'BIAKI"

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
1015  007204  000402                   BR      2$
1016  007206                    1$:
 (1)  007206  062706  000004           ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
1017  007212  005077  172164     2$:   CLR     @IBS2           ;CLEAR SECOUND MODULE
1018                                                           ;/-RESV-
 (1)  007216  013777  001432  172166   MOV     PRC2,@VECTC2 ;/RESTORE VECTOR FOR
 (1)  007224  012777  004700  172200   MOV     #4700,@PRC2     ;/ILLEGAL INTRO.
1019
```

```
                        ;;*****************************************************************
                        ;*TEST 50       *TEST THAT SRQ CAN GENERATE AN INTERRUPT
                        ;;*****************************************************************
007232  000004          TST50:  SCOPE
                        ;*WARNING! THIS TEST IS DESIGNED TO BE EXERCISED WITH A
                        ;*SECOUND MODULE (IBV-11) WITH SWITCH 'ER1 INH' SET (CLOSED/ON).
                        ;*ADDRESS OF THE SECOUND MODULE IS IN LOCATION 'IBS2' VECTOR
                        ;*ADDRESS IS IN LOCATION 'VECTA2'.
                        ;* THE SECOUND IBV-11 SHOULD BE ELECTRICALLY SECOUND ON Q BUSS.
                        ;*TO INHIBIT THE USE OF TESTING WITH A SECOUND MODULE, MAKE
                        ;*LOCATION '$CDW1' ZERO.
                        ;*THE SWITCH 'ER1 INH' SHOULD BE CLEARED (OPEN/OFF) FOR THE
                        ;*MODULE UNDER TEST
                        ;*
007234  005077  172122          CLR     @IBS            ;CLEAR CSRS.
007240  005077  172136          CLR     @IBS2

007244  012777  000200  172146  MOV     #200,@PRB       ;SET UP INTERRUPT VECTOR.
007252  012777  007316  172114  MOV     #1$,@VECTB
                                                        ;/PR
007260  012746  000000          MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
007264  012746  007272          MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
007270  000002                  RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
007272                  64$:                            ;/IN STATUS REG.)
007272  012777  000100  172062  MOV     #100,@IBS       ;ENABLE INTERRUPTS
007300  052777  100000  172074  BIS     #BIT15,@IBS2    ;SETTING SRQ IN THE ''CDW'' MODULE
                                                        ;WILL PUT SRQ ON THE IB BUS
                                                        ;IS ER1 INH SW IS SET.
007306  000240          NOP
007310  000240          NOP

                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

007312  104001          ERROR   1               ;/MODULE FAULT DETECTED:
                                                ;SRQ FAILED TO GENERATE AN INTERRUPT
                                        ;THIS ERROR OCCURS IS THE CABLE BETWEEN UNITS IS MISSING
                                        ; OR IF S1-8 ON THE KGM IS CLEARED (OPEN-OFF)

                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
007314  000402          BR      2$

007316                  1$:
007316  062706  000004  ADD     #4,SP           ;/ADD #4 TO STACK POINTER.

007322                  2$:
                                                ;/-RESV-
007322  013777  001420  172044  MOV     PRB,@VECTB ;/RESTORE VECTOR FOR
007330  012777  004700  172062  MOV     #4700,@PRB      ;/ILLEGAL INTRO.
007336  005077  172020          CLR     @IBS            ;CLEAR CSRS
007342  005077  172034          CLR     @IBS2
```

```
                                ;;*****************************************************************
                                ;*TEST 51       *TEST THAT ERROR 1 IS GENERATED IF ATN IS ON THE IB BUS
                                ;;*****************************************************************
007346  000004                  TST51:  SCOPE
                                ;*WARNING! THIS TEST IS DESIGNED TO BE EXERCISED WITH A
                                ;*SECOUND MODULE (IBV-11) WITH SWITCH 'ER1 INH' SET (CLOSED/ON).
                                ;*ADDRESS OF THE SECOUND MODULE IS IN LOCATION 'IBS2' VECTOR
                                ;*ADDRESS IS IN LOCATION 'VECTA2'.
                                ;* THE SECOUND IBV-11 SHOULD BE ELECTRICALLY SECOUND ON Q BUSS.
                                ;*TO INHIBIT THE USE OF TESTING WITH A SECOUND MODULE, MAKE
                                ;*LOCATION '$CDW1' ZERO.
                                ;*THE SWITCH 'ER1 INH' SHOULD BE CLEARED (OPEN/OFF) FOR THE
                                ;*MODULE UNDER TEST
                                ;*
007350  005077  172026          CLR     @IBS2           ;CLR CSR OF 2ND MODULE.
007354  032777  002000  171556  BIT     #BIT10,@SWR     ;TEST IF 'ER1 INH' SWITCH IS SET (CLOSED/ON)
007362  001407                  BEQ     1$              ;BR IF 'ER1 INH' SWITCH IS CLEARED
007364  005737  001176          TST     $PASS           ;TEST IF FIRST PASS
007370  001002                  BNE     2$              ;BR IF NOT
007372  104401  014303          TYPE    ,WARN1          ;INFORM OPERATOR ABOUT MISSING SEVERAL TESTS
007376  000137  007756          2$:     JMP     EOP     ;BYPASS REMAINDER OF TEST
007402  005277  171774          1$:     INC     @IBS2   ;ASSERT ATN ON IB BUS
                                                        ;ASSERTED ATN ON IBV UNDER TEST-
                                                        ;THIS SHOULD CAUSE AN ERROR 1
                                                        ;BECAUSE THE 2ND IBV HAS ATN SET.
007406  032777  020000  171746          BIT     #BIT13,@IBS     ;DID ERROR 1 SET?
007414  001001                  BNE     TST52           ;;

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

                                ERROR   1               ;/MODULE FAULT DETECTED:
007416  104001                                          ;FAILED TO GENERATE ERROR 1
```

```
                                        ;;*****************************************************************
                                        ;*TEST 52       *TEST THAT ERROR 1 IS GENERATED IF IFC IS ON IB BUS BY SECOUND MODULE
                                        ;;*****************************************************************
                                        TST52:  SCOPE
007420  000004                                  MOV     #5,$TIMES       ;;DO 5 ITERATIONS
007422  012737  000005  001160                   ;*WARNING! THIS TEST IS DESIGNED TO BE EXERCISED WITH A
                                                ;*SECOUND MODULE (IBV-11) WITH SWITCH "ER1 INH" SET (CLOSED/ON).
                                                ;*ADDRESS OF THE SECOUND MODULE IS IN LOCATION "IBS2" VECTOR
                                                ;*ADDRESS IS IN LOCATION "VECTA2".
                                                ;* THE SECOUND IBV-11 SHOULD BE ELECTRICALLY SECOUND ON Q BUSS.
                                                ;*TO INHIBIT THE USE OF TESTING WITH A SECOUND MODULE, MAKE
                                                ;*LOCATION "$CDW1" ZERO.
                                                ;*THE SWITCH "ER1 INH" SHOULD BE CLEARED (OPEN/OFF) FOR THE
                                                ;*MODULE UNDER TEST
                                                ;*
007430  005077  171726                           CLR     @IBS            ;CLEAR CSR
007434  012777  000010  171740                   MOV     #BIT3,@IBS2     ;ASSERT IFC FROM TESTOR
007442  032777  020000  171712                   BIT     #BIT13,@IBS     ;DID ERROR 1 GET SET?
                                                                        ;IF SO - NEXT TEST
007450  001010                                   BNE     TST53           ;;
007452  012777  000010  171722                   MOV     #BIT3,@IBS2     ;IF NOT WE'LL TRY AGAIN SENCE MEMORY
007460  032777  020000  171674                   BIT     #BIT13,@IBS     ;REFRESH COULD HAVE GO IN THE WAY.
007466  001001                                   BNE     TST53           ;;

                                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1)  007470  104001                     ERROR   1               ;/MODULE FAULT DETECTED:
                                                                ;ERROR 1 FAILED TO SET WHEN
                                                                ;IFC WAS ON IB-BUS AND MODULE
                                                                ;UNDER TEST DIDN'T PUT IT THERE.
```

```
                                ;;*****************************************************************
                                ;*TEST 53       *TEST THAT ERROR 1 IS GENERATED IF REN IS ON IB BUS
                                ;;*****************************************************************
007472  000004                TST53:  SCOPE

                                ;*WARNING! THIS TEST IS DESIGNED TO BE EXERCISED WITH A
                                ;*SECOUND MODULE (IBV-11) WITH SWITCH 'ER1 INH' SET (CLOSED/ON).
                                ;*ADDRESS OF THE SECOUND MODULE IS IN LOCATION 'IBS2' VECTOR
                                ;*ADDRESS IS IN LOCATION 'VECTA2'.
                                ;* THE SECOUND IBV-11 SHOULD BE ELECTRICALLY SECOUND ON Q BUSS.
                                ;*TO INHIBIT THE USE OF TESTING WITH A SECOUND MODULE, MAKE
                                ;*LOCATION '$CDW1' ZERO.
                                ;*THE SWITCH 'ER1 INH' SHOULD BE CLEARED (OPEN/OFF) FOR THE
                                ;*MODULE UNDER TEST
                                ;*

007474  005077  171662          CLR     @IBS            ;CLEAR CSRS.
007500  005077  171676          CLR     @IBS2
007504  052777  000004  171670  BIS     #BIT2,@IBS2     ;ASSERT REN ON IB BUS FROM 2ND
                                                        ;MODULE. 1ST IBV-11 SHOULD
007512  032777  020000  171642  BIT     #BIT13,@IBS     ;GENERATE AN ERROR 1;DID IT??
007520  001001                  BNE     TST54           ;;

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

007522  104001                  ERROR   1               ;/MODULE FAULT DETECTED:
                                                        ;FAILED TO GENERATE AN ERROR 1.
                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
```

```
1098                                    ;;*********************************************************************
 (3)                                    ;*TEST 54       *TEST THAT AN ERROR 1 CAN GENERATE AN INTERRUPT
 (3)                                    ;;*********************************************************************
 (2) 007524  000004                     TST54:  SCOPE
1099
1100                                            ;*WARNING! THIS TEST IS DESIGNED TO BE EXERCISED WITH A
 (1)                                            ;*SECOUND MODULE (IBV-11) WITH SWITCH 'ER1 INH'' SET (CLOSED/ON).
 (1)                                            ;*ADDRESS OF THE SECOUND MODULE IS IN LOCATION ''IBS2'' VECTOR
 (1)                                            ;*ADDRESS IS IN LOCATION 'VECTA2".
 (1)                                            ;* THE SECOUND IBV-11 SHOULD BE ELECTRICALLY SECOUND ON Q BUSS.
 (1)                                            ;*TO INHIBIT THE USE OF TESTING WITH A SECOUND MODULE, MAKE
 (1)                                            ;*LOCATION '$CDW1'' ZERO.
 (1)                                            ;*THE SWITCH 'ER1 INH'' SHOULD BE CLEARED (OPEN/OFF) FOR THE
 (1)                                            ;*MODULE UNDER TEST
 (1)                                            ;*
1101
1102 007526  005077  171650                     CLR     @IBS2           ;CLEAR CSRS.
1103 007532  005077  171624                     CLR     @IBS
1104
1105 007536  012777  000200  171652             MOV     #200,@PRA
1106 007544  012777  007610  171620             MOV     #1$,@VECTA      ;SET UP VECTOR ADDR.
1107
1108 007552  052777  000100  171602             BIS     #BIT06,@IBS     ;SET INTERRUPT ENABLE
1109                                                                     ;/PR
 (1) 007560  012746  000000                     MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
 (1) 007564  012746  007572                     MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
 (1) 007570  000002                             RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
 (1) 007572                             64$:                            ;/IN STATUS REG.)
1110 007572  052777  000004  171602             BIS     #BIT2,@IBS2     ;GENERATE AN ERROR 1 AS PER LAST TEST.
1111 007600  000240                             NOP
1112 007602  000240                             NOP
1113

                                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

 (1)
 (1) 007604  104001                             ERROR   1               ;/MODULE FAULT DETECTED:
1114                                                                    ;ERROR 1 FAILED TO GENERATE AN INTR.

                                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
1116 007606  000402                             BR      2$
1117
1118 007610                             1$:
 (1) 007610  062706  000004                     ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
1119 007614                             2$:                             ;/-RESV-
 (1)
 (1) 007614  013777  001416  171550             MOV     PRA,@VECTA ;/RESTORE VECTOR FOR
 (1) 007622  012777  004700  171566             MOV     #4700,@PRA      ;/ILLEGAL INTRO.
1120 007630  005077  171546                     CLR     @IBS2           ;CLEAR CSRS.
1121 007634  005077  171522                     CLR     @IBS
;122
1133
```

```
                                ;;*****************************************************************
                                ;*TEST 55        *TEST THAT DATA CAN BE XFERRED BETWEEN THE MODULE UNDER TEST AND THE KGM
                                ;*              NOTE: KGM =KNOWN GOOD MODULE
                                ;*IN THIS TEST WE'LL MAKE THE KGM A LISTENER
                                ;* AND THE MODULE UNDER TEST A TALKER.
                                ;*WE'VE ALREADY XFERRED DATA TO AND FROM THE IB-BUS
                                ;*VIA THE MODULE UNDER TEST. THE ONLY UNKNOWN
                                ;*IS THE CABLE CONNECTING THE KGM TO THE MODULE UNDER TEST,
                                ;*AS WELL AS THE KGM.
                                ;*
                                ;;*****************************************************************
(2) 007640  000004              TST55:  SCOPE

1136
1137 007642  012737  007662  0 1110      MOV     #1$,$LPERR      ;SET ERROR LOOP.
1138 007650  012737  000000  001124      MOV     #0,$GDDAT       ;START PATTERN.
1139 007656  005037  001126              CLR     $BDDAT
1140
1141 007662  005077  171474      1$:     CLR     @IBS            ;CLEAR CSRS.
1142 007666  005077  171510              CLR     @IBS2
1143 007672  052777  000041  171462      BIS     #BIT5!BIT0,@IBS ;SET TON AND TCS.
1144 007700  052777  000020  171474      BIS     #BIT4,@IBS2     ;SET LON ON KGM.
1145 007706  013777  001124  171450      MOV     $GDDAT,@IBD     ;SEND PATTERN.
1146 007714  117737  171464  001126      MOVB    @IBD2,$BDDAT    ;READ ATA FROM KGM.
1147 007722  123737  001124  001126      CMPB    $GDDAT,$BDDAT   ;DATA SENT = DATA RECEIVED?
1148 007730  001402                      BEQ     2$              ;YES, CONTINUE
1149

                                ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

(1)
(1) 007732  104004              ERROR   4               ;/MODULE FAULT DETECTED:
1150                                                    ;ERROR - BAD DATA PASSED BETWEEN
1151                                                    ;MODULE UNDER TEST AND KGM.

                                ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
1153 007734  000407              BR      TST56          ;;
1154
1155 007736  105237  001124      2$:     INCB    $GDDAT          ;CHANGE PATTERN.
1156 007742  001347              BNE     1$              ;IF NOT DONE, CONTINUE.
1157
1158 007744  005077  171412              CLR     @IBS            ;CLEAR CSR'S
1159 007750  005077  171426              CLR     @IBS2
1160
```

```
 1162                                   ;;*****************************************************************
  (3)                                   ;*TEST 56       *TEMP END OF TESTS
  (3)                                   ;;*****************************************************************
  (2)  007754  000004                   TST56:  SCOPE
 1163  007756                          EOP:
 1164                                           .SBTTL  SYSMAC ROUTINES:
 1165                                   .SBTTL  END OF PASS ROUTINE
  (1)
  (2)                                   ;;*****************************************************************
  (1)                                   ;*INCREMENT THE PASS NUMBER ($PASS)
  (1)                                   ;*TYPE 'END PASS #XXXXX'' (WHERE XXXXX IS A DECIMAL NUMBER)
  (1)                                   ;*IF THERES A MONITOR GO TO IT
  (1)                                   ;*IF THERE ISN'T JUMP TO RSTART
  (1)
  (1)  007756                           $EOP:
  (1)  007756  000004                           SCOPE
  (1)  007760  005037  001102                   CLR     $TSTNM          ;;ZERO THE TEST NUMBER
  (1)  007764  005037  001160                   CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
  (1)  007770  005237  001176                   INC     $PASS           ;;INCREMENT THE PASS NUMBER
  (1)  007774  042737  100000  001176           BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
  (1)  010002  005327                           DEC     (PC)+           ;;LOOP?
  (1)  010004  000001                   $EOPCT: .WORD   1
  (1)  010006  003022                           BGT     $DOAGN          ;;YES
  (1)  010010  012737                           MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
  (1)  010012  000001                   $ENDCT: .WORD   1
  (1)  010014  010004                           $EOPCT
  (1)  010016  104401  010063                   TYPE    ,$ENDMG         ;;TYPE 'END PASS #''
  (2)  010022  013746  001176                   MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
  (2)  010026  104405                           TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
  (1)  010030  104401  010060                   TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
  (1)  010034  013700  000042           $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
  (1)  010040  001405                           BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
  (1)  010042  000005                           RESET                   ;;CLEAR THE WORLD
  (1)  010044  004710                   $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
  (1)  010046  000240                           NOP                     ;;SAVE ROOM
  (1)  010050  000240                           NOP                     ;;FOR
  (1)  010052  000240                           NOP                     ;;ACT11
  (1)  010054                           $DOAGN:
  (1)  010054  000137                           JMP     @(PC)+          ;;RETURN
  (1)  010056  002354                   $RTNAD: .WORD   RSTART
  (1)  010060     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
  (1)  010063     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
  (1)  010070  050040  051501  020123
  (1)  010076  000043
```

```
1167
 (1)
 (1)                                        ;/DELMA
 (1)
 (1)                                ;/ROUTINE TO PROVIDE DELAYS IN INCREMENTS OF 25 US
 (1)                                ;/
 (1)                                ;/ CALL=        JSR     R5,DEL50
 (1)                                ;/              .WORD   X       (# OF 25 US TO DELAY)
 (1)                                ;/              ;RETURNS HERE
 (1)                                ;/
 (1)
 (1)  010100  012500          DEL50:  MOV     (R5)+,R0                ;/GET # OF 25 US DELAYS
 (1)  010102  013701  001436  1$:     MOV     CPUDLY,R1       ;/# FOR LOOP TO DO 25 US.
 (1)  010106  005301          2$:     DEC     R1              ;/DEC IT
 (1)  010110  001376                  BNE     2$              ;/WAIT FOR 25 US TIME?
 (1)  010112  005300                  DEC     R0              ;/DONE # OF 25 US DELAY DESIRED?
 (1)  010114  001372                  BNE     1$              ;/NO - NEXT ONE.
 (1)  010116  000205                  RTS     R5              ;/YES - EXIT.
 (1)
1168                            .SBTTL  ERROR HANDLER ROUTINE
 (1)
 (2)                            ;;**********************************************************************
 (1)                            ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 (1)                            ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 (1)                            ;*AND GO TO $ERRTYP ON ERROR
 (1)                            ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 (1)                            ;*SW15=1        HALT ON ERROR
 (1)                            ;*SW13=1        INHIBIT ERROR TYPEOUTS
 (1)                            ;*SW09=1        LOOP ON ERROR
 (1)                            ;*CALL
 (1)                            ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 (1)
 (1)  010120                    $ERROR:
 (1)  010120  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 (1)  010122  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 (1)  010126  001775                  BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 (1)  010130  013777  001102  171004          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 (1)  010136  005237  001112                  INC     $ERTTL          ;;INC THE ERROR COUNT
 (1)  010142  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 (1)  010146  162737  000002  001116          SUB     #2,$ERRPC
 (1)  010154  117737  170736  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 (1)  010162  032777  020000  170750          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 (1)  010170  001004                          BNE     20$             ;;SKIP TYPEOUTS
 (1)  010172  004737  010272                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
 (1)  010176  104401  001165                  TYPE    ,$CRLF
 (1)  010202                          20$:
 (1)  010202  122737  000001  CC 210          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 (1)  010210  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
 (1)  010212  113737  001114  010224          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
 (1)  010220  004737  012520                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
 (1)  010224     000                  21$:    .BYTE   0
 (1)  010225     000                          .BYTE   0
 (1)  010226  000777                  22$:    BR      22$             ;;APT ERROR LOOP
 (1)  010230  005777  170704          2$:     TST     @SWR            ;;HALT ON ERROR
 (1)  010234  100002                          BPL     3$              ;;SKIP IF CONTINUE
 (1)  010236  000000                          HALT                    ;;HALT ON ERROR!
 (1)  010240  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 (1)  010242  032777  001000  170670  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
```

```
(1)  010250  001402                    BEQ     4$                  ;;BR IF NO
(1)  010252  013716  001110            MOV     $LPERR,(SP)         ;;FUDGE RETURN FOR LOOPING
(1)  010256  005737  001162     4$:    TST     $ESCAPE             ;;CHECK FOR AN ESCAPE ADDRESS
(1)  010262  001402                    BEQ     5$                  ;;BR IF NONE
(1)  010264  013716  001162            MOV     $ESCAPE,(SP)        ;;FUDGE RETURN ADDRESS FOR ESCAPE
(1)  010270                     5$:
(1)  010270  000002                    RTI                         ;;RETURN
1169                            .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
(1)
(2)                             ;;**************************************************************************
(1)                             ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
(1)                             ;*ERROR IS TO BE REPOPTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
(1)                             ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
(1)
(1)  010272                     $ERRTYP:
(1)  010272  104401  001165            TYPE    ,$CRLF              ;;"CARRIAGE RETURN" & "LINE FEED"
(1)  010276  010046                    MOV     R0,-(SP)            ;;SAVE R0
(1)  010300  005000                    CLR     R0                  ;;PICKUP THE ITEM INDEX
(1)  010302  153700  001114            BISB    @#$ITEMB,R0
(1)  010306  001004                    BNE     1$                  ;;IF ITEM NUMBER IS ZERO, JUST
(1)                                                                ;;TYPE THE PC OF THE ERROR
(2)  010310  013746  001116            MOV     $ERRPC,-(SP)        ;;SAVE $ERRPC FOR TYPEOUT
(2)                                                                ;;ERROR ADDRESS
(2)  010314  104402                    TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
(1)  010316  000426                    BR      6$                  ;;GET OUT
(1)  010320  005300             1$:    DEC     R0                  ;;ADJUST THE INDEX SO THAT IT WILL
(1)  010322  006300                    ASL     R0                  ;;      WORK FOR THE ERROR TABLE
(1)  010324  006300                    ASL     R0
(1)  010326  006300                    ASL     R0
(1)  010330  062700  001252            ADD     #$ERRTB,R0          ;;FORM TABLE POINTER
(1)  010334  012037  010344            MOV     (R0)+,2$            ;;PICKUP "ERROR MESSAGE" POINTER
(1)  010340  001404                    BEQ     3$                  ;;SKIP TYPEOUT IF NO POINTER
(1)  010342  104401                    TYPE                        ;;TYPE THE "ERROR MESSAGE"
(1)  010344  000000             2$:    .WORD   0                   ;;"ERROR MESSAGE" POINTER GOES HERE
(1)  010346  104401  001165            TYPE    ,$CRLF              ;;"CARRIAGE RETURN" & "LINE FEED"
(1)  010352  012037  010362     3$:    MOV     (R0)+,4$            ;;PICKUP "DATA HEADER" POINTER
(1)  010356  001404                    BEQ     5$                  ;;SKIP TYPEOUT IF 0
(1)  010360  104401                    TYPE                        ;;TYPE THE "DATA HEADER"
(1)  010362  000000             4$:    .WORD   0                   ;;"DATA HEADER" POINTER GOES HERE
(1)  010364  104401  001165            TYPE    ,$CRLF              ;;"CARRIAGE RETURN" & "LINE FEED"
(1)  010370  011000             5$:    MOV     (R0),R0             ;;PICKUP "DATA TABLE" POINTER
(1)  010372  001004                    BNE     7$                  ;;GO TYPE THE DATA
(1)  010374  012600             6$:    MOV     (SP)+,R0            ;;RESTORE R0
(1)  010376  104401  001165            TYPE    ,$CRLF              ;;"CARRIAGE RETURN" & "LINE FEED"
(1)  010402  000207                    RTS     PC                  ;;RETURN
(1)  010404                     7$:
(2)  010404  013046                    MOV     @(R0)+,-(SP)        ;;SAVE @(R0)+ FOR TYPEOUT
(2)  010406  104402                    TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
(1)  010410  005710                    TST     (R0)                ;;IS THERE ANOTHER NUMBER?
(1)  010412  001770                    BEQ     6$                  ;;BR IF NO
(1)  010414  104401  010422            TYPE    ,8$                 ;;TYPE TWO(2) SPACES
(1)  010420  000771                    BR      7$                  ;;LOOP
(1)  010422  020040      000   8$:     .ASCIZ  /  /                ;;TWO(2) SPACES
(1)          010426                    .EVEN
1170                            .SBTTL  SCOPE HANDLER ROUTINE
(1)
```

```
    (2)                                 ;;*****************************************************************
    (1)                                 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
    (1)                                 ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
    (1)                                 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
    (1)                                 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
    (1)                                 ;*SW14=1        LOOP ON TEST
    (1)                                 ;*SW11=1        INHIBIT ITERATIONS
    (1)                                 ;*SW09=1        LOOP ON ERROR
    (1)                                 ;*SW08=1        LOOP ON TEST IN SWR<7:0>
    (1)                                 ;*CALL
    (1)                                 ;*      SCOPE           ;;SCOPE=IOT
    (1)
    (1)  010426                         $SCOPE:
    (1)  010426  104407                         CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
    (2)  010430  104407                         CKSWR
    (1)  010432  032777  040000  170500  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
    (1)  010440  001114                         BNE     $OVER           ;;YES IF SW14=1
    (1)                                 ;#####START OF CODE FOR THE XOR TESTER#####
    (1)  010442  000416                 $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
    (1)                                                                 ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
    (1)  010444  013746  000004                 MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
    (1)  010450  012737  010470  000004         MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
    (1)  010456  005737  177060                 TST     @#177060        ;;TIME OUT ON XOR?
    (1)  010462  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
    (1)  010466  000463                         BR      $SVLAD          ;;GO TO THE NEXT TEST
    (1)  010470  022626                 5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
    (1)  010472  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
    (1)  010476  000423                         BR      7$              ;;LOOP ON THE PRESENT TEST
    (1)  010500                         6$:;#####END OF CODE FOR THE XOR TESTER#####
    (1)  010500  032777  000400  170432         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
    (1)  010506  001404                         BEQ     2$              ;;BR IF NO
    (1)  010510  127737  170424  001102         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
    (1)  010516  001465                         BEQ     $OVER           ;;BR IF YES
    (1)  010520  105737  001103         2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
    (1)  010524  001421                         BEQ     3$              ;;BR IF NO
    (1)  010526  123737  001115  001103         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
    (1)  010534  101015                         BHI     3$              ;;BR IF NO
    (1)  010536  032777  001000  170374         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
    (1)  010544  001404                         BEQ     4$              ;;BR IF NO
    (1)  010546  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
    (1)  010554  000446                         BR      $OVER
    (1)  010556  105037  001103         4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
    (1)  010562  005037  001160                 CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
    (1)  010566  000415                         BR      1$              ;;ESCAPE TO THE NEXT TEST
    (1)  010570  032777  004000  170342  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
    (1)  010576  001011                         BNE     1$              ;;BR IF YES
    (1)  010600  005737  001176                 TST     $PASS           ;;IF FIRST PASS OF PROGRAM
    (1)  010604  001406                         BEQ     1$              ;;      INHIBIT ITERATIONS
    (1)  010606  005237  001104                 INC     $ICNT           ;;INCREMENT ITERATION COUNT
    (1)  010612  023737  001160  001104         CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
    (1)  010620  002024                         BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
    (1)  010622  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
    (1)  010630  013737  010706  001160         MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
    (1)  010636  105237  001102         $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
    (1)  010642  113737  001102  001174         MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
    (1)  010650  011637  001106                 MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
```

```
(1)  010654  011637  001110                 MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
(1)  010660  005037  001162                 CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
(1)  010664  112737  000001  001115         MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1)  010672  013777  001102  170242 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
(1)  010700  013716  001106                 MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
(1)  010704  000002                         RTI                     ;;FIXES PS
(1)  010706  003720                 $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
1171                                .SBTTL  TTY INPUT ROUTINE
(1)
(2)                                 ;;**********************************************************************
(1)                                 .ENABL  LSB
(1)
(2)                                 ;;**********************************************************************
(1)                                 ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
(1)                                 ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
(1)                                 ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
(1)                                 ;*WHEN OPERATING IN TTY FLAG MODE.
(1)  010710  022737  000176  001140 $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
(1)  010716  001074                         BNE     15$             ;;BRANCH IF NO
(1)  010720  105777  170220                 TSTB    @$TKS           ;;CHAR THERE?
(1)  010724  100071                         BPL     15$             ;;IF NO, DON'T WAIT AROUND
(1)  010726  117746  170214                 MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
(1)  010732  042716  177600                 BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
(1)  010736  022726  000007                 CMP     #7,(SP)+        ;;IS IT A CONTROL G?
(1)  010742  001062                         BNE     15$             ;;NO, RETURN TO USER
(1)  010744  123727  001134  000001         CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
(1)  010752  001456                         BEQ     15$             ;;BRANCH IF YES
(1)
(1)  010754  104401  011445                 TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
(1)  010760  104401  011452         $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
(2)  010764  013746  000176                 MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
(2)  010770  104402                         TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
(1)  010772  104401  011463                 TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
(1)  010776  005046                 19$:    CLR     -(SP)           ;;CLEAR COUNTER
(1)  011000  005046                         CLR     -(SP)           ;;THE NEW SWR
(1)  011002  105777  170136         7$:     TSTB    @$TKS           ;;CHAR THERE?
(1)  011006  100375                         BPL     7$              ;;IF NOT TRY AGAIN
(1)
(1)  011010  117746  170132                 MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
(1)  011014  042716  177600                 BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
(1)
(1)
(1)
(1)  011020  021627  000025         9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
(1)  011024  001005                         BNE     10$             ;;BRANCH IF NOT
(1)  011026  104401  011440                 TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
(1)  011032  062706  000006         20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
(1)  011036  000757                         BR      19$             ;;LET'S TRY IT AGAIN
(1)
(1)
(1)  011040  021627  000015         10$:    CMP     (SP),#15        ;;IS IT A <CR>?
(1)  011044  001022                         BNE     16$             ;;BRANCH IF NO
(1)  011046  005766  000004                 TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
(1)  011052  001403                         BEQ     11$             ;;BRANCH IF YES
(1)  011054  016677  000002  170056         MOV     2(SP),@SWR      ;;SAVE NEW SWR
(1)  011062  062706  000006         11$:    ADD     #6,SP           ;;CLEAR UP STACK
```

```
(1)  011066  104401  001165            14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
(1)  011072  123727  001135  000001             CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
(1)  011100  001003                             BNE     15$             ;;BRANCH IF NOT
(1)  011102  012777  000100  170034             MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
(1)  011110  000002                     15$:    RTI                     ;;RETURN
(1)  011112  004737  012360             16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
(1)  011116  021627  000060                     CMP     (SP),#60        ;;CHAR < 0?
(1)  011122  002420                             BLT     18$             ;;BRANCH IF YES
(1)  011124  021627  000067                     CMP     (SP),#67        ;;CHAR > 7?
(1)  011130  003015                             BGT     18$             ;;BRANCH IF YES
(1)  011132  042726  000060                     BIC     #60,(SP)+       ;;STRIP-OFF ASCII
(1)  011136  005766  000002                     TST     2(SP)           ;;IS THIS THE FIRST CHAR
(1)  011142  001403                             BEQ     17$             ;;BRANCH IF YES
(1)  011144  006316                             ASL     (SP)            ;;NO, SHIFT PRESENT
(1)  011146  006316                             ASL     (SP)            ;;   CHAR OVER TO MAKE
(1)  011150  006316                             ASL     (SP)            ;;   ROOM FOR NEW ONE.
(1)  011152  005266  000002             17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
(1)  011156  056616  177776                     BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
(1)  011162  000707                             BR      7$              ;;GET THE NEXT ONE
(1)  011164  104401  001164             18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
(1)  011170  000720                             BR      20$             ;;SIMULATE CONTROL-U
(1)                                     .DSABL  LSB
(1)
(1)
(2)                                     ;;**************************************************************************
(1)                                     ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                     ;*CALL:
(1)                                     ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                     ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
(1)                                     ;*                              ;;WITH PARITY BIT STRIPPED OFF
(1)                                     ;
(1)
(1)  011172  011646             $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
(1)  011174  016666  000004  000002     MOV     4(SP),2(SP)     ;;SAVE THE PS
(1)  011202  105777  167736             1$:     TSTB    @$TKS           ;;WAIT FOR
(1)  011206  100375                             BPL     1$              ;;A CHARACTER
(1)  011210  117766  167732  000004             MOVB    @$TKB,4(SP)     ;;READ THE TTY
(1)  011216  042766  177600  000004             BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
(1)  011224  026627  000004  000023             CMP     4(SP),#23       ;;IS IT A CONTROL-S?
(1)  011232  001013                             BNE     3$              ;;BRANCH IF NO
(1)  011234  105777  167704             2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
(1)  011240  100375                             BPL     2$              ;;LOOP UNTIL ITS THERE
(1)  011242  117746  167700                     MOVB    @$TKB,-(SP)     ;;GET CHARACTER
(1)  011246  042716  177600                     BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
(1)  011252  022627  000021                     CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
(1)  011256  001366                             BNE     2$              ;;IF NOT DISCARD IT
(1)  011260  000750                             BR      1$              ;;YES, RESUME
(1)  011262  026627  000004  000021     3$:     CMP     4(SP),#$XON     ;;IS IT A RANDOM XON?                   ;RAN001
(1)  011270  001744                             BEQ     1$              ;;BRANCH IF YES                         ;RAN001
(1)  011272  026627  000004  000140             CMP     4(SP),#140      ;;IS IT UPPER CASE?
(1)  011300  002407                             BLT     4$              ;;BRANCH IF YES
(1)  011302  026627  000004  000175             CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
(1)  011310  003003                             BGT     4$              ;;BRANCH IF YES
(1)  011312  042766  000040  000004             BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
(1)  011320  000002                     4$:     RTI                     ;;GO BACK TO USER
(2)                                     ;;**************************************************************************
```

```
(1)                                     ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1)                                     ;*CALL:
(1)                                     ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
(1)                                     ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1)                                     ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
(1)
(1)  011322  010346                     $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
(1)  011324  012703  011430             1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
(1)  011330  022703  011440             2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
(1)  011334  101405                             BLOS    4$              ;;BR IF YES
(1)  011336  104410                             RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
(1)  011340  112613                             MOVB    (SP)+,(R3)      ;;GET CHARACTER
(1)  011342  122713  000177             10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
(1)  011346  001003                             BNE     3$              ;;SKIP IF NOT
(1)  011350  104401  001164             4$:     TYPE    ,$QUES          ;;TYPE A '?'
(1)  011354  000763                             BR      1$              ;;CLEAR THE BUFFER AND LOOP
(1)  011356  111337  011426             3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
(1)  011362  104401  011426                     TYPE    ,9$
(1)  011366  122723  000015                     CMPB    #15,(R3)+       ;;CHECK FOR RETURN
(1)  011372  001356                             BNE     2$              ;;LOOP IF NOT RETURN
(1)  011374  105063  177777                     CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
(1)  011400  104401  001166                     TYPE    ,$LF            ;;TYPE A LINE FEED
(1)  011404  012603                             MOV     (SP)+,R3        ;;RESTORE R3
(1)  011406  011646                             MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
(1)  011410  016666  000004  000002             MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
(1)  011416  012766  011430  000004             MOV     #$TTYIN,4(SP)
(1)  011424  000002                             RTI                     ;;RETURN
(1)  011426     000             9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
(1)  011427     000                     .BYTE   0                       ;;TERMINATOR
(1)  011430  000010             $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
(1)  011440  052536  005015      000    $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
(1)  011445     136  006507  000012     $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
(1)  011452  005015  053523  020122     $MSWR:  .ASCIZ  <15><12>/SWR = /
(1)  011460  020075     000
(1)  011463     040  047040  053505     $MNEW:  .ASCIZ  /  NEW = /
(1)  011470  036440  000040
```

```
1173                                    .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 (1)                                    ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 (1)                                    ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 (1)                                    ;*CALL:
 (1)                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 (1)                                    ;*      TYPOS                   ;;CALL FOR TYPEOUT
 (1)                                    ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 (1)                                    ;*      .BYTE   M               ;;M=1 OR 0
 (1)                                    ;*                                      ;;1=TYPE LEADING ZEROS
 (1)                                    ;*                                      ;;0=SUPPRESS LEADING ZEROS
 (1)                                    ;*
 (1)                                    ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 (1)                                    ;*$TYPOS OR $TYPOC
 (1)                                    ;*CALL:
 (1)                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 (1)                                    ;*      TYPON                   ;;CALL FOR TYPEOUT
 (1)                                    ;*
 (1)                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 (1)                                    ;*CALL:
 (1)                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 (1)                                    ;*      TYPOC                   ;;CALL FOR TYPEOUT
 (1)
 (1)  011474  017646  000000            $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 (1)  011500  116637  000001  011717            MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 (1)  011506  112637  011721                    MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 (1)  011512  062716  000002                    ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 (1)  011516  000406                            BR      $TYPON
 (1)  011520  112737  000001  011717    $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 (1)  011526  112737  000006  011721            MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 (1)  011534  112737  000005  011716    $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 (1)  011542  010346                            MOV     R3,-(SP)        ;;SAVE R3
 (1)  011544  010446                            MOV     R4,-(SP)        ;;SAVE R4
 (1)  011546  010546                            MOV     R5,-(SP)        ;;SAVE R5
 (1)  011550  113704  011721                    MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 (1)  011554  005404                            NEG     R4
 (1)  011556  062704  000006                    ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 (1)  011562  110437  011720                    MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 (1)  011566  113704  011717                    MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 (1)  011572  016605  000012                    MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 (1)  011576  005003                            CLR     R3              ;;CLEAR THE OUTPUT WORD
 (1)  011600  006105                    1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 (1)  011602  000404                            BR      3$              ;;GO DO MSB
 (1)  011604  006105                    2$:     ROL     R5              ;;FORM THIS DIGIT
 (1)  011606  006105                            ROL     R5
 (1)  011610  006105                            ROL     R5
 (1)  011612  010503                            MOV     R5,R3
 (1)  011614  006103                    3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 (1)  011616  105337  011720                    DECB    $OMODE          ;;TYPE THIS DIGIT?
 (1)  011622  100016                            BPL     7$              ;;BR IF NO
 (1)  011624  042703  177770                    BIC     #177770,R3      ;;GET RID OF JUNK
 (1)  011630  001002                            BNE     4$              ;;TEST FOR 0
 (1)  011632  005704                            TST     R4              ;;SUPPRESS THIS 0?
 (1)  011634  001403                            BEQ     5$              ;;BR IF YES
```

```
(1)  011636  005204                    4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
(1)  011640  052703  000060                    BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
(1)  011644  052703  000040            5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
(1)  011650  110337  011714                    MOVB    R3,8$           ;;SAVE FOR TYPING
(1)  011654  104401  011714                    TYPE    ,8$             ;;GO TYPE THIS DIGIT
(1)  011660  105337  011716            7$:     DECB    $OCNT           ;;COUNT BY 1
(1)  011664  003347                            BGT     2$              ;;BR IF MORE TO DO
(1)  011666  002402                            BLT     6$              ;;BR IF DONE
(1)  011670  005204                            INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
(1)  011672  000744                            BR      2$              ;;GO DO THE LAST D'GIT
(1)  011674  012605                    6$:     MOV     (SP)+,R5        ;;RESTORE R5
(1)  011676  012604                            MOV     (SP)+,R4        ;;RESTORE R4
(1)  011700  012603                            MOV     (SP)+,R3        ;;RESTORE R3
(1)  011702  016666  000002  000004            MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
(1)  011710  012616                            MOV     (SP)+,(SP)
(1)  011712  000002                            RTI                     ;;RETURN
(1)  011714     000                    8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
(1)  011715     000                            .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
(1)  011716     000                    $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
(1)  011717     000                    $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
(1)  011720  000000                    $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
 1175                                   .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
  (1)
  (2)                                   ;;*****************************************************************
  (1)                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
  (1)                                   ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
  (1)                                   ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
  (1)                                   ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
  (1)                                   ;*REPLACED WITH SPACES.
  (1)                                   ;*CALL:
  (1)                                   ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
  (1)                                   ;*      TYPDS                   ;;GO TO THE ROUTINE
  (1)
  (1)  011722                           $TYPDS:
  (3)  011722  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
  (3)  011724  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
  (3)  011726  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
  (3)  011730  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
  (3)  011732  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
  (1)  011734  012746  020200                   MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
  (1)  011740  016605  000020                   MOV     20(SP),R5       ;;GET THE INPUT NUMBER
  (1)  011744  100004                           BPL     1$              ;;BR IF INPUT IS POS.
  (1)  011746  005405                           NEG     R5              ;;MAKE THE BINARY NUMBER POS.
  (1)  011750  112766  000055  000001           MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
  (1)  011756  005000                   1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
  (1)  011760  012703  012136                   MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
  (1)  011764  112723  000040                   MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
  (1)  011770  005002                   2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
  (1)  011772  016001  012126                   MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
  (1)  011776  160105                   3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
  (1)  012000  002402                           BLT     4$              ;;BR IF DONE
  (1)  012002  005202                           INC     R2              ;;INCREASE THE BCD DIGIT BY 1
  (1)  012004  000774                           BR      3$
  (1)  012006  060105                   4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
  (1)  012010  005702                           TST     R2              ;;CHECK IF BCD DIGIT=0
  (1)  012012  001002                           BNE     5$              ;;FALL THROUGH IF 0
  (1)  012014  105716                           TSTB    (SP)            ;;STILL DOING LEADING 0'S?
  (1)  012016  100407                           BMI     7$              ;;BR IF YES
  (1)  012020  106316                   5$:     ASLB    (SP)            ;;MSD?
  (1)  012022  103003                           BCC     6$              ;;BR IF NO
  (1)  012024  116663  000001  177777           MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
  (1)  012032  052702  000060           6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
  (1)  012036  052702  000040           7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
  (1)  012042  110223                           MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
  (1)  012044  005720                           TST     (R0)+           ;;JUST INCREMENTING
  (1)  012046  020027  000010                   CMP     R0,#10          ;;CHECK THE TABLE INDEX
  (1)  012052  002746                           BLT     2$              ;;GO DO THE NEXT DIGIT
  (1)  012054  003002                           BGT     8$              ;;GO TO EXIT
  (1)  012056  010502                           MOV     R5,R2           ;;GET THE LSD
  (1)  012060  000764                           BR      6$              ;;GO CHANGE TO ASCII
  (1)  012062  105726                   8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
  (1)  012064  100003                           BPL     9$              ;;BR IF NO
  (1)  012066  116663  177777  177776           MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
  (1)  012074  105013                   9$:     CLRB    (R3)            ;;SET THE TERMINATOR
  (3)  012076  012605                           MOV     (SP)+,R5        ;;POP STACK INTO R5
  (3)  012100  012603                           MOV     (SP)+,R3        ;;POP STACK INTO R3
  (3)  012102  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
(3)  012104  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  012106  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  012110  104401  012136                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
(1)  012114  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
(1)  012122  012616                          MOV     (SP)+,(SP)
(1)  012124  000002                          RTI                     ;;RETURN TO USER
(1)  012126  023420                  $DTBL:  10000.
(1)  012130  001750                          1000.
(1)  012132  000144                          100.
(1)  012134  000012                          10.
(1)  012136  000004                  $DBLK:  .BLKW   4
1176                                 .SBTTL  TYPE ROUTINE
(1)
(2)                                  ;;*****************************************************************
(1)                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1)                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1)                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1)                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1)                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1)                                  ;*
(1)                                  ;*CALL:
(1)                                  ;*1) USING A TRAP INSTRUCTION
(1)                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1)                                  ;*OR
(1)                                  ;*      TYPE
(1)                                  ;*      MESADR
(1)                                  ;*
(1)
(1)                                  ;*
(1)
(1)  012146  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
(1)  012152  100002                          BPL     1$              ;;BR IF YES
(1)  012154  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
(1)  012156  000430                          BR      3$              ;;LEAVE
(1)  012160  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
(1)  012162  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
(1)  012166  122737  000001  001210          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
(1)  012174  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
(1)  012176  132737  000100  001211          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
(1)  012204  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
(1)  012206  010037  012216                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
(1)  012212  004737  012510                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
(1)  012216  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
(1)  012220  132737  000040  00121?  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
(1)  012226  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
(1)  012230  112046                  ?$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
(1)  012232  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
(1)  012234  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
(1)  012236  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
(1)  012240  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
(1)  012244  000002                          RTI                     ;;RETURN
(1)  012246  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
(1)  012252  001430                          BEQ     8$
(1)  012254  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
(1)  012260  001006                          BNE     5$
(1)  012262  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
(1)  012264  104401                          TYPE                    ;;TYPE A CR AND LF
(1)  012266  001165                          $CRLF
```

```
(1) 012270 105037 012476          CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
(1) 012274 000755                 BR      2$              ;;GET NEXT CHARACTER
(1) 012276 004737 012360    5$:   JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
(1) 012302 123726 001156    6$:   CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
(1) 012306 001350                 BNE     2$              ;;IF NO GO GET NEXT CHAR.
(1) 012310 013746 001154          MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
(1)                                                       ;;AND THE NULL CHAR.
(1) 012314 105366 000001    7$:   DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
(1) 012320 002770                 BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
(1) 012322 004737 012360          JSR     PC,$TYPEC       ;;GO TYPE A NULL
(1) 012326 105337 012476          DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
(1) 012332 000770                 BR      7$              ;;LOOP
(1)
(1)                               ;HORIZONTAL TAB PROCESSOR
(1)
(1) 012334 112716 000040    8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
(1) 012340 004737 012360    9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
(1) 012344 132737 000007 012476   BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
(1) 012352 001372                 BNE     9$              ;;TAB STOP
(1) 012354 005726                 TST     (SP)+           ;;POP SPACE OFF STACK
(1) 012356 000724                 BR      2$              ;;GET NEXT CHARACTER
(1) 012360                  $TYPEC:
(1) 012360 105777 166560          TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?          ;MJD001
(1) 012364 100022                 BPL     10$             ;;BR IF NOT                     ;MJD001
(1) 012366 017746 166554          MOV     @$TKB,-(SP)     ;;GET CHAR                      ;MJD001
(1) 012372 042716 177600          BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS         ;MJD001
(1) 012376 122716 000023          CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF                 ;MJD001
(1) 012402 001012                 BNE     102$            ;;BR IF NOT                     ;MJD001
(1) 012404                  101$:                                                         ;MJD001
(1) 012404 105777 166534          TSTB    @$TKS           ;;WAIT FOR CHAR                 ;MJD001
(1) 012410 100375                 BPL     101$                                            ;MJD001
(1) 012412 117716 166530          MOVB    @$TKB,(SP)      ;;GET CHAR                      ;MJD001
(1) 012416 042716 177600          BIC     #177600,(SP)    ;;STRIP IT                      ;MJD001
(1) 012422 122716 000021          CMPB    #$XON,(SP)      ;;WAS IT XON?                   ;MJD001
(1) 012426 001366                 BNE     101$            ;;BR IF NOT                     ;MJD001
(1) 012430                  102$:                                                         ;MJD001
(1) 012430 005726                 TST     (SP)+           ;;FIX STACK                     ;MJD001
(1) 012432                  10$:                                                          ;MJD001
(1) 012432 105777 166512          TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
(1) 012436 100375                 BPL     10$                                             ;MJD001
(1) 012440 116677 000002 166504   MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1) 012446 122766 000015 000002   CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
(1) 012454 001003                 BNE     1$              ;;BRANCH IF NO
(1) 012456 105037 012476          CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
(1) 012462 000406                 BR      $TYPEX          ;;EXIT
(1) 012464 122766 000012 000002   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
(1) 012472 001402                 BEQ     $TYPEX          ;;BRANCH IF YES
(1) 012474 105227                 INCB    (PC)+           ;;COUNT THE CHARACTER
(1) 012476 000000           $CHARCNT:.WORD  0             ;;CHARACTER COUNT STORAGE
(1) 012500 000207           $TYPEX: RTS     PC
(1)
1177                              .SBTTL  APT COMMUNICATIONS ROUTINE
(1)
(2)                               ;;**********************************************************************
(1) 012502 112737 000001 012746 $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
(1) 012510 112737 000001 012744 $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
```

```
(1)  012516  000403                          BR      $ATYC
(1)  012520  112737  000001  012746  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
(1)  012526                        $ATYC:
(3)  012526  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  012530  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(1)  012532  105737  012744                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
(1)  012536  001450                          BEQ     5$              ;;IF NOT:  BR
(1)  012540  122737  000001  001210          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
(1)  012546  001031                          BNE     3$              ;;IF NOT:  BR
(1)  012550  132737  000100  001211          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
(1)  012556  001425                          BEQ     3$              ;;IF NOT:  BR
(1)  012560  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
(1)  012564  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
(1)  012572  005737  001170          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
(1)  012576  001375                          BNE     1$              ;;IF NOT:  WAIT
(1)  012600  010037  001204                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
(1)  012604  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
(1)  012606  001376                          BNE     2$
(1)  012610  163700  001204                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
(1)  012614  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
(1)  012616  010037  001206                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
(1)  012622  012737  000004  001170          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
(1)  012630  000413                          BR      5$
(1)  012632  017637  000004  012656  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
(1)  012640  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
(3)    2646  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
(1)  012652  004737  012146                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
(1)  012656  000000                  4$:     .WORD   0
(1)  012660                          5$:
(1)  012660  105737  012746          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
(1)  012664  001416                          BEQ     12$             ;;IF NOT:  BR
(1)  012666  005737  001210                  TST     $ENV            ;;RUNNING UNDER APT?
(1)  012672  001413                          BEQ     12$             ;;IF NOT:  BR
(1)  012674  005737  001170          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
(1)  012700  001375                          BNE     11$             ;;IF NOT:  WAIT
(1)  012702  017637  000004  001172          MOV     @4(SP),$FATAL   ;;GET ERROR #
(1)  012710  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
(1)  012716  005237  001170                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
(1)  012722  105037  012746          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
(1)  012726  105037  012745                  CLRB    $LFLG           ;;CLEAR LOG FLAG
(1)  012732  105037  012744                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
(3)  012736  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  012740  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  012742  000207                          RTS     PC              ;;RETURN
(1)  012744     000  $MFLG:  .BYTE   0       ;;MESSG. FLAG
(1)  012745     000  $LFLG:  .BYTE   0       ;;LOG FLAG
(1)  012746     000  $FFLG:  .BYTE   0       ;;FATAL FLAG
(1)          012750          .EVEN
(1)          000200  APTSIZE=200
(1)          000001  APTENV=001
(1)          000100  APTSPOOL=100
(1)          000040  APTCSUP=040
```

```
1179                                    .SBTTL  POWER DOWN AND UP ROUTINES
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    ;POWER DOWN ROUTINE
 (1)  012750  012737  013110  000024    $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
 (1)  012756  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;;PRIO:7
 (3)  012764  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 (3)  012766  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 (3)  012770  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 (3)  012772  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 (3)  012774  010446                            MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 (3)  012776  010546                            MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 (3)  013000  017746  166134                    MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
 (1)  013004  010637  013114                    MOV     SP,$SAVR6       ;;SAVE SP
 (1)  013010  012737  013022  000024            MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
 (1)  013016  000000                            HALT
 (1)  013020  000776                            BR      .-2             ;;HANG UP
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    ;POWER UP ROUTINE
 (1)  013022  012737  013110  000024    $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
 (1)  013030  013706  013114                    MOV     $SAVR6,SP       ;;GET SP
 (1)  013034  005037  013114                    CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
 (1)  013040  005237  013114            1$:     INC     $SAVR6          ;;WAIT FOR THE INC
 (1)  013044  001375                            BNE     1$              ;;OF  WORD
 (3)  013046  012677  166066                    MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
 (3)  013052  012605                            MOV     (SP)+,R5        ;;POP STACK INTO R5
 (3)  013054  012604                            MOV     (SP)+,R4        ;;POP STACK INTO R4
 (3)  013056  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
 (3)  013060  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
 (3)  013062  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
 (3)  013064  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
 (1)  013066  012737  012750  000024            MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
 (1)  013074  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;;PRIO:7
 (1)  013102  104401                            TYPE                    ;REPORT THE POWER FAILURE
 (1)  013104  013116            $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
 (1)  013106  000002                    RTI
 (1)  013110  000000            $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
 (1)  013112  000776                    BR      .-2                     ;; BEFORE THE POWER DOWN WAS COMPLETE
 (1)  013114  000000            $SAVR6: 0                               ;;PUT THE SP HERE
 (1)  013116  005015  047520  042527  $POWER: .ASCIZ  <15><12>'POWER'
 (1)  013124  000122
 (1)                                    .EVEN
```

```
1181
1182                                            ;*
1183                                            ;*THIS ROUTINE WILL PROTECT THE PROGRAM
1184                                            ;*FROM INTERRUPTS.
1185                                            ;*
1186                                            ;*THE TRAP CATCHER IS SET UP FOR
1187                                            ;*      .WORD   .+2
1188                                            ;*      .WORD   JSR     PC,R0
1189                                            ;*
1190                                            ;*ILLEGAL INTERRUPTS OR INTERRUPTS TO THE WRONG VECTOR
1191                                            ;*GOTO THE VECTOR AND PICK UP THE ".+2" AS AN ADDRESS
1192                                            ;*AND "4700" AS NEW STATUS.
1193                                            ;*THE .+2 AS A PC WILL CAUSE EXECUTION OF THE "JSR PC,R0" (AN ILLEGAL INSTR).
1194                                            ;*AND TRAP TO LOCATION "4". IN LOCATION 10 WE HAVE A
1195                                            ;*POINTER HERE. IF THIS CONDITION CAUSES A TRAP TO LOC. 4
1196                                            ;*WE WILL REPORT IT IN THE SAME MANNER THAT WE WOULD
1197                                            ;*REPORT ANY OTHER ERROR.
1198
1199                                            ;*IF A BUSS ERROR TRAP DID OCCUR AND CAUSE A TRAP TO 4,
1200                                            ;*WE WILL HALT.
1201
1202  013126  011637  013172          IOTRD:  MOV     (6),TRTO        ;GET WHERE WE CAME TO.
1203  013132  162737  000004  013172          SUB     #4,TRTO         ;FORM REAL ADDR.
1204
1205  013140  023727  013172  001000          CMP     TRTO,#1000      ;DID TRAP COME FROM LESS THAN ADDR. 1000?
1206  013146  003402                         BLE     2$
1207
1208  013150  000000                  1$:     HALT                    ;NO!   MUST BE A BUSS ILLEGAL ADDR. TIME OUT.
1209                                                                  ; ADDRESS CONTAINED IN TRTO.
1210
1211  013152  000776                          BR      1$              ;DON'T ALLOW A CONTINUE.
1212  013154                          2$:
1213
1214  013154  016637  000004  013174          MOV     4(6),TRFRO      ;GET TRAPPED FROM ADDR.
1215
1216  013162  062706  000004                  ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
1217
1218

                                            ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$

 (1)
 (1)  013166  104007                        ERROR   7               ;/MODULE FAULT DETECTED:
1219                                                                ;ERROR! ILLEGAL INTERRUPT
1220                                                                ;OR INTERRUPT TO WRONG
1221                                                                ;VECTOR - IF TEST NUMBER
1222                                                                ;IS LESS THAN 10, ITS LIKELY
1223                                                                ;(BUT NOT EXCLUSIVELY) TO BE A
1224                                                                ;DEVICE OTHER THAN THE IBV-11
1225                                                                ;TO BLAME.
1226                                                                ;IF THE INTERRUPT OCCURRED
1227                                                                ;DURING AN INTERRUPT TEST, I'D
1228                                                                ;SUSPECT A PROBLEM WITH THE
1229                                                                ;IBV-11.
1230                                                                ;IF THE ADDRESS THE INTERRUPT
1231                                                                ;VECTOR TO IS WITHIN THE RANGE
```

```
                                                        ;OF VECTORS ASSIGNED TO THE IBV-11,
                                                        ;THEN I'D SUSPECT THE IBV-11
                                                        ;INTERRUPTED ILLEGALLY.
                                                        ;IF THE ADDRESS THE INTERRUPT
                                                        ;VECTORED TO IS OUTSIDE OF THE
                                                        ;RANGE ASSIGNED TO THE IBV-11,
                                                        ;I'D SUSPECT THAT THE
                                                        ;IBV-11 PUT THE WRONG VECTOR ON
                                                        ;THE BUSS DURING THE INTERRUPT
                                                        ;PROCESS.
                                                        ;FOR THIS ERROR - DON'T
                                                        ;USE "LOOP ON ERROR" OPTION.
                                                        ;ALSO EXPECT THE INTERRUPT TEST TO
                                                        ;REPORT THAT THE IBV-11 DIDN'T
                                                        ;INTERRUPT.
                                                        ;FOLLOW RECOMMENDED PROCEDURE
                                                        ;IN THE DOCUMENT (ON THIS DIAGNOSTIC)
                                                        ;FOR LOOPING ON ERROR

                                        ;;$$$$$$$$$$^^^ ERROR ^^^$$$$$$$$$$
013170  000002                          RTI
013172  000000              TRTO:   .WORD   0             ;ADDR THAT WE INTERRUPTED TO
013174  000000              TRFRO:  .WORD   0             ;ADDR THAT WE INTERRUPTED FROM.
```

```
 1255                                   .SBTTL  TRAP DECODER
  (1)
  (2)                                   ;;**********************************************************************
  (1)                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
  (1)                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
  (1)                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
  (1)                                   ;*GO TO THAT ROUTINE.
  (1)
  (1)  013176  010046                   $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
  (1)  013200  016600  000002                   MOV     2(SP),R0        ;;GET TRAP ADDRESS
  (1)  013204  005740                           TST     -(R0)           ;;BACKUP BY 2
  (1)  013206  111000                           MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
  (1)  013210  006300                           ASL     R0              ;;POSITION FOR INDEXING
  (1)  013212  016000  013232                   MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
  (1)  013216  000200                           RTS     R0              ;;GO TO ROUTINE
  (1)
  (1)
  (1)                                   ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
  (1)
  (1)  013220  011646                   $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
  (1)  013222  016666  000004  000002           MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
  (1)  013230  000002                           RTI                     ;;RESTORE THE PSW
  (1)
  (3)                                   .SBTTL  TRAP TABLE
  (3)
  (3)                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
  (3)                                   ;*BY THE "TRAP" INSTRUCTION.
  (3)
  (3)                                   ;       ROUTINE
  (3)                                   ;       -------
  (3)  013232  013220                   $TRPAD: .WORD   $TRAP2
  (3)  013234  012146                           $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
  (3)  013236  011520                           $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
  (3)  013240  011474                           $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
  (3)  013242  011534                           $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
  (3)  013244  011722                           $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
  (1)
  (3)  013246  010760                           $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
  (1)
  (3)  013250  010710                           $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
  (3)  013252  011172                           $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
  (3)  013254  011322                           $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
```

```
 1257                                                  .SBTTL MESSAGES AND TABLES
 1258
 1259  013256  005007  044415  051502  EM1:    .ASCIZ<7><12><15>#IBS FUNCTION ERROR#
       013264  043040  047125  052103
       013272  047511  020116  051105
       013300  047522  000122
 1260
 1261  013304  005007  044415  042102  EM2:    .ASCIZ<7><12><15>#IBD FUNCTION ERROR#
       013312  043040  047125  052103
       013320  047511  020116  051105
       013326  047522  000122
 1262
 1263  013332  005007  044415  051502  EM3:    .ASCIZ<7><12><15>#IBS DATA ERROR#
       013340  042040  052101  020101
       013346  051105  047522  000122
 1264
 1265  013354  005007  044415  042102  EM4:    .ASCIZ<7><12><15>#IBD DATA ERROR#
       013362  042040  052101  020101
       013370  051105  047522  000122
 1266
 1267  013376  100007  041111  027523  EM5:    .ASCIZ<7><200>#IBS/IBD ADDRESS ERROR#
       013404  041111  020104  042101
       013412  051104  051505  020123
       013420  051105  047522  000122
 1268  013426  100007  041111  041527  EM6:    .ASCIZ  <7><200>#IBWC/IBCA DATA ERROR#
       013434  044457  041502  020101
       013442  040504  040524  042440
       013450  051122  051117     000
 1269
 1270  013455     007  044600  052116  EM7:    .ASCIZ  <7><200>#INTERRUPT ERROR#
       013462  051105  052522  052120
       013470  042440  051122  051117
       013476     000
 1271
 1272  013477     007  046200  043517  EM10:   .ASCII  <7><200>#LOGIC ERROR DETECTED WITH THE "IBC" SIGNAL#
       013504  041511  042440  051122
       013512  051117  042040  052105
       013520  041505  042524  020104
       013526  044527  044124  052040
       013534  042510  021040  041111
       013542  021103  051440  043511
       013550  040516     114
 1273  013553     200  051511  021040          .ASCII  <200>#IS "$CPUOP" OR "SWR12" CORRECT ?#
       013560  041444  052520  050117
       013566  020042  051117  021040
       013574  053523  030522  021062
       013602  041440  051117  042522
       013610  052103  037440
 1274  013614  022200  050103  047525          .ASCII  <200>#$CPUOP DEFAULTS TO PDP-11/03 CPU TYPE#
       013622  020120  042504  040506
       013630  046125  051524  052040
       013636  020117  042120  026520
       013644  030461  030057  020063
       013652  050103  020125  054524
       013660  042520
 1275  013662  051600  051127  031061          .ASCII  <200>#SWR12=0 INDICATES THE MODULE IS NOT A -YA VERSION
```

```
                013670  030075  044440  042116
                013676  041511  052101  051505
                013704  052040  042510  046440
                013712  042117  046125  020105
                013720  051511  047040  052117
                013726  040440  026440  040531
                013734  053040  051105  044523
                013742  047117
  1276          013744  051600  051127  031061          .ASCIZ  <200>#SWR12=1 INDICATES THE MODULE IS A -YA VERSION#
                013752  030475  044440  042116
                013760  041511  052101  051505
                013766  052040  042510  046440
                013774  042117  046125  020105
                014002  051511  040440  026440
                014010  040531  053040  051105
                014016  044523  047117     000
  1277          014023     007  046200  043517  EM11:   .ASCII  <7><200>#LOGIC ERROR DETECTED WITH THE "SRQ" SIGNAL#
                014030  041511  042440  051122
                014036  051117  042040  052105
                014044  041505  042524  020104
                014052  044527  044124  052040
                014060  042510  021040  051123
                014066  021121  051440  043511
                014074  040516     114
  1278          014077     200  051511  021040          .ASCII  <200>#IS "SWR10" ("ER1-INHIBIT <S1-8>") CORRECT ?#
                014104  053523  030522  021060
                014112  024040  042442  030522
                014120  044455  044116  041111
                014126  052111  036040  030523
                014134  034055  021076  020051
                014142  047503  051122  041505
                014150  020124     077
  1279          014153     200  053523  030522          .ASCII  <200>#SWR10=0 INDICATES S1-8 IS CLEARED (OPEN-OFF)#
                014160  036460  020060  047111
                014166  044504  040503  042524
                014174  020123  030523  034055
                014202  044440  020123  046103
                014210  040505  042522  020104
                014216  047450  042520  026516
                014224  043117  024506
  1280          014230  051600  051127  030061          .ASCIZ  <200>#SWR10=1 INDICATES S1-8 IS SET (CLOSED-ON)#
                014236  030475  044440  042116
                014244  041511  052101  051505
                014252  051440  026461  020070
                014260  051511  051440  052105
                014266  024040  046103  051517
                014274  042105  047455  024516
                014302     000
  1281          014303     200  042502  040503  WARN1:  .ASCIZ  <CRLF>#BECAUSE "S1-8" IS SET, HARDWARE ERROR 1 IS NOT BEING TESTED#<CRLF
                014310  051525  020105  051442
                014316  026461  021070  044440
                014324  020123  042523  026124
                014332  044040  051101  053504
                014340  051101  020105  051105
                014346  047522  020122  020061
                014354  051511  047040  052117
```

```
                014362  041040  044505  043516
                014370  052040  051505  042524
                014376  100104     000
  1282          014401     200  042524  052123  DH1:    .ASCIZ  <CRLF>#TEST    ERRPC   IB ADR   IBS     IBD#
                014406  020040  020040  051105
                014414  050122  020103  020040
                014422  041111  040440  051104
                014430  020040  044440  051502
                014436  020040  020040  044440
                014444  042102     000
  1283
  1284          014447     200  042524  052123  DH3:    .ASCIZ  <CRLF>#TEST    ERRPC   GOOD    BAD#
                014454  020040  020040  051105
                014462  050122  020103  020040
                014470  047507  042117  020040
                014476  020040  040502  000104
  1285
  1286          014504  052200  051505  020124  DH5:    .ASCIZ  <CRLF>#TEST    ERRPC   IB ADDR#
                014512  020040  042440  051122
                014520  041520  020040  044440
                014526  020102  042101  051104
                014534     000
  1287
  1288          014535     200  042524  052123  DH7:    .ASCIZ  <CRLF>#TEST    ERRPC   TO      FROM ADDR.#
                014542  020040  020040  051105
                014550  050122  020103  020040
                014556  047524  020040  020040
                014564  020040  051106  046517
                014572  040440  042104  027122
                014600     000
  1289
  1290                  014602                          .EVEN
  1291
  1292          014602  001174  001116  001362  DT1:    .WORD   $TESTN,$ERRPC,IBS
  1293
  1294          014610  000000                  IBSA:   .WORD   0
  1295
  1296          014612  000000  000000          IBDA:   .WORD   0,0
  1297
  1298          014616  001174  001116  001124  DT3:    .WORD   $TESTN,$ERRPC,$GDDAT,$BDDAT,0
                014624  001126  000000
  1299
  1300          014630  001174  001116  001362  DT5:    .WORD   $TESTN,$ERRPC,IBS,0
                014636  000000
  1301
  1302          014640  001174  001116  013172  DT7:    .WORD   $TESTN,$ERRPC,TRTO,TRFRO,0
                014646  013174  000000
  1303
  1304          014652  000000  000000          DF0:    .WORD   0,0
  1305
  1306                  000001                          .END
```

```
ABASE = 160150              48#     60     165     166     167     168     173     174
ACDW1 = 000000              60
ACDW2 = 000000              60
ACPUOP= 000010              51#     60
ADDW0 = 000000              60
ADDW1 = 000000              60
ADDW10= 000000              60
ADDW11= 000000              60
ADDW12= 000000              60
ADDW13= 000000              60
ADDW14= 000000              60
ADDW15= 000000              60
ADDW2 = 000000              60
ADDW3 = 000000              60
ADDW4 = 000000              60
ADDW5 = 000000              60
ADDW6 = 000000              60
ADDW7 = 000000              60
ADDW8 = 000000              60
ADDW9 = 000000              60
ADEVCT= 000000              60
ADEVM = 000000              60
AENV  = 000000              60
AENVM = 000000              60
AFATAL= 000000              60
AMADR1= 000000              60
AMADR2= 000000              60
AMADR3= 000000              60
AMADR4= 000000              60
AMAMS1= 000000              60
AMAMS2= 000000              60
AMAMS3= 000000              60
AMAMS4= 000000              60
AMSGAD= 000000              60
AMSGLG= 000000              60
AMSGTY= 000000              60
AMTYP1= 000000              60
AMTYP2= 000000              60
AMTYP3= 000000              60
AMTYP4= 000000              60
APASS = 000000              60
APRIOR= 000200              50#     60
APTCSU= 000040            1176    1177#
APTENV= 000001            1168    1176    1177#
APTSIZ= 000200             212    1177#
APTSPO= 000100            1176    1177#
ASWREG= 000000              60
ATESTN= 000000              60
AUNIT = 000000              60
AUSWR = 000000              60
AVECT1= 000420              49#     60     169     170     171     172     181     182     183     184     185     186     187
                           188
AVECT2= 000000              60
BIT0  = 000001              46#    397     398     407     474     620     751     800     855    1143
BIT00 = 000001              46#
BIT01 = 000002              46#
```

```
BIT02 = 000004              46#   439   440   449
BIT03 = 000010              46#   460   461   465
BIT04 = 000020              46#
BIT05 = 000040              46#
BIT06 = 000100              46#  1108
BIT07 = 000200              46#
BIT08 = 000400              46#  1170
BIT09 = 001000              46#  1168  1170
BIT1  = 000002              46#   418   419   428   622   754
BIT10 = 002000              46#   398   474   574   761   782   790   955  1053
BIT11 = 004000              46#   763   771  1170
BIT12 = 010000              46#   745
BIT13 = 020000              46#   748  1063  1072  1076  1091  1168
BIT14 = 040000              46#   751   804  1170
BIT15 = 100000              46#   572   573   585   754   760   770   960  1031
BIT2  = 000004              46#   624   745  1089  1110
BIT3  = 000010              46#   626   748   825  1071  1075
BIT4  = 000020              46#   506   507   515   620   622   624   626   628   630   632   634   668
                            68    836   878  1144
BIT5  = 000040              46#   484   485   495   620   622   624   626   628   630   632   634   668
                           683    800   836   878   931  1007  1143
BIT6  = 000100              46#   529   530   540   632   855   879   931   960  1007
BIT7  = 000200              46#   551   552   561   634   668   789   836
BIT8  = 000400              46#   656   663   678   685
BIT9  = 001000              46#   485   671   690
BPTVEC= 000014              46#
CKSWR = 104407             1168  1170  1255#
CPUDLY  001436              190#  264*  1167
CPUSPD  001440              191#  264
CR    = 000015              46#  1176
CRLF  = 000200              46#   271  1176  1281  1282  1284  1286  1288
DDISP = 177570              46#    60   212
DEL50   010100              273   472   748   826  1167#
DF0     014652               68    73    78    83    88    93    98   100   102  1304#
DH1     014401               66    71  1282#
DH3     014447               76    81    91   100   102  1284#
DH5     014504               86  1286#
DH7     014535               96  1288#
DISPLA  001142               60#  212*  1168* 1170*
DISPRE  000174               38#  212
DSWR  = 177570               46#    60   212
DT1     014602               67    72  1292#
DT3     014616               77    82    92   100   102  1298#
DT5     014630               87  1300#
DT7     014640               97  1302#
EMTVEC= 000030               46#  212*   214*
EM1     013256               65  1259#
EM10    013477              100  1272#
EM11    014023              102  1277#
EM2     013304               70  1261#
EM3     013332               75  1263#
EM4     013354               80  1265#
EM5     013376               85  1267#
EM6     013426               90  1268#
EM7     013455               95  1270#
EOP     007756             1001  1058  1163#
```

```
ERRVEC= 000004   46#  212*  217*  218*  283   284*  290*  304*  317   318*  324*  342*  1170*
GNS   = ****** U 271   1255
GTSWR = 104406   271   1255#
HT    = 000011   46#   1176
IBCA    001370   168#  225*  226*  326   386
IBD     001364   166#  221*  222*  223   230   292   361   620*  622*  624*  626*  628*  630*
                 632*  634*  640*  642   655*  656   669*  670   684   685   690   745   748
                 751   754   763   771   782   790   802*  837*  839   902*  932*  1145*
IBDA    014612   230*  1296#
IBD2    001404   174#  231*  232*  1146
IBS     001362   165#  220*  221   229   288   350   396*  397*  399   407*  409   417*  418*
                 420   428*  430   438*  439*  441   449*  451   459*  460*  462   465*  466
                 475   483*  484*  486   495*  497   505*  506*  508   515*  517   527*  529*
                 531   540*  541   550*  551*  553   561*  563   571*  572*  577   585*  587
                 620*  622*  624*  626*  628*  630*  632*  634*  639*  654*  662   668*  677
                 683*  695*  697   745*  748*  751*  754*  759*  760*  770*  777*  781*  789*
                 799*  800*  804   813*  815   824*  825*  828   836*  852*  855*  872*  878*
                 879*  894*  921*  927*  931*  949*  954*  960*  977*  1002*  1024*  1030*  1046*
                 1063  1070*  1072  1076  1087*  1091  1103*  1108*  1121*  1141*  1143*  1158*  1292
                 1300
IBSA    014610   229*  1294#
IBS2    001402   173#  231   1003*  1007*  1017*  1025*  1031*  1047*  1052*  1059*  1071*  1075*  1088*
                 1089*  1102*  1110*  1120*  1142*  1144*  1159*
IBWC    001366   167#  223*  224*  225   321   378
IOTRD   013126   34    217   1202#
IOTVEC= 000020   46#   212*  213*
LF    = 000012   46#   1176
PIRQ  = 177772   46#
PIRQVE= 000240   46#
PRA     001416   181#  246*  247*  248   266*  928*  950*  1105*  1119*
PRA2    001426   185#  254*  255*  256
PRB     001420   182#  248*  249*  250   267*  958*  978*  1027*  1045*
PRB2    001430   186#  256*  257*  258
PRC     001422   183#  250*  251*  252   268*  853*  873*  876*  895*  899*
PRC2    001432   187#  258*  259*  260   1004*  1018*
PRD     001424   184#  252*  253*  269*  900*  922*
PRD2    001434   188#  260*  261*
PR0   = 000000   46#
PR1   = 000040   46#
PR2   = 000100   46#
PR3   = 000140   46#
PR4   = 000200   46#
PR5   = 000240   46#
PR6   = 000300   46#
PR7   = 000340   46#
PS    = 177776   46#
PSW   = 177776   46#
PWRVEC= 000024   46#   212*  216*  1179*
RDCHR = 104410   1171  1255#
RDLIN = 104411   1255#
RESVEC= 000010   46#
RSTART  002354   262#  1165
STACK = 001100   46#   212
START   001460   44    212#
STKLMT= 177774   46#
SWR     001140   60#   212*  271   574   761   955   1053  1168  1170  1171*  1179*
```

```
SWREG   000176              39#    212     271     1171
SW0   = 000001              46#
SW00  = 000001              46#
SW01  = 000002              46#
SW02  = 000004              46#
SW03  = 000010              46#
SW04  = 000020              46#
SW05  = 000040              46#
SW06  = 000100              46#
SW07  = 000200              46#
SW08  = 000400              46#
SW09  = 001000              46#
SW1   = 000002              46#
SW10  = 002000              46#
SW11  = 004000              46#
SW12  = 010000              46#
SW13  = 020000              46#
SW14  = 040000              46#
SW15  = 100000              46#
SW2   = 000004              46#
SW3   = 000010              46#
SW4   = 000020              46#
SW5   = 000040              46#
SW6   = 000100              46#
SW7   = 000200              46#
SW8   = 000400              46#
SW9   = 001000              46#
TBITVE= 00001               46#
TKVEC = 0000(               46#
TPVEC = 000064              46#
TRAPVE= 000034              46#    212*    215*
TRFRO   013174             1214*  1253#   1302
TRTO    013172             1202*  1203*   1205    1252#   1302
TRTVEC= 000014              46#
TST1    002572             277#
TST10   003304             426     431     436#
TST11   003374             447     452     458#
TST12   003520             471     477     481#
TST13   003610             493     498     504#
TST14   003700             514     518     523#
TST15   004002             537     542     548#
TST16   004072             559     564     570#
TST17   004176             584     588     620#
TST2    002716             315#
TST20   004270             620     622#
TST21   004362             622     624#
TST22   004454             624     626#
TST23   004546             626     628#
TST24   004640             628     630#
TST25   004732             630     632#
TST26   005024             632     634#
TST27   005116             634     637#
TST3    003002             345#
TST30   005150             643     652#
TST31   005404             660     667     675     682     689     694     699     745#
TST32   005472             745     748#
```

```
TST33   005560              748     751#
TST34   005646              751     754#
TST35   005734              754     758#
TST36   006020              779#
TST37   006062              787     791     797#
TST4    003032              351     356#
TST40   006114              805     811#
TST41   006144              816     822#
TST42   006210              829     834#
TST43   006246              840     850#
TST44   006364              875#
TST45   006626              896     925#
TST46   006760              953#
TST47   007116              956     984#
TST5    003062              362     375#
TST50   007232             1021#
TST51   007346             1050#
TST52   007420             1064    1068#
TST53   007472             1074    1077    1083#
TST54   007524             1092    1098#
TST55   007640             1135#
TST56   007754             1153    1162#
TST6    003124              384     387     394#
TST7    003214              405     410     415#
TYPDS = 104405             1165    1255#
TYPE  = 104401              271    1057    1165    1168    1169    1171    1173    1175    1176    1179    1255#
TYPOC = 104402             1169    1171    1255#
TYPON = 104404             1255#
TYPOS = 104403             1255#
VECTA   001372              169#    227*    228*    233     246     266*    929*    950*   1106*   1119*
VECTA2  001406              175#    239     254
VECTB   001374              170#    233*    234*    235     267*    959*    978*   1028*   1045*
VECTB2  001410              176#    239*    240*    241
VECTC   001376              171#    235*    236*    237     268*    854*    873*    877*    895*    899*
VECTC2  001412              177#    241*    242*    243    1005*   1018*
VECTD   001400              172#    237*    238*    269*    901*    922*
VECTD2  001414              178#    243*    244*
WARN1   014303             1057    1281#
$APTHD  001000               58#
$ASTAT= ****** U           1177
$ATYC   012526             1177#
$ATY1   012502             1177#
$ATY3   012510             1176    1177#
$ATY4   012520             1168    1177#
$AUTOB  001134               60#    271*    1171
$BASE   001244               60#    220
$BDADR  001122               60#
$BDDAT  001126               60#    350*    361*    378*    386*    399*    400     409*    420*    421     430*    441*    442
                            451*    462*    463     466*    467     475*    476     486*    487     497*    508*    509     517*
                            531*    532     541*    553*    554     563*    577*    578     587*    620*    622*    624*    626*
                            628*    630*    632*    634*    642*    662*    663     670*    671     677*    678     697*    698
                           1139*   1146*   1147    1298
$CDW1   001250               60#    999
$CHARC  012476             1176#*
$CKSWR  010710             1171#   1255
$CMTAG  001100               60#    212
```

```
$CM3  = 000000            60#
$CNTLG  011445          1171#
$CNTLU  011440          1171#
$CPUOP  001216            60#   262
$CRLF   001165            60#  1168    1169    1171    1176
$DBLK   012136          1175#
$DEVCT  001200            60#
$DEVM   001246            60#
$DOAGN  010054          1165#
$DTBL   012126          1175#
$ENDAD  010044            54   1165#
$ENDCT  010012          1165#
$ENDMG  010063          1165#
$ENULL  010060          1165#
$ENV    001210            60#   271    1168    1176    1177
$ENVM   001211            60#   212    1176    1177
$EOP    007756          1165#
$EOPCT  010004          1165#
$ERFLG  001103            60#  1168*   1170*
$ERMAX  001115            60#   212*   1170*
$ERROR  010120           212   1168#
$ERRPC  001116            60#  1168*   1169    1292    1298    1300    1302
$ERRTB  001252            60#  1169
$ERRTY  010272          1168   1169#
$ERTTL  001112            60#  1168*
$ESCAP  001162            60#   212*   1168    1170*
$ETABL  001210            60#
$ETEND  001252            58     60#
$FATAL  001172            60#  1177*
$FFLG   012746          1177#*
$FILLC  001156            60#  1176
$FILLS  001155            60#  1176
$GDADR  001120            60#
$GDDAT  001124            60#   349*   360*   377*   398*   400    408*   419*   421    429*   440*   442    450*
                         461*   463    467    474*   476    485*   487    496*   507*   509    516*   530*   532
                         539*   552*   554    562*   573*   576*   578    586*   620*   622*   624*   626*   628*
                         630*   632*   634*   641*   661*   676*   696*   698   1138*  1145   1147   1155*  1298
$GET42  010034          1165#
$GTSWR  010760          1171#  1255
$HD   = 000001            12
$HIBTS  001000            58#
$ICNT   001104            60#  1170*
$ILLUP  013110          1179#
$INTAG  001135            60#  1171
$ITEMB  001114            60#  1168*   1169
$LF     001166            60#  1168    1171    1176
$LFLG   012745          1177#*
$LPADR  001106            60#   212*   277*   1170*
$LPERR  001110            60#   212*   281*   1137*   1168    1170*
$MADR1  001222            60#
$MADR2  001226            60#
$MADR3  001232            60#
$MADR4  001236            60#
$MAIL   001170            58     60#    212     271    1168    1170    1176
$MAMS1  001220            60#
$MAMS2  001224            60#
```

```
$MAMS3  001230            60#
$MAMS4  001234            60#
$MBADR  001002            58#
$MFLG   012744          1177#*
$MNEW   011463          1171#
$MSGAD  001204            60#   1177*
$MSGLG  001206            60#   1177*
$MSGTY  001170            60#   1177*
$MSWR   011452          1171#
$MTYP1  001221            60#
$MTYP2  001225            60#
$MTYP3  001231            60#
$MTYP4  001235            60#
$MXCNT  010706          1170#
$NULL   001154            60#   1176
$NWTST= 000001           277#    315#    345#    356#    375#    394#    415#    436#    458#    481#    504#    523#    548#
                         570#    620#    622#    624#    626#    628#    630#    632#    634#    637#    652#    745#    748#
                         751#    754#    758#    779#    797#    811#    822#    834#    850#    875#    925#    953#    984#
                        1021#   1050#   1068#   1083#   1098#   1135#   1162#
$OCNT   011716          1173#*
$OMODE  011720          1173#*
$OVER   010672          1170#
$PASS   001176            60#    212*   1055   1165*   1170
$PASTM  001006            58#
$POWER  013116          1179#
$PWRDN  012750           212    1179#
$PWRMG  013104          1179#
$PWRUP  013022          1179#
$QUES   001164            60#   1168    1171    1176
$RDCHR  011172          1171#   1255
$RDDEC= ****** U        1255
$RDLIN  011322          1171#   1255
$RDOCT= ****** U        1255
$RDSZ = 000010          1171#
$RTNAD  010056          1165#
$R2A  = ****** U        1255
$SAVRE= ****** U        1255
$SAVR6  013114          1179#*
$SCOPE  010426           212    1170#
$SETUP= 000117           210#    212     271    1165    1168    1170    1171
$STUP = 177777           210#
$SVLAD  010636          1170#
$SVPC = 000204            54#
$SWR  = 165400            10#     12      14      60     212     277     315     345     356     375     394     415     436
                          458     481     504     523     548     570     620     622     624     626     628     630     632
                          634     637     652     745     748     751     754     758     779     797     811     822     834
                          850     875     925     953     984    1021    1050    1068    1083    1098    1135    1162    1165
                         1168    1170    1179
$SWREG  001212            60#    212
$SWRMK= 000000            14    1170
$TESTN  001174            60#    280*   1170*   1292    1298    1300    1302
$TIMES  001160            60#    212*    277*    315*    345*    356*    375*    811*    822*    834*   1068*   1165*   1170*
$TKB    001146            60#   1171    1176
$TKS    001144            60#   1171*   1176
$TN   = 000057            12#     52#    277#    315#    345#    351     356#    362     375#    384     387     394#    405
                          410     415#    426     431     436#    447     452     458#    471     477     481#    493     498
```

```
                          504#    514     518     523#    537     542     548#    559     564     570#    584     588     620#
                          622#    624#    626#    628#    630#    632#    634#    637#    643     652#    660     667     675
                          682     689     694     699     745#    748#    751#    754#    758#    779#    787     791     797#
                          805     811#    816     822#    829     834#    840     850#    875#    896     925#    953#    956
                          984#    1021#   1050#   1064    1068#   1074    1077    1083#   1092    1098#   1135#   1153    1162#
$TPB    001152             60#    1176*
$TPFLG  001157             60#    1176
$TPS    001150             60#    1176
$TRAP   013176            212     1255#
$TRAP2  013220           1255#
$TRP  = 000012           1255#
$TRPAD  013232           1255#
$TSTM   001004             58#
$TSTNM  001102             60#     279*    1165*   1168    1170*
$TTYIN  011430           1171#
$TYPBN= ****** U         1255
$TYPDS  011722           1175#   1255
$TYPE   012146           1176#   1177    1255
$TYPEC  012360           1171    1176#
$TYPEX  012500           1176#
$TYPOC  011520           1173#   1255
$TYPON  011534           1173#   1255
$TYPOS  011474           1173#   1255
$UNIT   001202             60#
$UNITM  001010             58#
$USWR   001214             60#
$VECT1  001240             60#     227
$VECT2  001242             60#
$XOFF = 000023           1176
$XON  = 000021           1171    1176
$XTSTR  010442           1170#
$SGET4= 000000           1165#
$OFILL  011717           1173#*
$40CAT= ****** U         1168    1170
.     = 014656             18#     29      30#     33#     35#     37#     40#     43#     54#     56#     58#     60#     212
                           271#    1165    1168    1169#   1170    1171#   1175#   1176    1177#   1179    1290#
.$ASTA= ****** U         1177
.$X   = 001000             58#
```

```
BDT       592#  620   622   624   626   628   630   632   634
COMMEN     46#
DELMA     142#  1167
DFC        61#    68    73    78    83    88    93    98
ECB       117#  299   303   333   340   354   365   383   391   404   413   425   434   446   455
           492   501   513   521   536   545   558   567   583   591   620   622   624   626   628
           630   632   634   649   745   748   751   754   768   776   786   794   808   820   844
           869   893   917   946   974  1014  1040  1095  1115  1152  1250
ENDCOM     46#
ERC       103#  296   301   330   337   352   363   380   388   402   411   423   432   444   453
           469   478   490   499   511   519   534   543   556   565   580   589   620   622   624
           626   628   630   632   634   644   658   665   673   680   687   692   700   745   748
           751   754   765   773   784   792   806   817   830   841   867   891   915   944   971
          1010  1036  1065  1078  1093  1113  1149  1218
ERROR      46#  296   301   330   337   352   363   380   388   402   411   423   432   444   453
           469   478   490   499   511   519   534   543   556   565   580   589   620   622   624
           626   628   630   632   634   644   658   665   673   680   687   692   700   745   748
           751   754   765   773   784   792   806   817   830   841   867   891   915   944   971
          1010  1036  1065  1078  1093  1113  1149  1218
ESCAPE     46#
GETPRI     46#
GETSWR     46#  271#
MULT       46#
NEWTST     46#  277   315   345   356   375   394   415   436   458   481   504   523   548   570
           620   622   624   626   628   630   632   634   637   652   745   748   751   754   758
           779   797   811   822   834   850   875   925   953   984  1021  1050  1068  1083  1098
          1135  1162
POP        46#  1175  1177  1179
POPSP2    127#  295   300   329   336   871   898   920   948   976  1016  1043  1118  1216
PR        130#  305   525   856   880   904   930   935   957   964  1006  1029  1109
PUSH       46#  1175  1177  1179
REPORT     46#
RESV      137#  266   267   268   269   873   895   899   922   950   978  1018  1045  1119
SCOPE      46#  315   345   356   375   394   415   436   458   481   504   523   548   570   620
           622   624   626   628   630   632   634   637   652   745   748   751   754   758   779
           797   811   822   834   850   875   925   953   984  1021  1050  1068  1083  1098  1135
          1162  1165
SETPRI     46#
SETTRA    1255#
SETUP      46#  212
SIGC      702#  745   748   751   754
SKIP       46#  351   362   384   387   405   410   426   431   447   452   471   477   493   498
           510   514   518   537   542   559   564   579   584   588   620   622   624   626   628
           630   632   634   643   660   664   667   675   682   689   694   699   745   748   751
           754   762   764   769   772   787   791   805   816   829   840   896   956  1064  1074
          1077  1092  1153
SLASH      46#
SPACE      46#
STARS      46#   54    58    60   277   315   345   356   375   394   415   436   458   481   504
           523   548   570   620   622   624   626   628   630   632   634   637   652   745   748
           751   754   758   779   797   811   822   834   850   875   925   953   984  1021  1050
          1068  1083  1098  1135  1162  1165  1168  1169  1170  1171  1173  1175  1176  1177  1179
          1255
SWRSU      46#  212#
TRMTRP    1255#
TYPBIN     46#
```

```
TYPDEC     46#  1165
TYPNAM     46#   271
TYPNUM     46#
TYPOCS      6#    46#
TYPOCT     46#  1169   1171
TYPTXT     46#
WARN      985#   998   1022   1051   1069   1085   1100
ZZ1       307#   315    368#   375   1123#  1135
$$CMRE     60#
$$CMTM     60#
$$ESCA     46#
$$NEWT     46#   277    315    345    356    375    394    415    436    458    481    504    523    548    570
           620   622    624    626    628    630    632    634    637    652    745    748    751    754    758
           779   797    811    822    834    850    875    925    953    984   1021   1050   1068   1083   1098
          1135  1162
$$SET    1255#
$$SETM    212#
$$SKIP     46#   351    362    384    387    405    410    426    431    447    452    471    477    493    498
           514   518    537    542    559    564    584    588    620    622    624    626    628    630    632
           634   643    660    667    675    682    689    694    699    745    748    751    754    787    791
           805   816    829    840    896    956   1064   1074   1077   1092   1153
.EQUAT      6#    46
.HEADE      5#    12
.SETTR      5#
.SETUP      5#   210
.SWRHI      7#    14
.SWRLO     14#
.TRMTR      5#
.$ACT1      9#    54
.$APTB     60#
.$APTH      9#    58
.$APTY      9#  1177
.$CATC      6#
.$CMTA      7#    60
.$EOP       7#  1165
.$ERRO      7#  1168
.$ERRT      7#  1169
.$POWE      6#  1179
.$RDOC      5#
.$READ      8#  1171
.$SCOP      8#  1170
.$TRAP      5#  1255
.$TYPB      6#
.$TYPD      8#  1175
.$TYPE      8#  1176
.$TYPO      6#  1173

. ABS.  014656     000     OVR   RW    REL   LCL   D

 ERRORS DETECTED:  0

 CVIBAB,CVIBAB/CRF=CVIBAB
 RUN-TIME: 24 10 1 SECONDS
 RUN-TIME RATIO: 208/36=5.6
```

CORE USED: 26K (51 PAGES)