IBV11-A

IBV11-A DIAGNOSTIC

MD-11-DVIBA-A

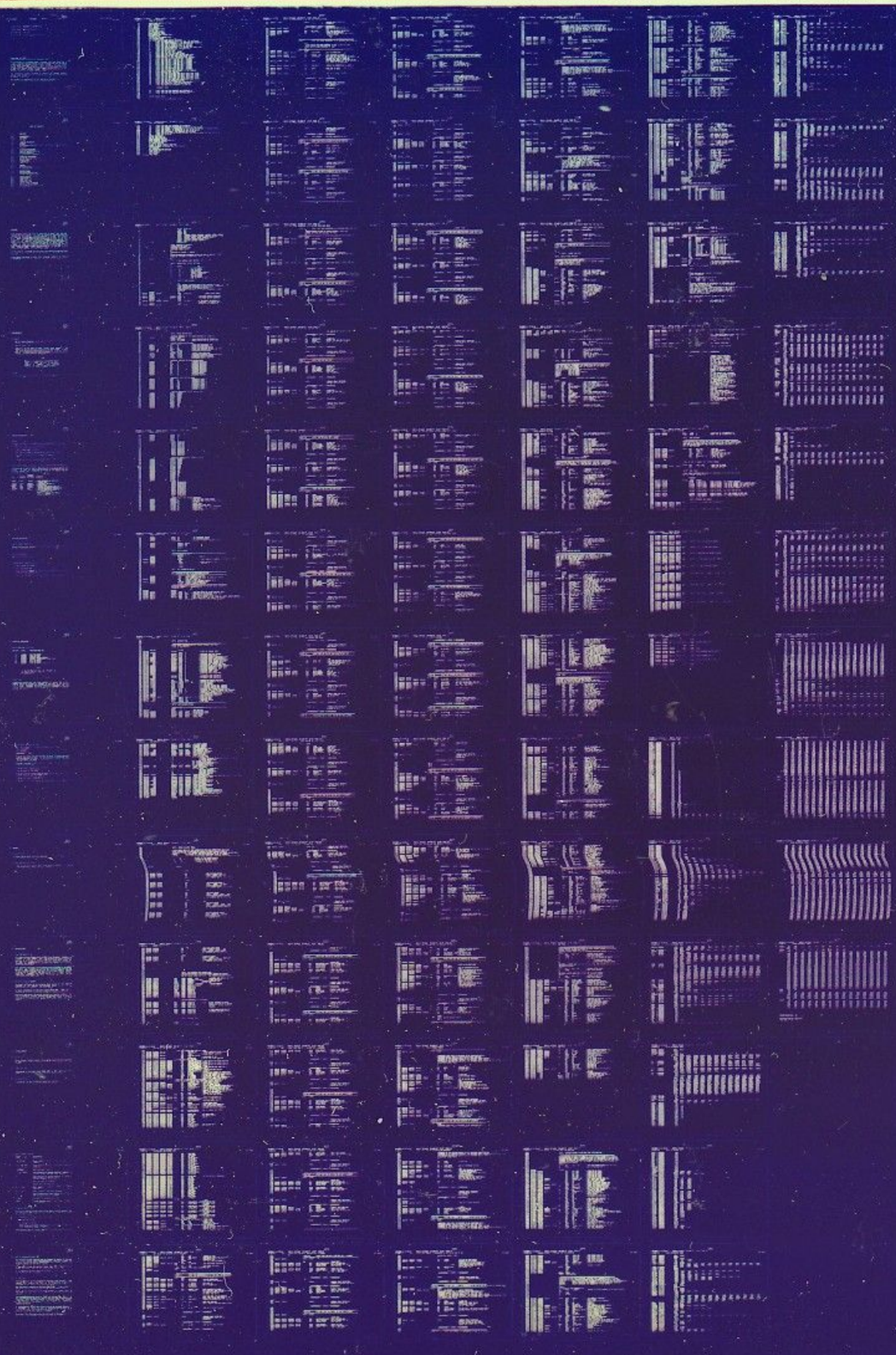
EP-DVIBA-A-DL-A

FICHE 1 OF 1

JUN 1977

digital

MADE IN USA



Product Code: MAINDEC-11-DVIBA-A-D

Product Name: IBV11-A Diagnostic

Date Created: JAN 1977

Maintainer: Diagnostic Engineering

TABLE OF CONTENTS

SEQ 0003

1.0 ABSTRACT

This program allows the user to check-out or debug the IBV11-A, LSI/IB interface option. In order to check-out a greater portion of logic on this option, a second IBV11-A is needed. See section 2.1. When a second IBV11-A can be obtained in order to run this diagnostic, 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. 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", "DA11" 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 register 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 be electrically second on the LSI-11-BUS. It 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 "SCDW1". No test will be performed that requires the "KGM" if SCDW1 is zero.

2.2 Storage

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

SEQ 0005

3.0 LOADING PROCEDURE

3.1 Method

Standard procedure for normal binary tapes should be followed.

1. Absolute loader must be in memory.
2. Place binary tape in reader.
3. Type address *7500 (* determine by location of loader).
4. Type "G" (program will be loaded into memory).

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:	1250	160150	;;BASE ADDRESS OF EQUIPMENT ;; UNDER TEST
$VECT1:	1244	000640	;;INTERRUPT VECTOR #1
SWREG:	176	000000	;;MANUAL SWR.
IBS2:	1366	160160	; ADDRESS OF SECOND IBV11-A.
VECTA2:	1372	660	; VECTOR ADDRESS OF SECOND IBV11-A.
$CDW1:	1254	000000	;;DEVICE DESCRIPTOR WORD #1 (If = 000001 to use "KGM" in testing 1st IBV-11) (Default = 00000 to disable use of KGM in tests.

SEQ 0006

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. 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	BELL ON ERROR
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 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", "SSWREG" should be altered to "060000" to loop on current test and inhibit error typeout, then type "P" to continue program execution.

5.3 Program and/or Operator Action

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

   MD-11-DVIBA-A

   SWR=000000 NEW=

2. Program now waits for the operator to enter a switch register setting (see section 8.6). If the program is restarted, only "MD-11-DVIBA-A" is typed. 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.

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

0.1 Minutes (6 sec) Iteration Inhibited - No Errors
0.5 Minutes (30 sec) With Iterations - 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.
SN/ 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 [illegible], 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".

```
                                .NLIST  MC,MD,CND
                                .LIST   ME
                                .ENABL  ABS
                                .ENABL  AMA
                                .MCALL  .HEADER,.SETUP,.SETTRAP,.TRMTRP,.STRAP,.SRDOCT
                                .MCALL  .STYPBIN,TYPOCS,.SPOWER,.SCATCH,.STYPOCT,.EQUAT
                                .MCALL  .SCMTAG,.SWRHI,.SEOP,.SERROR,.SERRTYP
                                .MCALL  .STYPDEC,.SSCOPE,.SREAD,.STYPE
                                .MCALL  .SACT11,.SAPTHDR,.SAPTYPE
167400                          $SWR=167400

                        .TITLE  MAINDEC-11-DVIBA-A
                        ;*COPYRIGHT (C) 1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY EDWARD C. BADGER
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                        ;*
000001                  $TN=1
                                ;THIS VERSION LAST EDITED - OCT. 6, 1976

                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         8            LOOP ON TEST IN SWR<7:0>

                                .SBTTL  TRAP CATCHER

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                          .=4
000004  012066  000200          .WORD   IOTRD,200               ;HANDLE BUSS ERROR.
000174                          .=174
000174  000000          DISPREG:.WORD   0               ;;SOFTWARE DISPLAY REGISTER.
000176  000000          SWREG:  .WORD   0               ;;SOFTWARE SWITCH REGISTER.
000100                          .=100
000100  000104  000200  000002  .WORD   104,200,2       ;IF "B EVENT" ON Q-BUS IS
                                                        ;CONNECTED, WE NEED A WAY OF
000200                          .=200                   ;IGNORING ITS INTERRUPTS.
000200  000137  001422          JMP     START
```

```
                        .SBTTL  BASIC DEFINITIONS

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

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
```

```
                        SW04=   20
                        SW03=   10
                        SW02=   4
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
000001                  BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
```

```
                        TKVEC=  60              ;;TTY KEYBOARD VECTOR
                        TPVEC=  64              ;;TTY PRINTER VECTOR
                        PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR

                                ABASE=  160150
                                AVECT1= 640
                                APRIOR= 200
                                $TN=1

                        .SBTTL  ACT11 HOOKS

                        ;;******************************************************************
                        ;HOOKS REQUIRED BY ACT11
                                $SVPC=.                 ;SAVE PC
                                .=46
                                $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
                                .=52
                                .WORD   0               ;;2)SET LOC.52 TO ZERO
                                .=$SVPC                 ;; RESTORE PC

                                .=1000

                        .SBTTL  APT PARAMETER BLOCK

                        ;;******************************************************************
                        ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                        ;;******************************************************************
                                .$X=.   ;;SAVE CURRENT LOCATION
                                .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
                                200     ;;FOR APT START UP
                                .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
                                $APTHDR ;;POINT TO APT HEADER BLOCK
                                .=.$X   ;;RESET LOCATION COUNTER
                        ;;******************************************************************
                        ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                        ;INTERFACE SPEC.

                        $APTHD:
                        $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
                        $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
                        $TSTM:  .WORD   60.     ;;RUN TIM OF LONGEST TEST
                        $PASTM: .WORD   120.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
                        $UNITM: .WORD   120.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
                                .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .SBTTL  COMMON TAGS

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

                                        .=1100
001100                          $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                          .WORD   0
001102  000                     $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000                     $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000                  $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000                  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000                  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000                  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000                     $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001                     $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000                  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000                  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000                  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000                  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                          .WORD   0
001134  000                     $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000                     $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                          .WORD   0
001140  177570                  SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560                  $TKS:   177560                  ;;TTY KBD STATUS
001146  177562                  $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564                  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566                  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000                     $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002                     $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012                     $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000                     $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001162  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001164  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001170  077                     $QUES:  .ASCII  /?/             ;;QUESTION MARK
001171  015                     $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001172  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001174                          $MAIL:                          ;;APT MAILBOX
001174  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001176  000000                  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001200  000000                  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001202  000000                  $PASS:  .WORD   APASS           ;;PASS COUNT
001204  000000                  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
```

```
001206 000000                 $UNIT:   .WORD    AUNIT    ;;I/O UNIT NUMBER
001210 000000                 $MSGAD:  .WORD    AMSGAD   ;;MESSAGE ADDRESS
001212 000000                 $MSGLG:  .WORD    AMSGLG   ;;MESSAGE LENGTH
001214                        $ETABLE:                   ;;APT ENVIRONMENT TABLE
001214    000                 $ENV:    .BYTE    AENV     ;;ENVIRONMENT BYTE
001215    000                 $ENVM:   .BYTE    AENVM    ;;ENVIRONMENT MODE BITS
001216 000000                 $SWREG:  .WORD    ASWREG   ;;APT SWITCH REGISTER
001220 000000                 $USWR:   .WORD    AUSWR    ;;USER SWITCHES
001222 000000                 $CPUOP:  .WORD    ACPUOP   ;;CPU TYPE,OPTIONS
                              ;*                         BITS 15-11=CPU TYPE
                              ;*                                  11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                              ;*                                  11/70=06,PDQ=07,Q=10
                              ;*                         BIT 10=REAL TIME CLOCK
                              ;*                         BIT  9=FLOATING POINT PROCESSOR
                              ;*                         BIT  8=MEMORY MANAGEMENT
001224    000                 $MAMS1:  .BYTE    AMAMS1   ;;HIGH ADDRESS,M.S. BYTE
001225    000                 $MTYP1:  .BYTE    AMTYP1   ;;MEM. TYPE,BLK#1
                              ;*                         MEM.TYPE BYTE   --  (HIGH BYTE)
                              ;*                                  900 NSEC CORE=001
                              ;*                                  300 NSEC BIPOLAR=002
                              ;*                                  500 NSEC MOS=003
001226 000000                 $MADR1:  .WORD    AMADR1   ;;HIGH ADDRESS,BLK#1
                              ;*                           MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001230    000                 $MAMS2:  .BYTE    AMAMS2   ;;HIGH ADDRESS,M.S. BYTE
001231    000                 $MTYP2:  .BYTE    AMTYP2   ;;MEM.TYPE,BLK#2
001232 000000                 $MADR2:  .WORD    AMADR2   ;;MEM.LAST ADDRESS,BLK#2
001234    000                 $MAMS3:  .BYTE    AMAMS3   ;;HIGH ADDRESS,M.S.BYTE
001235    000                 $MTYP3:  .BYTE    AMTYP3   ;;MEM.TYPE,BLK#3
001236 000000                 $MADR3:  .WORD    AMADR3   ;;MEM.LAST ADDRESS,BLK#3
001240    000                 $MAMS4:  .BYTE    AMAMS4   ;;HIGH ADDRESS,M.S.BYTE
001241    000                 $MTYP4:  .BYTE    AMTYP4   ;;MEM.TYPE,BLK#4
001242 000000                 $MADR4:  .WORD    AMADR4   ;;MEM.LAST ADDRESS,BLK#4
001244 000640                 $VECT1:  .WORD    AVECT1   ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001246 000000                 $VECT2:  .WORD    AVECT2   ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001250 160150                 $BASE:   .WORD    ABASE    ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001252 000000                 $DEVM:   .WORD    ADEVM    ;;DEVICE MAP
001254 000000                 $CDW1:   .WORD    ACDW1    ;;CONTROLLER DESCRIPTION WORD#1
001256                        $ETEND:
                              .MEXIT
```

```
                                .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


001256                          $ERRTB:

                                        ;ITEM   1

001256  012216                          EM1                     ;IBS FUNCTION ERROR
001260  012437                          DH1                     ;TEST    ERRPC   IB ADDR
001262  012640                          DT1                     ;$TESTN,$ERRPC,IBS
001264  012710                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   2

001266  012244                          EM2                     ;IBD FUNCTION ERROR
001270  012437                          DH1                     ;TEST    ERRPC   IB ADDR
001272  012640                          DT1                     ;$TESTN,$ERRPC,IBS
001274  012710                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   3

001276  012272                          EM3                     ;IBS DATA ERROR
001300  012505                          DH3                     ;TEST    ERRPC   GOOD    BAD
001302  012654                          DT3                     ;$TESTN,$ERRPC,$GDDAT,$BDDAT
001304  012710                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   4

001306  012314                          EM4                     ;IBD DATA ERROR
001310  012505                          DH3                     ;TEST    ERRPC   GOOD    BAD
001312  012654                          DT3                     ;$TESTN,$ERRPC,$GDDAT,$BDDAT
001314  012710                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   5

001316  012336                          EM5                     ;IBS/IBD ADDRESS ERROR
001320  012542                          DH5                     ;TEST   ERROR PC   ADDRESS
001322  012666                          DT5                     ;$TSTNM,$ERRPC,IBS
001324  012710                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   6
```

```
                                        EM6             ;IBWC/IBCA DATA ERROR
                                        DH3             ;TEST    ERRPC    GOOD     BAD
                                        DT3             ;$TESTN,$ERRPC,$GDDAT,$BDDAT
                                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   7
                                        EM7             ;INTERRUPT ERROR
                                        DH7             ;TEST    ERRPC    TO       FROM ADDR.
                                        DT7             ;TSTNM,#ERRPC,TRTO,TRFRO
                                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM.




                                        .SBTTL  REG ADDRESS AND COMMON TAGS
                                        ;WARNING IF DEVICE # IS AT DIFFERENT ADDRESS OR VECTOR
                                        ;DO NOT PATCH THESE LOCATIONS - SEE PROGRAM DOCUMENTATION.

001346  160150          IBS:    .WORD   ABASE   ;>NO            <;CONTROL AND STATUS REGISTER.
001350  160152          IBD:    .WORD   ABASE+2 ;>PATCHES       <;DATA REGISTER.
001352  160154          IBWC:   .WORD   ABASE+4         ;ADDRESS RESERVED FOR
001354  160156          IBCA:   .WORD   ABASE+6         ;FUTURE USE
001356  000640          VECTA:  .WORD   AVECT1  ;>ALLOWED       <;VECTOR ADDRESS.
001360  000644          VECTB:  .WORD   AVECT1+4 ;>HERE!        <;VECTOR ADDR. +4.
001362  000650          VECTC:  .WORD   AVECT1+10
001364  000654          VECTD:  .WORD   AVECT1+14
001366  160160          IBS2:   .WORD   ABASE+10
001370  160162          IBD2:   .WORD   ABASE+12
001372  000660          VECTA2: .WORD   AVECT1+20
001374  000664          VECTB2: .WORD   AVECT1+24
001376  000670          VECTC2: .WORD   AVECT1+30
001400  000674          VECTD2: .WORD   AVECT1+34

                                        ;VECTOR ADDRESSES +2 LOCATIONS.

001402  000642          PRA:    .WORD   AVECT1+2        ;NOTE:  DO NOT ATTEMPT TO PATCH
001404  000646          PRB:    .WORD   AVECT1+6        ;       THESE LOCATIONS IF A VECTOR
001406  000652          PRC:    .WORD   AVECT1+12       ;       VARYIES .   ALTER LOCATION
001410  000656          PRD:    .WORD   AVECT1+16       ;       "$VECT1:".

001412  000662          PRA2:   .WORD   AVECT1+22       ;       IF TEST MODULE VECTOR IS
001414  000666          PRB2:   .WORD   AVECT1+26       ;       DIFFERENT,YOU MUST CHANGE
001416  000672          PRC2:   .WORD   AVECT1+32       ;       LOCATION "VECTA2:".
001420  000676          PRD2:   .WORD   AVECT1+36


                                        .SBTTL  PROGRAM START


001422                  START:
                        .SBTTL  INITIALIZE THE COMMON TAGS
```

```
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001422  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001426  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001430  022706  001140                  CMP     #SWR,R6 ;;DONE?
001434  001374                          BNE     .-6             ;;LOOP BACK IF NO
001436  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                        ;;INITIALIZE A FEW VECTORS
001442  012737  007450  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001450  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001456  012737  007126  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001464  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
001472  012737  012136  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001500  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
001506  012737  011710  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
001514  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
001522  005037  001160                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
001526  005037  001162                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
001532  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
001540  012737  001540  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
001546  012737  001546  001110          MOV     #.,$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.
001554  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
001560  012737  001614  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
001566  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
001574  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
001602  022777  177777  177330          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
001610  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
001612  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
001614  012716  001622          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
001620  000002                          RTI
001622  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
001630  012737  000174  001142          MOV     #DISPREG,DISPLAY
001636  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

001642  005037  001202                  CLR     $PASS           ;;CLEAR PASS COUNT
001646  132737  000200  001215          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
001654  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
001656  012737  001216  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
001664                          67$:
001664  012737  012066  000004          MOV     #IOTRD,ERRVEC   ;SET TO HANDLE BUS ERRORS.
001672  012737  000200  000006          MOV     #200,ERRVEC+2

001700  013737  001250  001346          MOV     $BASE,IBS       ;GET BASE ADDR.
001706  013737  001346  001350          MOV     IBS,IBD         ;FIX DATA BUFFER=
001714  062737  000002  001350          ADD     #2,IBD          ;CSR+2
001722  013737  001350  001352          MOV     IBD,IBWC
001730  062737  000002  001352          ADD     #2,IBWC
001736  013737  001352  001354          MOV     IBWC,IBCA
001744  062737  000002  001354          ADD     #2,IBCA
001752  013737  001244  001356          MOV     $VECT1,VECTA    ;GET VECTOR ADDR.
001760  042737  170000  001356          BIC     #170000,VECTA   ;STRIP JUNK
001766  013737  001346  012646          MOV     IBS,IBSA
```

```
 001774  013737  001350  012650          MOV     IBD,IBDA
 002002  013737  001366  001370          MOV     IBS2,IBD2
 002010  062737  000002  001370          ADD     #2,IBD2
 002016  013737  001356  001360          MOV     VECTA,VECTB
 002024  062737  000004  001360          ADD     #4,VECTB
 002032  013737  001360  001362          MOV     VECTB,VECTC
 002040  062737  000004  001362          ADD     #4,VECTC
 002046  013737  001362  001364          MOV     VECTC,VECTD
 002054  062737  000004  001364          ADD     #4,VECTD
 002062  013737  001372  001374          MOV     VECTA2,VECTB2
 002070  062737  000004  001374          ADD     #4,VECTB2
 002076  013737  001374  001376          MOV     VECTB2,VECTC2
 002104  062737  000004  001376          ADD     #4,VECTC2
 002112  013737  001376  001400          MOV     VECTC2,VECTD2
 002120  062737  000004  001400          ADD     #4,VECTD2

 002126  013737  001356  001402          MOV     VECTA,PRA          ;SET UP VECTOR+2 ADDRESSES.
 002134  062737  000002  001402          ADD     #2,PRA
 002142  013737  001402  001404          MOV     PRA,PRB
 002150  062737  000004  001404          ADD     #4,PRB
 002156  013737  001404  001406          MOV     PRB,PRC
 002164  062737  000004  001406          ADD     #4,PRC
 002172  013737  001406  001410          MOV     PRC,PRD
 002200  062737  000004  001410          ADD     #4,PRD
 002206  013737  001372  001412          MOV     VECTA2,PRA2
 002214  062737  000002  001412          ADD     #2,PRA2
 002222  013737  001412  001414          MOV     PRA2,PRB2
 002230  062737  000004  001414          ADD     #4,PRB2
 002236  013737  001414  001416          MOV     PRB2,PRC2
 002244  062737  000004  001416          ADD     #4,PRC2
 002252  013737  001416  001420          MOV     PRC2,PRD2
 002260  062737  000004  001420          ADD     #4,PRD2
 002266                          RSTART:
                                                            ;/-RESV-
 002266  013777  001402  177062          MOV     PRA,@VECTA ;/RESTORE VECTOR FOR
 002274  012777  004700  177100          MOV     #4700,@PRA         ;/ILLEGAL INTRO.
                                                            ;/-RESV-
 002302  013777  001404  177050          MOV     PRB,@VECTB ;/RESTORE VECTOR FOR
 002310  012777  004700  177066          MOV     #4700,@PRB         ;/ILLEGAL INTRO.
                                                            ;/-RESV-
 002316  013777  001406  177036          MOV     PRC,@VECTC ;/RESTORE VECTOR FOR
 002324  012777  004700  177054          MOV     #4700,@PRC         ;/ILLEGAL INTRO.
                                                            ;/-RESV-
 002332  013777  001410  177024          MOV     PRD,@VECTD ;/RESTORE VECTOR FOR
 002340  012777  004700  177042          MOV     #4700,@PRD         ;/ILLEGAL INTRO.
                                 .SBTTL  TYPE PROGRAM NAME
                                 ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
 002346  005227  177777                  INC     #-1                ;;FIRST TIME?
 002352  001033                          BNE     64$                ;;BRANCH IF NO
 002354  104401  002422                  TYPE    ,65$               ;;TYPE ASCIZ STRING
                                 .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
 002360  005737  000042                  TST     @#42               ;;ARE WE RUNNING UNDER XXDP/ACT?
 002364  001012                          BNE     66$                ;;BRANCH IF YES
 002366  123727  001214  000001          CMPB    $ENV,#1            ;;ARE WE RUNNING UNDER APT?
```

```
                                BEQ     66$                     ;;BRANCH IF YES
                                CMP     SWR,#SWREG              ;;SOFTWARE SWITCH REG SELECTED?
                                BNE     67$                     ;;BRANCH IF NO
                                GTSWR                           ;;GET SOFT-SWR SETTINGS
                                BR      67$
002412  112737  000001  001134 66$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
002420                          67$:
002420  000410                 ;;65$:  BR      64$             ;;GET OVER THE ASCIZ
                                        .ASCIZ  <CRLF>#MD11-DVIBA-A#<CRLF>
002442                          64$:
002442  000005                         RESET

                                ;;**************************************************************
                                ;#TEST 1        #TEST THE ADDRESSABILITY OF THE IBS, IBD REGISTERS
                                ;;**************************************************************
002444  000240                 TST1:   NOP
002446  012737  000050  001160         MOV     #50,$TIMES      ;;DO 50 ITERATIONS
002454  012737  002504  001106         MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS

002462  012737  000001  001102         MOV     #1,$TSTNM       ;SET TEST #1.
002470  012737  000001  001200         MOV     #1,$TESTN       ;DON'T FORGET APT!
002476  012737  002504  001110         MOV     #1$,$LPERR

002504  013746  000004         1$:     MOV     ERRVEC,-(SP)    ;SAVE CONTENTS OF ADDR. 4
002510  012737  002536  000004         MOV     #2$,ERRVEC      ;SET TIME-OUT TRAP VECTOR TO HANDLE
                                                                ;IN CASE WE TIME OUT WHEN
                                                                ;WE ADDR. THE IBV-11

002516  005777  176624                 TST     @IBS            ;ADDR THE IBS, IF NO RESPONSE,
                                                                ;WILL TRAP TO 2$ FROM HERE
002522  012737  002544  000004         MOV     #3$,ERRVEC      ;CHANGE FOR ADDRESSING THE IBD REG.

002530  005777  176614                 TST     @IBD            ;ADDR THE IBD REG.
                                                                ;WE'LL TRAP TO 3$ FROM HERE IF BAD.
002534  000406                 BR      4$
002536                          2$:
002536  062706  000004         ADD     #4,SP           ;/ADD #4 TO STACK POINTER.

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

                                ERROR   5               ;/MODULE FAULT DETECTED:
                                                        ;IBS REGISTER COULD NOT BE
                                                        ;ADDRESSED

                                ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$
002544                          3$:
002544  062706  000004         ADD     #4,SP           ;/ADD #4 TO STACK POINTER.

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

```
                                        ERROR   5               ;/MODULE FAULT DETECTED:
                                                                ;ADDRESSED

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
002552  012637  000004          4$:   MOV     (SP)+,ERRVEC      ;RESTORE CONTENTS OF LOC 4.
                                                                ;/PR
002556  012746  000000                MOV     #0,-(SP)          ;/SET CPU PRIORITY ON RETURN
002562  012746  002570                MOV     #64$,-(SP)        ;/SHOW RETURN ADDRESS
002566  000002                        RTI                       ;/CAUSE A RETURN (PUTS NEW STATUS
002570                          64$:                            ;/IN STATUS REG.)


                                ;;**************************************************************
                                ;*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
                                ;*
                                ;;**************************************************************
002570  000004                TST2:   SCOPE
002572  012737  000010  001160        MOV     #10,$TIMES        ;;DO 10 ITERATIONS

002600  013746  000004                MOV     ERRVEC,-(SP)      ;SAVE CONTENTS OF ADDR 4.
002604  012737  002632  000004        MOV     #1$,ERRVEC        ;SET TIME OUT TRAP VECTOR TO HANDLE
                                                                ;IN CASE WE TIME OUT WHEN WE
                                                                ;ADDRESS THE IBV-11 ADDRESSES +4,+6.
002612  005777  176534                TST     @IBWC             ;TEST BASE ADDRESS +4, IF NO RESPONSE
                                                                ;WILL TRAP TO 1$ FROM HERE

002616  012737  002642  000004        MOV     #2$,ERRVEC        ;CHANGE FOR ADDRESSING +6 ADDR.

002624  005777  176524                TST     @IBCA             ;ADDR THE +6 ADDR. - TRAP IF BAD.
002630  000407                        BR      3$                ;CONTINUE IF GOOD.

002632                          1$:
002632  062706  000004                ADD     #4,SP             ;/ADD #4 TO STACK POINTER.


                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

                                        ERROR   5               ;/MODULE FAULT DETECTED:
                                                                ;BASE ADDR+4 COULD NOT
                                                                ;BE ADDRESSED.

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
002640  000403                        BR      3$

002642                          2$:
002642  062706  000004                ADD     #4,SP             ;/ADD #4 TO STACK POINTER.


                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS
```

```
                                        ERROR   5               ;/MODULE FAULT DETECTED:
                                                                ;BASE ADDR+6 COULD NOT
                                                                ;BE ADDRESSED.

                                        ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$

002650  012637  000004        3$:     MOV     (SP)+,ERRVEC    ;RESTORE CONTENTS OF LOC 4.

                                ;;**********************************************************************
                                ;*TEST 3         *TEST THAT IBS IS CLEAR AT INIT OF TESTING
                                ;;**********************************************************************
002654  000004                  TST3:   SCOPE
002656  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION

002664  000005                          RESET                   ;ISSUE SYSTEM INIT.

002666  005037  001124                  CLR     $GDDAT          ;EXPECT ZERO CSR.
002672  017737  176450  001126          MOV     @IBS,$BDDAT     ;READ CSR.
002700  001401                          BEQ     TST4            ;;

                                        ;;$$$$$$$$$$››› ERROR ‹‹‹$$$$$$$$$$

002702  104003                          ERROR   3               ;/MODULE FAULT DETECTED:
                                                                ;IBS NOT CLEAR ON INT.

                                        ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$

                                ;;**********************************************************************
                                ;*TEST 4         *TEST THAT IBD IS CLEAR AT INIT OF TESTING
                                ;;**********************************************************************
002704  000004                  TST4:   SCOPE
002706  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION

002714  000005                          RESET                   ;ISSUE SYSTEM INITIALIZE.

002716  005037  001124                  CLR     $GDDAT          ;EXPECT ZERO CSR.
002722  117737  176422  001126          MOVB    @IBD,$BDDAT     ;READ DBR
002730  001401                          BEQ     TST5            ;;

                                        ;;$$$$$$$$$$››› ERROR ‹‹‹$$$$$$$$$$

002732  104004                          ERROR   4               ;/MODULE FAULT DETECTED:
                                                                ;IBD NOT CLEAR ON INIT.

                                        ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$

                                ;;**********************************************************************
```

```
                                ;*TEST 5        *TEST THAT BASE ADDRESSES +4,+6 RETURN ZERO WHEN READ
                                ;*
                                ;*BASE ADDRESS +4 AND +6 SHOULD RETURN A ZERO WHEN
                                ;*READ, IN THIS TEST WE WILL TRY THAT.
                                ;*
                                ;;**************************************************************
002734  000004                  TST5:   SCOPE
002736  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS

002744  005037  001124          CLR     $GDDAT          ;EXPECT ZERO RETURN
002750  017737  176376  001126  MOV     @IBWC,$BDDAT    ;READ BASE ADDRESS+4
002756  001402                  BEQ     1$              ;IF ZERO - GOOD.

                                ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

002760  104006                  ERROR   6               ;/MODULE FAULT DETECTED:
                                                        ;SHOULD HAVE READ BACK ZERO FROM
                                                        ;THIS ADDR.

                                ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
002762  000405                  BR      TST6            ;;

002764  017737  176364  001126  1$:     MOV     @IBCA,$BDDAT    ;READ BASE ADDR+6, SHOULD BE ZERO
002772  001401                  BEQ     TST6            ;;

                                ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

                                ;;**************************************************************
                                ;*TEST 6        *TEST THAT WE CAN SET TCS, TCS SETS CMD
                                ;;**************************************************************
002776  000004                  TST6:   SCOPE

003000  005077  176342          CLR     @IBS            ;CLEAR CLR
003004  052777  000001  176334  BIS     #BIT0,@IBS      ;SET TCS.
003012  052737  002001  001124  BIS     #BIT0!BIT10,$GDDAT ;EXPECT ONLY TCS AND CMD TO SET
003020  017737  176322  001126  MOV     @IBS,$BDDAT     ;READ THE IBS.
003026  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;DID TCS AND CMD SET?
003034  000402                  BR      1$              ;YES - CONTINUE

                                ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

003036  104003                  ERROR   3               ;/MODULE FAULT DETECTED:
```

```
                                                        ;TCS AND/OR CMD FAILED TO SET

                                                ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$
003040  000412                                  BR      TST7                    ;;
                                                ;CLEAR TCS,
003042  042777  000001  176276  1$:     BIC     #BIT0,@IBS              ;EXPECT TCS AND CMD TO CLEAR
003050  005037  001124                  CLR     $GDDAT                  ;READ IBS, DID THEY CLEAR?
003054  017737  176266  001126          MOV     @IBS,$BDDAT             ;;
003062  001401                          BEQ     TST7

                                                ;;$$$$$$$$$$››› ERROR ‹‹‹$$$$$$$$$$

003064  104003                          ERROR   3                       ;/MODULE FAULT DETECTED:
                                                                        ;TCS AND/OR CMD FAILED TO CLEAR.

                                                ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$

                                ;;**************************************************************
                                ;*TEST 7         *TEST THAT EOP WILL SET
                                ;;**************************************************************
003066  000004                  TST7:   SCOPE

003070  005077  176252                  CLR     @IBS                    ;CLEAR CSR.
003074  052777  000002  176244          BIS     #BIT1,@IBS              ;SET EOP
003102  012737  000002  001124          MOV     #BIT1,$GDDAT            ;EXPECT ONLY EOP TO SET.
003110  017737  176232  001126          MOV     @IBS,$BDDAT             ;READ IBS
003116  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;DID EOP SET?
003124  001402                          BEQ     1$                      ;YES - CONTINUE

                                                ;;$$$$$$$$$$››› ERROR ‹‹‹$$$$$$$$$$

003126  104003                          ERROR   3                       ;/MODULE FAULT DETECTED:
                                                                        ;EOP BIT SETTING ERROR.

                                                ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$
003130  000412                          BR      TST10                   ;;

003132  042777  000002  176206  1$:     BIC     #BIT1,@IBS              ;CLEAR EOP
003140  005037  001124                  CLR     $GDDAT                  ;EXPECT A ZERO CSR.
003144  017737  176176  001126          MOV     @IBS,$BDDAT             ;READ IBS, IS IT CLEAR?
003152  001401                          BEQ     TST10                   ;;

                                                ;;$$$$$$$$$$››› ERROR ‹‹‹$$$$$$$$$$

003154  104003                          ERROR   3                       ;/MODULE FAULT DETECTED:
                                                                        ;IBS FAILED TO CLEAR

                                                ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$
```

```
                                ;;*****************************************************************
                                ;*TEST 10       *TEST THAT RE WILL SET + CLEAR
                                ;;*****************************************************************
003156  000004                  TST10:  SCOPE

003160  005077  176162                  CLR     @IBS            ;CLEAR CSR.
003164  052777  000004  176154          BIS     #BIT02,@IBS     ;SET REM
003172  012737  000004  001124          MOV     #BIT02,$GDDAT   ;EXPECT ONLY REM TO SET.
003200  017737  176142  001126          MOV     @IBS,$BDDAT     ;READ IBS.
003206  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DID REM AND ONLY REM SET?
003214  001402                          BEQ     1$              ;YES - CONTINUE

                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

003216  104003                          ERROR   3               ;/MODULE FAULT DETECTED:
                                                                ;REM BIT SETTING ERROR.

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
003220  000412                          BR      TST11           ;;
003222  042777  000004  176116  1$:     BIC     #BIT02,@IBS     ;CLEAR REM BIT.
003230  005037  001124                  CLR     $GDDAT          ;EXPECT ZERO CSR.
003234  017737  176106  001126          MOV     @IBS,$BDDAT     ;READ IBS - IS IT CLEAR?
003242  001401                          BEQ     TST11           ;;

                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

003244  104003                          ERROR   3               ;/MODULE FAULT DETECTED:
                                                                ;IBS FAILED TO CLEAR.

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS

                                ;;*****************************************************************
                                ;*TEST 11       *TEST THAT IBC WILL SET AND CLEAR
                                ;;*****************************************************************
003246  000004                  TST11:  SCOPE

003250  005077  176072                  CLR     @IBS            ;CLEAR CSR.
003254  052777  000010  176064          BIS     #BIT03,@IBS     ;SET IBC
003262  012737  000010  001124          MOV     #BIT03,$GDDAT   ;EXPECT ONLY IBC TO BE SET
003270  017737  176052  001126          MOV     @IBS,$BDDAT     ;READ IBS.
003276  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID IBS SET?
003304  001414                          BEQ     1$              ;YES CONTINUE
003306  012777  000010  176032          MOV     #BIT03,@IBS     ;TRY SETTING IBC AGAIN.
003314  017737  176026  001126          MOV     @IBS,$BDDAT     ;MEMORY REFRESH MIGHT HAVE
003322  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;GOT IN THE WAY.
003330  001402                          BEQ     1$
```

```
                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

  843
  844  003332  104003                   ERROR   3               ;/MODULE FAULT DETECTED:
  845                                                           ;IBS BIT SETTING ERROR.

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
  848  003334  000416                   BR      TST12           ;;
  849
  850  003336  004537  007106    1$:    JSR     R5,DEL50        ;DELAY 150 US.
  851  003342  000006                   .WORD   6
  852
  853  003344  012737  002001  001124   MOV     #BIT10!BIT0,$GDDAT      ;EXP CMD AND TCS.
  854  003352  017737  175770  001126   MOV     @IBS,$BDDAT     ;READ IBS - IS IT CLEAR?
  855  003360  023737  001124  001126   CMP     $GDDAT,$BDDAT
  856  003366  001401                   BEQ     TST12           ;;
  857

                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

  861
  862  003370  104003                   ERROR   3               ;/MODULE FAULT DETECTED:
  863                                                           ;IBS NOT CLEAR AFTER IBC

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
                                        ;;**************************************************************
  866                                   ;*TEST 12        #TEST THAT TON (BIT05) AND TKR SET AND CLEAR
  867                                   ;;**************************************************************
  868
  869  003372  000004            TST12: SCOPE
  870
  871  003374  005077  175746           CLR     @IBS            ;CLEAR THE CSR.
  872  003400  052777  000040  175740   BIS     #BIT5,@IBS      ;SET TON.
  873  003406  012737  001040  001124   MOV     #BIT5!BIT9,$GDDAT ;EXPECT ONLY TON AND TKR TO SET.
  874  003414  017737  175726  001126   MOV     @IBS,$BDDAT     ;READ CSR.
  875  003422  023737  001124  001126   CMP     $GDDAT,$BDDAT   ;DID THEY BOTH SET?
  876  003430  001402                   BEQ     1$              ;YES - CONTINUE.
  877
  878

                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

  882
  883  003432  104003                   ERROR   3               ;/MODULE FAULT DETECTED:
  884                                                           ;ERROR IN SETTING TON BIT.

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
  887  003434  000412                   BR      TST13           ;;
  888
  889  003436  042777  000040  175702  1$:    BIC     #BIT5,@IBS      ;WHEN TON CLEARED, TKR SHOULD CLEAR.
  890  003444  005037  001124           CLR     $GDDAT          ;EXPECT ZERO CSR.
  891  003450  017737  175672  001126   MOV     @IBS,$BDDAT     ;DID IT CLEAR?
  892  003456  001401                   BEQ     TST13           ;;
  893
```

```
                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

  897
  898  003460  104003                   ERROR   3               ;/MODULE FAULT DETECTED:
  899                                                           ;CSR FAILED TO CLEAR.

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS

  902
  903                                 ;;**********************************************************************
  904                                 ;*TEST 13       *MAKE SURE WE CAN SET AND CLEAR BIT06 (IE)
  905                                 ;;**********************************************************************
  906  003462  000004                 TST13:  SCOPE
  907
  908                                                           ;/PR
  909  003464  012746  000340                 MOV     #340,-(SP)      ;/SET CPU PRIORITY ON RETURN
  910  003470  012746  003476                 MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
  911  003474  000002                         RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
  912  003476                          64$:                           ;/IN STATUS REG.)
  913
  914  003476  005077  175644                 CLR     @IBS            ;CLEAR CSR.
  915
  916  003502  052777  000100  175636         BIS     #BIT6,@IBS      ;SET IE.
  917  003510  012737  000100  001124         MOV     #BIT6,$GDDAT    ;EXPECT ONLY BIT 6 TO SET.
  918  003516  017737  175624  001126         MOV     @IBS,$BDDAT     ;READ IBS.
  919  003524  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;DID IE SET?
  920  003532  001402                         BEQ     1$              ;YES - CONTINUE.
  921

                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

  925
  926  003534  104003                   ERROR   3               ;/MODULE FAULT DETECTED:
  927                                                           ;ERROR IN SETTING IE BIT.

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
  930  003536  000412                 BR      TST14           ;;
  931
  932  003540  005037  001124         1$:     CLR     $GDDAT          ;EXPECT ZERO CSR AFTER.
  933  003544  042777  000100  175574         BIC     #BIT6,@IBS      ;IE IS CLEARED.
  934  003552  017737  175570  001126         MOV     @IBS,$BDDAT     ;READ CSR - IS IT CLEAR?
  935  003560  001401                         BEQ     TST14           ;;
  936

                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

  940
  941  003562  104003                   ERROR   3               ;/MODULE FAULT DETECTED:
  942                                                           ;FAILED TO CLEAR CSR.

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS

  945
  946                                 ;;**********************************************************************
  947                                 ;*TEST 14       *TEST THAT BIT 7 (ACC) CAN BE SET AND CLEARED
  948                                 ;;**********************************************************************
```

```
003564  000004                            TST14:  SCOPE

003566  005077  175554                            CLR     @IBS                ;CLEAR CSR.
003572  052777  000200  175546                    BIS     #BIT7,@IBS          ;SET ACC.
003600  012737  000200  001124                    MOV     #BIT7,$GDDAT        ;EXPECT ONLY ACC TO SET.
003606  017737  175534  001126                    MOV     @IBS,$BDDAT         ;READ IBS.
003614  023737  001124  001126                    CMP     $GDDAT,$BDDAT       ;DID ACC SET?
003622  001402                                    BEQ     1$                  ;YES - CONTINUE.

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

003624  104003                                    ERROR   3                   ;/MODULE FAULT DETECTED:
                                                                              ;FAILURE IN SETTING BIT 7 (ACC).

                                                  ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$
003626  000412                                    BR      TST15               ;;

003630  042777  000200  175510    1$:             BIC     #BIT7,@IBS          ;TRY CLEARING ACC.
003636  005037  001124                            CLR     $GDDAT              ;EXPECT ZERO CSR.
003642  017737  175500  001126                    MOV     @IBS,$BDDAT         ;READ IBS, IS IT CLEAR?
003650  001401                                    BEQ     TST15               ;;

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

003652  104003                                    ERROR   3                   ;/MODULE FAULT DETECTED:
                                                                              ;IBS FAILED TO CLEAR.

                                                  ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$




                                          ;;***********************************************************
                                          ;*TEST 15         *TEST THAT IBD BIT 0 CAN BET SET + CLEARED
                                          ;;***********************************************************
003654  000004                            TST15:  SCOPE

                                                                              ;/MACRO BDT
003656  012777  000060  175462                    MOV     #BIT4!BIT5,@IBS     ;/SET TON AND LON.
003664  012737  000001  001124                    MOV     #BIT0,$GDDAT        ;/WE'RE GONNA TEST BIT 0.
003672  013777  001124  175450                    MOV     $GDDAT,@IBD         ;/SET THE BIT.

003700  117737  175444  001126                    MOVB    @IBD,$BDDAT         ;/READ THE IBD.
003706  123737  001124  001126                    CMPB    $GDDAT,$BDDAT       ;/DID IT GET THRU OK?
003714  001402                                    BEQ     1$                  ;/YES - CONTINUE.


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

```

```
1003  003716  104004                          ERROR   4               ;/MODULE FAULT DETECTED:
1004                                                                  ;/ERROR IN SETTING IBD BIT 0.

                                              ;;SSSSSSSSSS↑↑↑ ERROR ↑↑↑SSSSSSSSSS
1007  003720  000412                          BR      TST16           ;;
1008  003722  005037  001124            1$:   CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
1009  003726  042777  000001  175414        BIC     #BIT0,@IBD      ;/BIT 0 IS CLEARED.
1010  003734  117737  175410  001126        MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
1011  003742  001401                        BEQ     TST16           ;;
1012

                                              ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

1016
1017  003744  104004                          ERROR   4               ;/MODULE FAULT DETECTED:
1018                                                                  ;FAILED TO CLEAR IBD.

                                              ;;SSSSSSSSSS↑↑↑ ERROR ↑↑↑SSSSSSSSSS

1021
1022
1023                                  ;;********************************************************************
1024                                  ;*TEST 16       *TEST THAT IBD BIT 1 CAN BET SET + CLEARED
1025                                  ;;********************************************************************
1026  003746  000004                  TST16:  SCOPE
1027
1028                                                                  ;/MACRO BDT
1029  003750  012777  000060  175370        MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
1030  003756  012737  000002  001124        MOV     #BIT1,$GDDAT    ;/WE'RE GONNA TEST BIT 1.
1031  003764  013777  001124  175356        MOV     $GDDAT,@IBD     ;/SET THE BIT.
1032
1033  003772  117737  175352  001126        MOVB    @IBD,$BDDAT     ;/READ THE IBD.
1034  004000  123737  001124  001126        CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
1035  004006  001402                        BEQ     1$              ;/YES - CONTINUE.
1036
1037

                                              ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

1041
1042  004010  104004                          ERROR   4               ;/MODULE FAULT DETECTED:
1043                                                                  ;/ERROR IN SETTING IBD BIT 1.

                                              ;;SSSSSSSSSS↑↑↑ ERROR ↑↑↑SSSSSSSSSS
1046  004012  000412                          BR      TST17           ;;
1047  004014  005037  001124            1$:   CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
1048  004020  042777  000002  175322        BIC     #BIT1,@IBD      ;/BIT 1 IS CLEARED.
1049  004026  117737  175316  001126        MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
1050  004034  001401                        BEQ     TST17           ;;
1051

                                              ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

1055
1056  004036  104004                          ERROR   4               ;/MODULE FAULT DETECTED:
```

```
                                                        ;FAILED TO CLEAR IBD.

                                        ;;SSSSSSSSSS↑↑↑ ERROR ↑↑↑SSSSSSSSSS

                                ;;**********************************************************************
                                ;*TEST 17        *TEST THAT IBD BIT 2 CAN BET SET + CLEARED
                                ;;**********************************************************************
004040  000004          TST17:  SCOPE
                                                        ;/MACRO BDT
004042  012777  000060  175276          MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
004050  012737  000004  001124          MOV     #BIT2,$GDDAT    ;/WE'RE GONNA TEST BIT 2.
004056  013777  001124  175264          MOV     $GDDAT,@IBD     ;/SET THE BIT.

004064  117737  175260  001126          MOVB    @IBD,$BDDAT     ;/READ THE IBD.
004072  123737  001124  001126          CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
004100  001402                          BEQ     1$              ;/YES - CONTINUE.


                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

004102  104004                        ERROR   4               ;/MODULE FAULT DETECTED:
                                                                ;/ERROR IN SETTING IBD BIT 2.

                                        ;;SSSSSSSSSS↑↑↑ ERROR ↑↑↑SSSSSSSSSS
004104  000412                        BR      TST20           ;;
004106  005037  001124          1$:     CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
004112  042777  000004  175230          BIC     #BIT2,@IBD      ;/BIT 2 IS CLEARED.
004120  117737  175224  001126          MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
004126  001401                          BEQ     TST20           ;;


                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

004130  104004                        ERROR   4               ;/MODULE FAULT DETECTED:
                                                                ;FAILED TO CLEAR IBD.

                                        ;;SSSSSSSSSS↑↑↑ ERROR ↑↑↑SSSSSSSSSS


                                ;;**********************************************************************
                                ;*TEST 20        *TEST THAT IBD BIT 3 CAN BET SET + CLEARED
                                ;;**********************************************************************
004132  000004          TST20:  SCOPE
                                                        ;/MACRO BDT
004134  012777  000060  175204          MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
004142  012737  000010  001124          MOV     #BIT3,$GDDAT    ;/WE'RE GONNA TEST BIT 3.
004150  013777  001124  175172          MOV     $GDDAT,@IBD     ;/SET THE BIT.
```

```
1111  004156  117737  175166  001126          MOVB    @IBD,$BDDAT     ;/READ THE IBD.
1112  004164  123737  001124  001126          CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
1113  004172  001402                          BEQ     1$              ;/YES - CONTINUE.
1114
1115

                                              ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

1119
1120  004174  104004                          ERROR   4               ;/MODULE FAULT DETECTED:
1121                                                                  ;/ERROR IN SETTING IBD BIT 3.

                                              ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
1124  004176  000412                          BR      TST21           ;;
1125  004200  005037  001124          1$:     CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
1126  004204  042777  000010  175136          BIC     #BIT3,@IBD      ;/BIT 3 IS CLEARED.
1127  004212  117737  175132  001126          MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
1128  004220  001401                          BEQ     TST21           ;;
1129

                                              ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

1133
1134  004222  104004                          ERROR   4               ;/MODULE FAULT DETECTED:
1135                                                                  ;FAILED TO CLEAR IBD.

                                              ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS

1138
1139
1140                                          ;;=====================================================================
1141                                          ;*TEST 21       *TEST THAT IBD BIT 4 CAN BET SET + CLEARED
1142                                          ;;=====================================================================
1143  004224  000004                  TST21:  SCOPE
1144
1145                                                                  ;/MACRO BDT
1146  004226  012777  000060  175112          MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
1147  004234  012737  000020  001124          MOV     #BIT4,$GDDAT    ;/WE'RE GONNA TEST BIT 4.
1148  004242  013777  001124  175100          MOV     $GDDAT,@IBD     ;/SET THE BIT.
1149
1150  004250  117737  175074  001126          MOVB    @IBD,$BDDAT     ;/READ THE IBD.
1151  004256  123737  001124  001126          CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
1152  004264  001402                          BEQ     1$              ;/YES - CONTINUE.
1153
1154

                                              ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

1158
1159  004266  104004                          ERROR   4               ;/MODULE FAULT DETECTED:
1160                                                                  ;/ERROR IN SETTING IBD BIT 4.

                                              ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
1163  004270  000412                          BR      TST22           ;;
1164  004272  005037  001124          1$:     CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
```

```
1165  004276  042777  000020  175044          BIC     #BIT4,@IBD      ;/BIT 4 IS CLEARED.
1166  004304  117737  175040  001126          MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
1167  004312  001401                          BEQ     TST22           ;;
1168

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

1172
1173  004314  104004                          ERROR   4               ;/MODULE FAULT DETECTED:
1174                                                                  ;FAILED TO CLEAR IBD.

                                              ;;$$$$$$$$$$!!! ERROR !!!$$$$$$$$$$

1177
1178
1179                                  ;;######################################################################
1180                                  ;*TEST 22       #TEST THAT IBD BIT 5 CAN BET SET + CLEARED
1181                                  ;;######################################################################
1182  004316  000004                  TST22:  SCOPE
1183
1184                                                                  ;/MACRO BDT
1185  004320  012777  000060  175020          MOV     #BIT4!BIT5,@IBS ;/SET ION AND LON.
1186  004326  012737  000040  001124          MOV     #BIT5,$GDDAT    ;/WE'RE GONNA TEST BIT 5.
1187  004334  013777  001124  175006          MOV     $GDDAT,@IBD     ;/SET THE BIT.
1188
1189  004342  117737  175002  001126          MOVB    @IBD,$BDDAT     ;/READ THE IBD.
1190  004350  123737  001124  001126          CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
1191  004356  001402                          BEQ     1$              ;/YES - CONTINUE.
1192
1193

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

1197
1198  004360  104004                          ERROR   4               ;/MODULE FAULT DETECTED:
1199                                                                  ;/ERROR IN SETTING IBD BIT 5.

                                              ;;$$$$$$$$$$!!! ERROR !!!$$$$$$$$$$
1202  004362  000412                          BR      TST23           ;;
1203  004364  005037  001124          1$:     CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
1204  004370  042777  000040  174752          BIC     #BIT5,@IBD      ;/BIT 5 IS CLEARED.
1205  004376  117737  174746  001126          MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
1206  004404  001401                          BEQ     TST23           ;;
1207

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

1211
1212  004406  104004                          ERROR   4               ;/MODULE FAULT DETECTED:
1213                                                                  ;FAILED TO CLEAR IBD.

                                              ;;$$$$$$$$$$!!! ERROR !!!$$$$$$$$$$

1216
1217
1218                                  ;;######################################################################
```

```
1219                                    ;*TEST 23       *TEST THAT IBD BIT 6 CAN BET SET + CLEARED
1220                                    ;;*****************************************************************
1221  004410  000004                    TST23:  SCOPE
1222
1223                                                                    ;/MACRO BDT
1224  004412  012777  000060  174726            MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
1225  004420  012737  000100  001124            MOV     #BIT6,$GDDAT    ;/WE'RE GONNA TEST BIT 6.
1226  004426  013777  001124  174714            MOV     $GDDAT,@IBD     ;/SET THE BIT.
1227
1228  004434  117737  174710  001126            MOVB    @IBD,$BDDAT     ;/READ THE IBD.
1229  004442  123737  001124  001126            CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
1230  004450  001402                            BEQ     1$              ;/YES - CONTINUE.
1231
1232

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

1236
1237  004452  104004                            ERROR   4               ;/MODULE FAULT DETECTED:
1238                                                                    ;/ERROR IN SETTING IBD BIT 6.

                                        ;;$$$$$$$$$$!!! ERROR !!!$$$$$$$$$$
1241  004454  000412                            BR      TST24           ;;
1242  004456  005037  001124            1$:     CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
1243  004462  042777  000100  174660            BIC     #BIT6,@IBD      ;/BIT 6 IS CLEARED.
1244  004470  117737  174654  001126            MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
1245  004476  001401                            BEQ     TST24           ;;
1246

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

1250
1251  004500  104004                            ERROR   4               ;/MODULE FAULT DETECTED:
1252                                                                    ;FAILED TO CLEAR IBD.

                                        ;;$$$$$$$$$$!!! ERROR !!!$$$$$$$$$$

1255
1256
1257                                    ;;*****************************************************************
1258                                    ;*TEST 24       *TEST THAT IBD BIT 7 CAN BET SET + CLEARED
1259                                    ;;*****************************************************************
1260  004502  000004                    TST24:  SCOPE
1261
1262                                                                    ;/MACRO BDT
1263  004504  012777  000060  174634            MOV     #BIT4!BIT5,@IBS ;/SET TON AND LON.
1264  004512  012737  000200  001124            MOV     #BIT7,$GDDAT    ;/WE'RE GONNA TEST BIT 7.
1265  004520  013777  001124  174622            MOV     $GDDAT,@IBD     ;/SET THE BIT.
1266
1267  004526  117737  174616  001126            MOVB    @IBD,$BDDAT     ;/READ THE IBD.
1268  004534  123737  001124  001126            CMPB    $GDDAT,$BDDAT   ;/DID IT GET THRU OK?
1269  004542  001402                            BEQ     1$              ;/YES - CONTINUE.
1270
1271
```

```
                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

1275
1276  004544  104004                    ERROR   4               ;/MODULE FAULT DETECTED:
1277                                                            ;/ERROR IN SETTING IBD BIT 7.

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
1280  004546  000412                    BR      TST2S           ;;
1281  004550  005037  001124      1$:   CLR     $GDDAT          ;/EXPECT ZERO IBD WHEN
1282  004554  042777  000200  174566    BIC     #BIT7,@IBD      ;/BIT 7 IS CLEARED.
1283  004562  117737  174562  001126    MOVB    @IBD,$BDDAT     ;/READ IBD, IS IT CLEAR?
1284  004570  001401                    BEQ     TST2S           ;;
1285

                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

1289
1290  004572  104004                    ERROR   4               ;/MODULE FAULT DETECTED:
1291                                                            ;FAILED TO CLEAR IBD.

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
1294
1295
1296                            ;;**************************************************************
1297                            ;*TEST 25       *TEST THAT NO DATA GETS XFERRED, IF NOT ENABLED
1298                            ;;**************************************************************
1299  004574  000004            TST25:  SCOPE
1300
1301  004576  005077  174544            CLR     @IBS            ;CLEAR CSR
1302  004602  112777  000252  174540    MOVB    #252,@IBD       ;TRY XFERRING DATA
1303  004610  005037  001124            CLR     $GDDAT          ;NO DATA SHOULD XFERR
1304  004614  117737  174530  001126    MOVB    @IBD,$BDDAT     ;READ BUFFER REG.
1305  004622  001401                    BEQ     TST26           ;;
1306

                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

1310
1311  004624  104002                    ERROR   2               ;/MODULE FAULT DETECTED:
1312                                                            ;DATA WAS XFERRED THROUGH IBD
1313                                                            ;EVEN THOUGH TON AND LON CLEARED.
1314                                                            ;SIGNAL "ENB XFER L" PROBABLY
1315                                                            ;STUCK LOW.

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
1318
1319                            ;;**************************************************************
1320                            ;*TEST 26       *TEST IBD BITS DAC, AND DAV
1321                            ;;**************************************************************
1322  004626  000004            TST26:  SCOPE
1323
1324  004630  005077  174512            CLR     @IBS            ;CLEAR CSR.
1325  004634  005077  174510            CLR     @IBD            ;CLEAR DATA REG.
1326  004640  032777  000400  174502    BIT     #BIT8,@IBD      ;IS DAC SET?
```

```
                                        BNE     4$              ;YES (GOOD)
                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS
                                        ERROR   2               ;/MODULE FAULT DETECTED:
                                                                ;DAC NOT SET.
                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
                                        BR      TST27           ;;
                                4$:
                                        BIS     #BIT5!BIT4!BIT7,@IBS    ;SET TON AND LON
                                        MOV     #252,@IBD       ;PUT DATA IN IBD.
                                        MOV     @IBD,$BDOAT     ;READ IBD.
                                        BIT     #BIT9,$BDOAT    ;DID DAV SET?
                                        BNE     1$              ;YES - CONTINUE.
                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS
                                        ERROR   2               ;/MODULE FAULT DETECTED:
                                                                ;DAV FAILED TO SET.
                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
                                        BR      TST27           ;;
                                1$:     MOV     #BIT4!BIT5,@IBS ;CLEAR ACC.
                                2$:     TSTB    @IBD            ;READ LOW BYTE OF IBD.
                                        BIT     #BIT8,@IBD      ;DID DAC CLEAR?
                                        BEQ     3$              ;YES - CONTINUE.
                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS
                                        ERROR   2               ;/MODULE FAULT DETECTED:
                                                                ;DAC FAILED TO CLEAR.
                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
                                        BR      TST27           ;;
                                3$:     BIT     #BIT9,@IBD      ;DID DAV CLEAR?
                                        BEQ     TST27           ;;
                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS
                                        ERROR   2               ;/MODULE FAULT DETECTED:
                                                                ;DAV FAILED TO CLEAR.
                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
```

```
                                                                ;/MACRO -SIGC-

                                        ;;*********************************************************
                                        ;*TEST 27        *TEST THAT REN SETS WHEN REM SETS, ALSO TEST CLEAR
                                        ;;*********************************************************
 004752  000004                         TST27:  SCOPE

 004754  005077  174366                         CLR     @IBS            ;/CLEAR CSR.
 004760  052777  000004  174360                 BIS     #BIT2,@IBS      ;/SET REM, SHOULD SET REN.
 004766  032777  010000  174354                 BIT     #BIT12,@IBD     ;/DID REN SET?
 004774  001011                                 BNE     1$              ;/YES - LETS TRY CLEARING IT.
 004776  052777  000004  174342                 BIS     #BIT2,@IBS      ;/SET REM, MEMORY
                                                                        ;/REFRESH COULD HAVE
                                                                        ;/INTERRUPTED US.
 005004  032777  010000  174336                 BIT     #BIT12,@IBD     ;/DID REN SET THIS TIME?
 005012  001002                                 BNE     1$


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

 005014  104002                                 ERROR   2               ;/MODULE FAULT DETECTED:
                                                                        ;/REN FILED TO SET WHEN REM SET.

                                                ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$
 005016  000410                                 BR      TST30           ;;
 005020  042777  000004  174320 1$:             BIC     #BIT2,@IBS      ;/CLEAR REM, SHOULD CLEAR REN.
 005026  032777  010000  174314                 BIT     #BIT12,@IBD     ;/DID REN CLEAR?
 005034  001401                                 BEQ     TST30           ;;


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

 005036  104002                                 ERROR   2               ;/MODULE FAULT DETECTED:
                                                                        ;/REN FAILED TO CLEAR.

                                                ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$


                                                                        ;/MACRO -SIGC-

                                        ;;*********************************************************
                                        ;*TEST 30        *TEST THAT IFC SETS WHEN IBS SETS, ALSO TEST CLEAR
                                        ;;*********************************************************
 005040  000004                         TST30:  SCOPE

 005042  005077  174300                         CLR     @IBS            ;/CLEAR CSR.
 005046  052777  000010  174272                 BIS     #BIT3,@IBS      ;/SET IBS, SHOULD SET IFC.
 005054  032777  020000  174266                 BIT     #BIT13,@IBD     ;/DID IFC SET?
 005062  001011                                 BNE     1$              ;/YES - LETS TRY CLEARING IT.
 005064  052777  000010  174254                 BIS     #BIT3,@IBS      ;/SET IBS, MEMORY
                                                                        ;/REFRESH COULD HAVE
                                                                        ;/INTERRUPTED US.
```

```
1435  005072  032777  020000  174250          BIT     #BIT13,@IBD     ;/DID IFC SET THIS TIME?
1436  005100  001002                          BNE     1$
1437

                                              ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

1441
1442  005102  104002                          ERROR   2               ;/MODULE FAULT DETECTED:
1443                                                                  ;/IFC FILED TO SET WHEN IBS SET.

                                              ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
1446  005104  000411                          BR      TST31           ;;
1447
1448  005106  032777  000010  174232  1$:     BIT     #BIT3,@IBS      ;/WAIT FOR IBS TO CLEAR.
1449  005114  001374                          BNE     1$
1450
1451  005116  032777  020000  174224          BIT     #BIT13,@IBD     ;/IBS CLEAR, DID IFC CLEAR?
1452  005124  001401                          BEQ     TST31           ;;
1453

                                              ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

1457
1458  005126  104002                          ERROR   2               ;/MODULE FAULT DETECTED:
1459                                                                  ;/IFC FAILED TO CLEAR.

                                              ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
1462
1463
1464                                                                  ;/MACRO -SIGC-
1465
1466                                          ;;=====================================================================
1467                                          ;*TEST 31       *TEST THAT ATN SETS WHEN TCS SETS, ALSO TEST CLEAR
1468                                          ;;=====================================================================
1469  005130  000004                  TST31:  SCOPE
1470
1471  005132  005077  174210                  CLR     @IBS            ;/CLEAR CSR.
1472  005136  052777  000001  174202          BIS     #BIT0,@IBS      ;/SET TCS, SHOULD SET ATN.
1473  005144  032777  040000  174176          BIT     #BIT14,@IBD     ;/DID ATN SET?
1474  005152  001011                          BNE     1$              ;/YES - LETS TRY CLEARING IT.
1475  005154  052777  000001  174164          BIS     #BIT0,@IBS      ;/SET TCS, MEMORY
1476                                                                  ;/REFRESH COULD HAVE
1477                                                                  ;/INTERRUPTED US.
1478  005162  032777  040000  174160          BIT     #BIT14,@IBD     ;/DID ATN SET THIS TIME?
1479  005170  001002                          BNE     1$
1480

                                              ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

1484
1485  005172  104002                          ERROR   2               ;/MODULE FAULT DETECTED:
1486                                                                  ;/ATN FILED TO SET WHEN TCS SET.

                                              ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
```

```
1489  005174  000410                          BR      TST32           ;;
1490  005176  042777  000001  174142  1$:     BIC     #BIT0,@IBS      ;/CLEAR TCS, SHOULD CLEAR ATN.
1491  005204  032777  040000  174136          BIT     #BIT14,@IBD     ;/DID ATN CLEAR?
1492  005212  001401                          BEQ     TST32           ;;
1493

                                              ;;SSSSSSSSSS>>> ERROR <<<SSSSSSSSSS

1497
1498  005214  104002                          ERROR   2               ;/MODULE FAULT DETECTED:
1499                                                                  ;/ATN FAILED TO CLEAR.

                                              ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS

1502
1503
1504                                                                  ;/MACRO -SIGC-
1505
1506                                  ;;**************************************************************
1507                                  ;*TEST 32       *TEST THAT EOI SETS WHEN EOP SETS, ALSO TEST CLEAR
1508                                  ;;**************************************************************
1509  005216  000004                  TST32:  SCOPE
1510
1511  005220  005077  174122                  CLR     @IBS            ;/CLEAR CSR.
1512  005224  052777  000002  174114          BIS     #BIT1,@IBS      ;/SET EOP, SHOULD SET EOI.
1513  005232  032777  100000  174110          BIT     #BIT15,@IBD     ;/DID EOI SET?
1514  005240  001011                          BNE     1$              ;/YES - LETS TRY CLEARING IT.
1515  005242  052777  000002  174076          BIS     #BIT1,@IBS      ;/SET EOP, MEMORY
1516                                                                  ;/REFRESH COULD HAVE
1517                                                                  ;/INTERRUPTED US.
1518  005250  032777  100000  174072          BIT     #BIT15,@IBD     ;/DID EOI SET THIS TIME?
1519  005256  001002                          BNE     1$
1520

                                              ;;SSSSSSSSSS>>> ERROR <<<SSSSSSSSSS

1524
1525  005260  104002                          ERROR   2               ;/MODULE FAULT DETECTED:
1526                                                                  ;/EOI FILED TO SET WHEN EOP SET.

                                              ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
1529  005262  000410                          BR      TST33           ;;
1530  005264  042777  000002  174054  1$:     BIC     #BIT1,@IBS      ;/CLEAR EOP, SHOULD CLEAR EOI.
1531  005272  032777  100000  174050          BIT     #BIT15,@IBD     ;/DID EOI CLEAR?
1532  005300  001401                          BEQ     TST33           ;;
1533

                                              ;;SSSSSSSSSS>>> ERROR <<<SSSSSSSSSS

1537
1538  005302  104002                          ERROR   2               ;/MODULE FAULT DETECTED:
1539                                                                  ;/EOI FAILED TO CLEAR.

                                              ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS

1542
```

```
1543
1544                                    ;;*****************************************************************
1545                                    ;*TEST 33       *TEST THAT RFD SET WHEN CSR CLEAR,CLEAR WHEN ACC SET
1546                                    ;;*****************************************************************
1547   005304  000004                   TST33:  SCOPE
1548
1549   005306  005077  174034                   CLR     @IBS            ;CLEAR CSR.
1550   005312  032777  002000  174030           BIT     #BIT10,@IBD     ;DID RFD SET?
1551   005320  001002                           BNE     1$              ;YES CONTINUE.
1552
                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS
1556
1557   005322  104002                   ERROR   2                       ;/MODULE FAULT DETECTED:
1558                                                                    ;RFD FAILED TO SET.
                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
1561   005324  000410                   BR      TST34                   ;;
1562
1563   005326  052777  000200  174012  1$:      BIS     #BIT7,@IBS      ;NOW SET ACC,RFD SHOULD CLEAR.
1564   005334  032777  002000  174006           BIT     #BIT10,@IBD     ;DID IT CLEAR?
1565   005342  001401                           BEQ     TST34           ;;
1566
                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS
1570
1571   005344  104002                   ERROR   2                       ;/MODULE FAULT DETECTED:
1572                                                                    ;RFD FAILED TO CLEAR.
                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
1575
1576
1577                                    ;;*****************************************************************
1578                                    ;*TEST 34       *TEST THAT WE CAN GENERATE AN ER2
1579                                    ;;*****************************************************************
1580   005346  000004                   TST34:  SCOPE
1581
1582   005350  005077  173772                   CLR     @IBS            ;CLEAR THE STATUS REG.
1583   005354  052777  000041  173764           BIS     #BIT5!BIT0,@IBS ;SET TON; THIS SHOULD CAUSE AN
1584                                                                    ;ERROR SENCE NO LISTENERS ARE ON
1585   005362  105077  173762                   CLRB    @IBD            ;AND WE SENT DATA TO THE BUS.
1586                                                                    ;BUS.
1587   005366  032777  040000  173752           BIT     #BIT14,@IBS     ;DID ER2 SET?
1588   005374  001001                           BNE     TST35           ;;
1589
                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS
1593
1594   005376  104001                   ERROR   1                       ;/MODULE FAULT DETECTED:
1595                                                                    ;ER2 FAILED TO SET.
```

```
                                        ;;SSSSSSSSSS↑↑↑ ERROR ↑↑↑SSSSSSSSSS

1598
1599                                    ;;*********************************************************************
1600                                    ;*TEST 35       *TEST THAT BUS INIT CLEARS ACC,TON,LON,REM,EIP,TCS
1601                                    ;;*********************************************************************
1602  005400  000004                   TST35:  SCOPE
1603  005402  012737  000005  001160           MOV     #5,$TIMES       ;;DO 5 ITERATIONS
1604
1605  005410  012777  000367  173730           MOV     #367,@IBS       ;SET ACC,TON,LON,REM,EOP, AND TCS.
1606  005416  000005                           RESET                   ;ISSUE SYS INIT.
1607  005420  105777  173722                   TSTB    @IBS            ;DID THEY ALL CLEAR?
1608  005424  001401                           BEQ     TST36           ;;
1609

                                        ;;SSSSSSSSSS>>> ERROR <<<SSSSSSSSSS

1613
1614  005426  104001                           ERROR   1               ;/MODULE FAULT DETECTED:
1615                                                                   ;BUS INIT FAILED TO CLEAR CSR.
1616

                                        ;;SSSSSSSSSS↑↑↑ ERROR ↑↑↑SSSSSSSSSS

1619
1620                                    ;;*********************************************************************
1621                                    ;*TEST 36       *TEST IBC CLEARS ACC,TON,LON,REM AND EOP
1622                                    ;;*********************************************************************
1623  005430  000004                   TST36:  SCOPE
1624  005432  012737  000005  001160           MOV     #5,$TIMES       ;;DO 5 ITERATIONS
1625
1626  005440  012777  000266  173700           MOV     #266,@IBS       ;SET ACC,TONLON,REM, AND EOP.
1627  005446  052777  000010  173672           BIS     #BIT3,@IBS      ;SET IBC, THIS SHOULD CLEAR ABOVE BITS.
1628  005454  032777  000010  173664  1$:      BIT     #BIT3,@IBS      ;WAIT TILL IBS CLEARS
1629  005462  001374                           BNE     1$              ;
1630  005464  032777  000266  173654           BIT     #266,@IBS       ;DID THEY CLEAR?
1631  005472  001401                           BEQ     TST37           ;;
1632

                                        ;;SSSSSSSSSS>>> ERROR <<<SSSSSSSSSS

1636
1637  005474  104001                           ERROR   1               ;/MODULE FAULT DETECTED:
1638                                                                   ;ACC,TON,LON,REM, AND/OR EOP
1639                                                                   ;FAILED TO CLEAR ON IBC

                                        ;;SSSSSSSSSS↑↑↑ ERROR ↑↑↑SSSSSSSSSS

1642
1643                                    ;;*********************************************************************
1644                                    ;*TEST 37       *TEST THAT BUS INIT INDIRECTLY CLEARS IBD
1645                                    ;;*********************************************************************
1646  005476  000004                   TST37:  SCOPE
1647  005500  012737  000005  001160           MOV     #5,$TIMES       ;;DO 5 ITERATIONS
1648
1649  005506  012777  000260  173632           MOV     #BIT7!BIT5!BIT4,@IBS ;SET ACC,TON, AND LON.
1650  005514  012777  000377  173626           MOV     #377,@IBD       ;LOAD IBD
```

```
1651  005522  000005                        RESET                   ;ISSUE SYS INIT.
1652  005524  105777  173620                TSTB    @IBD            ;DID IT CLEAR?
1653  005530  001401                        BEQ     TST40           ;;
1654

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

1658
1659  005532  104002                        ERROR   2               ;/MODULE FAULT DETECTED:
1660                                                                ;FAILED TO CLEAR LOW BYTE OF IBD ON
1661                                                                ;SYSTEM INIT.

                                            ;;$$$$$$$$$$!!! ERROR !!!$$$$$$$$$$

1664
1665                                .SBTTL
1666                                .SBTTL          INTERRUPT TESTS
1667                                .SBTTL
1668
1669                                ;;******************************************************************
1670                                ;*TEST 40        *TEST THAT CMD CAN GENERATE AN INTERRUPT B
1671                                ;;******************************************************************
1672  005534  000004                TST40:  SCOPE
1673
1674  005536  005077  173604                MOV     @IBS            ;CLEAR THE CSR.
1675  005542  012777  000200  173636        MOV     #200,@PRC
1676  005550  012777  005606  173604        MOV     #1$,@VECTC      ;SET UP INTERRUPT VECTOR
1677  005556  052777  000101  173562        BIS     #BIT0!BIT6,@IBS ;SET TCS, SHOULD CAUSE
1678                                                                ;/PR
1679  005564  012746  000000                MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
1680  005570  012746  005576                MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
1681  005574  000002                        RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
1682  005576                        64$:                            ;/IN STATUS REG.)
1683  005576  000240                        NOP                     ;CMD TO SET AND GIVE US AN
1684  005600  000240                        NOP                     ;INTERRUPT.
1685

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

168$
169.  005602  104001                        ERROR   1               ;/MODULE FAULT DETECTED:
1691                                                                ;CMD FAILED TO GENERATE AN INTERRUPT.

                                            ;;$$$$$$$$$$!!! ERROR !!!$$$$$$$$$$
1694  005604  000402                        BR      2$
1695  005606                        1$:
1696  005606  062706  000004                ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
1697  005612  005077  173530        2$:     CLR     @IBS            ;CLEAR INTERRUPT
1698                                                                ;/-RESV-
1699  005616  013777  001406  173536        MOV     PRC,@VECTC ;/RESTORE VECTOR FOR
1700  005624  012777  004700  173554        MOV     #4700,@PRC      ;/ILLEGAL INTRO.
1701
1702                                ;;******************************************************************
1703                                ;*TEST 41        *TEST THAT TKR AND LNR CAN GENERATE INTERRUPTS
1704                                ;;******************************************************************
```

```
1705  005632  000004                          TST41:  SCOPE
1706  005634  012777  000200  173544                  MOV     #200,@PRC
1707  005642  012777  005726  173512                  MOV     #1$,@VECTC      ;SET UP INTERRUPT VECTOR FOR TKR INTERRUPT
1708  005650  012777  000060  173470                  MOV     #BIT4!BIT5,@IBS ;SET TON AND LON
1709  005656  052777  000100  173462                  BIS     #BIT6,@IBS      ;ALLOW INTERRUPT
1710                                                                          ;/PR
1711  005664  012746  000000                          MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
1712  005670  012746  005676                          MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
1713  005674  000002                                  RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
1714  005676                                  64$:                            ;/IN STATUS REG.)
1715  005676  000240                                  NOP
1716  005700  000240                                  NOP
1717

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

1721
1722  005702  104001                                  ERROR   1               ;/MODULE FAULT DETECTED:
1723                                                                          ;FAILED TO GENERATE A TKR INTERRUPT.

                                                 ;;$$$$$$$$$$!!! ERROR !!!$$$$$$$$$$
1726  005704  005077  173436                          CLR     @IBS            ;CLR CSR
1727                                                                          ;/-RESV-
1728  005710  013777  001406  173444                  MOV     PRC,@VECTC ;/RESTORE VECTOR FOR
1729  005716  012777  004700  173462                  MOV     #4700,@PRC      ;/ILLEGAL INTRO.
1730  005724  000443                                  BR      TST42           ;;
1731
1732  005726                                  1$:
1733  005726  062706  000004                          ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
1734                                                                          ;/-RESV-
1735  005732  013777  001406  173422                  MOV     PRC,@VECTC ;/RESTORE VECTOR FOR
1736  005740  012777  004700  173440                  MOV     #4700,@PRC      ;/ILLEGAL INTRO.
1737  005746  012777  000200  173434                  MOV     #200,@PRD
1738  005754  012777  006010  173402                  MOV     #2$,@VECTD      ;SET UP FOR LNR INTERRUPT.
1739                                                                          ;/PR
1740  005762  012746  000000                          MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
1741  005766  012746  005774                          MOV     #65$,-(SP)      ;/SHOW RETURN ADDRESS
1742  005772  000002                                  RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
1743  005774                                  65$:                            ;/IN STATUS REG.)
1744  005774  105277  173350                          INCB    @IBD            ;SEND DATA - CLRS TKR SETS LNR
1745                                                                          ;FOR INTERRUPT  .
1746  006000  000240                                  NOP
1747  006002  000240                                  NOP
1748

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

1752
1753  006004  104001                                  ERROR   1               ;/MODULE FAULT DETECTED:
1754                                                                          ;FAILED TO GENERATE LNR INTERRUPT

                                                 ;;$$$$$$$$$$!!! ERROR !!!$$$$$$$$$$
1757  006006  000402                                  BR      3$
1758
```

```
1759  006010                          2$:
1760  006010  062706  000004                  ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
1761  006014  005077  173326          3$:     CLR     @IBS            ;CLEAR THE STATUS REG.
1762                                                                    ;/-RESV-
1763  006020  013777  001410  173336          MOV     PRD,@VECTD ;/RESTORE VECTOR FOR
1764  006026  012777  004700  173354          MOV     #4700,@PRD      ;/ILLEGAL INTRO.
1765
1766                          ;;**********************************************************************
1767                          ;*TEST 42        *TEST THAT ER2 CAN GENERATE AN INTERRUPT
1768                          ;;**********************************************************************
1769  006034  000004          TST42:  SCOPE
1770
1771  006036  005077  173304          CLR     @IBS            ;START WITH CSR CLEAR
1772  006042  012777  000200  173332          MOV     #200,@PRA
1773  006050  012777  006126  173300          MOV     #1$,@VECTA      ;SET UP INTERRUPT VECTOR
1774                                                                  ;/PR
1775  006056  012746  000200                  MOV     #200,-(SP)      ;/SET CPU PRIORITY ON RETURN
1776  006062  012746  006070                  MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
1777  006066  000002                          RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
1778  006070                          64$:                            ;/IN STATUS REG.)
1779  006070  052777  000140  173250          BIS     #BIT5!BIT6,@IBS ;SET TON - NO LISTNERS ON
1780  006076  105077  173246                  CLRB    @IBD            ;BUS BUT DATA PUT ON
1781  006102  000240                          NOP                     ;BUS - THEREFORE AN INTERRUPT
1782  006104  000240                          NOP                     ;SHOULD BE POSTED.
1783                                                                  ;/PR
1784  006106  012746  000000                  MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
1785  006112  012746  006120                  MOV     #65$,-(SP)      ;/SHOW RETURN ADDRESS
1786  006116  000002                          RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
1787  006120                          65$:                            ;/IN STATUS REG.)
1788  006120  000240                          NOP
1789
1790

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

1794
1795  006122  104001                          ERROR   1               ;/MODULE FAULT DETECTED:
1796                                                                  ;FAILED TO INTERRUPT ON ERROR2

                                        ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$
1799  006124  000402                          BR      2$
1800  006126                          1$:
1801  006126  062706  000004                  ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
1802  006132  005077  173210          2$:     CLR     @IBS            ;CLEAR CSR
1803                                                                  ;/-RESV-
1804  006136  013777  001402  173212          MOV     PRA,@VECTA ;/RESTORE VECTOR FOR
1805  006144  012777  004700  173230          MOV     #4700,@PRA      ;/ILLEGAL INTRO.
1806
1807                                          .SBTTL
1808                                          .SBTTL  SECOND MODULE TESTS
1809                                          .SBTTL
1810
1811                          ;;**********************************************************************
1812                          ;*TEST 43        *TEST THAT MODULE PASSES "BIAKI"
```

```
                        ;;*****************************************************************
006152  000004         TST43:  SCOPE

                        ;*WARNING! THIS TEST IS DESIGNED TO BE EXERCISED WITH A
                        ;*SECOUND MODULE (IBV-11) WITH SWITCH "ER1 INH" SET.
                        ;*ADDRESS OF THE SECOUND MODULE IS IN LOCATION "IBS2" VECTOR
                        ;*ADDRESS IS IN LOCATION "VECTA2". THE SECOUND IBV-11 SHOULD BE ELECTRICALLY SEC
                        ;*TO INHIBIT THE USE OF TESTING WITH A SECOUND MODULE, MAKE
                        ;*LOCATION "SCDW1" ZERO.
                        ;*

006154  005737  001254          TST     $CDW1           ;TESTING WITH
006160  001002          BNE     3$              ;SECOUND IBV11?
006162  000137  006764          JMP     EOP             ;NO-END PASS.
006166                  3$:

006166  005077  173154          CLR     @IBS            ;CLEAR CSR.
006172  005077  173170          CLR     @IBS2           ;CLEAR SECOUND MODULE.
006176  012777  000200  173212  MOV     #200,@PRC2
006204  012777  006242  173164  MOV     #1$,@VECTC2     ;SET UP VECTOR ADDR.
                                                        ;/PR
006212  012746  000000          MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
006216  012746  006224          MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
006222  000002          RTI                             ;/CAUSE A RETURN (PUTS NEW STATUS
006224                  64$:                            ;/IN STATUS REG.)
006224  012777  000140  173134  MOV     #BIT6!BIT5,@IBS2 ;SET INTR ENABLE AND TON ON SECOUND
006232  000240          NOP                             ;IBV - SHOULD CAUSE A TKR  INTERRUPT.
006234  000240          NOP

                        ;;SSSSSSSSS>>> ERROR <<<SSSSSSSSS

006236  104001          ERROR   1                       ;/MODULE FAULT DETECTED:
                                                        ;ASSUMING SECOUND MODULE IS GOOD,
                                                        ;MODULE (IBV-11) UNDER TEST FAILED
                                                        ;TO PASS @ BUSS SIGNAL "BTAKI"

                        ;;SSSSSSSSS††† ERROR †††SSSSSSSSS
006240  000402          BR      2$
006242                  1$:
006242  062706  000004          ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
006246  005077  173114  2$:     CLR     @IBS2           ;CLEAR SECOUND MODULE
                                                        ;/-RESV-
006252  013777  001416  173116  MOV     PRC2,@VECTC2 ;/RESTORE VECTOR FOR
006260  012777  004700  173130  MOV     #4700,@PRC2     ;/ILLEGAL INTRO.

                        ;;*****************************************************************
                        ;*TEST 44        *TEST THAT SRQ CAN GENERATE AN INTERRUPT
                        ;;*****************************************************************
006266  000004         TST44:  SCOPE
                        ;*WARNING! THIS TEST IS DESIGNED TO BE EXERCISED WITH A
                        ;*SECOUND MODULE (IBV-11) WITH SWITCH "ER1 INH" SET.
```

```
                                            ;*ADDRESS OF THE SECOUND MODULE IS IN LOCATION "IBS2" VECTOR
                                            ;*ADDRESS IS IN LOCATION "VECTA2". THE SECOUND IBV-11 SHOULD BE ELECTRICALLY SEC
                                            ;*TO INHIBIT THE USE OF TESTING WITH A SECOUND MODULE, MAKE
                                            ;*LOCATION "SCDW1" ZERO.
                                            ;*

006270  005077  173052                      CLR     @IBS            ;CLEAR CSRS.
006274  005077  173066                      CLR     @IBS2

006300  012777  000200  173076              MOV     #200,@PRB       ;SET UP INTERRUPT VECTOR.
006306  012777  006352  173044              MOV     #1$,@VECTB
                                                                    ;/PR
006314  012746  000000                      MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
006320  012746  006326                      MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
006324  000002                              RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
006326                              64$:                            ;/IN STATUS REG.)
006326  012777  000100  173012              MOV     #100,@IBS       ;ENABLE INTERRUPTS
006334  052777  100000  173024              BIS     #BIT15,@IBS2    ;SETTING SRQ IN THE "CDW" MODULE
                                                                    ;WILL PUT SRQ ON THE IB BUS
                                                                    ;IS ER1 INH SW IS SET.
006342  000240                              NOP
006344  000240                              NOP


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

006346  104001                              ERROR   1               ;/MODULE FAULT DETECTED:
                                                                    ;SRQ FAILED TO GENERATE
                                                                    ;AN INTERRUPT

                                            ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$
006350  000402                              BR      2$

006352                              1$:
006352  062706  000004                      ADD     #4,SP           ;/ADD #4 TO STACK POINTER.

006356                              2$:
                                                                    ;/-RESV-
006356  013777  001404  172774              MOV     PRB,@VECTB ;/RESTORE VECTOR FOR
006364  012777  004700  173012              MOV     #4700,@PRB      ;/ILLEGAL INTRO.
006372  005077  172750                      CLR     @IBS            ;CLEAR CSRS
006376  005077  172764                      CLR     @IBS2

                                    ;;*********************************************************************
                                    ;*TEST 45       *TEST THAT ERROR1 IS GENERATED IF ATN IS ON THE IB BUS
                                    ;;*********************************************************************
006402  000004                      TST45:  SCOPE

                                            ;*WARNING! THIS TEST IS DESIGNED TO BE EXERCISED WITH A
                                            ;*SECOUND MODULE (IBV-11) WITH SWITCH "ER1 INH" SET.
                                            ;*ADDRESS OF THE SECOUND MODULE IS IN LOCATION "IBS2" VECTOR
                                            ;*ADDRESS IS IN LOCATION "VECTA2". THE SECOUND IBV-11 SHOULD BE ELECTRICALLY SEC
                                            ;*TO INHIBIT THE USE OF TESTING WITH A SECOUND MODULE, MAKE
```

```
                                        ;#LOCATION "$CDW1" ZERO.
                                        ;#
                                        CLR     @IBS2           ;CLR CSR OF 2ND MODULE.
006404  005077  172756                  INC     @IBS2           ;ASSERT ATN ON IB BUS
006410  005277  172752                                          ;ASSERTED ATN ON IBV UNDER TEST-
                                                                ;THIS SHOULD CAUSE AN ERROR 1
                                                                ;SENCE THE 2ND IBV HAS ATN SET.
006414  032777  020000  172724          BIT     #BIT13,@IBS     ;DID ERROR 1 SET?
006422  001001                          BNE     TST46           ;;

                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

006424  104001                          ERROR   1               ;/MODULE FAULT DETECTED:
                                                                ;FAILED TO GENERATE ERROR 1

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS

                                ;;**************************************************************
                                ;#TEST 46       #TEST THAT ERROR 1 IS GENERATED IF IFC IS PUT ON IB BUS BY SECOUND MODUL
                                ;;**************************************************************
000004                  TST46:  SCOPE
006430  012737  000005  001160          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
                                        ;#WARNING! THIS TEST IS DESIGNED TO BE EXERCISED WITH A
                                        ;#SECOUND MODULE (IBV-11) WITH SWITCH "ER1 INH" SET.
                                        ;#ADDRESS OF THE SECOUND MODULE IS IN LOCATION "IBS2" VECTOR
                                        ;#ADDRESS IS IN LOCATION "VECTA2". THE SECOUND IBV-11 SHOULD BE ELECTRICALLY SEC
                                        ;#TO INHIBIT THE USE OF TESTING WITH A SECOUND MODULE, MAKE
                                        ;#LOCATION "$CDW1" ZERO.
                                        ;#
006436  005077  172704                  CLR     @IBS            ;CLEAR CSR
006442  012777  000010  172716          MOV     #BIT3,@IBS2     ;ASSERT IFC FROM TESTOR
006450  032777  020000  172670          BIT     #BIT13,@IBS     ;DID ERROR 1 GET SET?
                                                                ;IF SO - NEXT TEST
006456  001010                          BNE     TST47           ;;
006460  012777  000010  172700          MOV     #BIT3,@IBS2     ;IF NOT WE'LL TRY AGAIN SENCE MEMORY
006466  032777  020000  172652          BIT     #BIT13,@IBS     ;REFRESH COULD HAVE GO IN THE WAY.
006474  001001                          BNE     TST47           ;;

                                        ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

006476  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.

                                        ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS

                                ;;**************************************************************
                                ;#TEST 47       #TEST THAT ERROR 1 IS GENERATED IF REN IS ON IB BUS
```

```
                        ;;*****************************************************************
006500  000004          TST47:  SCOPE
                        ;
                        ;*WARNING! THIS TEST IS DESIGNED TO BE EXERCISED WITH A
                        ;*SECOUND MODULE (IBV-11) WITH SWITCH "ERI INH" SET.
                        ;*ADDRESS OF THE SECOUND MODULE IS IN LOCATION "IBS2" VECTOR
                        ;*ADDRESS IS IN LOCATION "VECTA2". THE SECOUND IBV-11 SHOULD BE ELECTRICALLY SEC
                        ;*TO INHIBIT THE USE OF TESTING WITH A SECOUND MODULE, MAKE
                        ;*LOCATION "$CDW1" ZERO.
                        ;*
006502  005077  172640          CLR     @IBS            ;CLEAR CSRS.
006506  005077  172654          CLR     @IBS2
006512  052777  000004  172646  BIS     #BIT2,@IBS2     ;ASSERT REN ON IB BUS FROM 2ND
                                                        ;MODULE. 1ST IBV-11 SHOULD
006520  032777  020000  172620  BIT     #BIT13,@IBS     ;GENERATE AN ERROR 1;DID IT??
006526  001001                  BNE     TST50           ;;
                        ;;$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$
                        ;
006530  104001                  ERROR   1               ;/MODULE FAULT DETECTED:
                                                        ;FAILED TO GENERATE AN ERROR 1.
                        ;;$$$$$$$$$$!!! ERROR !!!$$$$$$$$$$
                        ;
                        ;;*****************************************************************
                        ;*TEST 50       *TEST THAT AN ERROR 1 CAN GENERATE AN INTERRUPT
                        ;;*****************************************************************
006532  000004          TST50:  SCOPE
                        ;
                        ;*WARNING! THIS TEST IS DESIGNED TO BE EXERCISED WITH A
                        ;*SECOUND MODULE (IBV-11) WITH SWITCH "ERI INH" SET.
                        ;*ADDRESS OF THE SECOUND MODULE IS IN LOCATION "IBS2" VECTOR
                        ;*ADDRESS IS IN LOCATION "VECTA2". THE SECOUND IBV-11 SHOULD BE ELECTRICALLY SEC
                        ;*TO INHIBIT THE USE OF TESTING WITH A SECOUND MODULE, MAKE
                        ;*LOCATION "$CDW1" ZERO.
                        ;*
006534  005077  172626          CLR     @IBS2           ;CLEAR CSRS.
006540  005077  172602          CLR     @IBS
                        ;
006544  012777  000200  172630  MOV     #200,@PRA
006552  012777  006616  172576  MOV     #1$,@VECTA      ;SET UP VECTOR ADDR.
                        ;
006560  052777  000100  172560  BIS     #BIT06,@IBS     ;SET INTERRUPT ENABLE
                                                        ;/PR
006566  012746  000000          MOV     #0,-(SP)        ;/SET CPU PRIORITY ON RETURN
006572  012746  006600          MOV     #64$,-(SP)      ;/SHOW RETURN ADDRESS
006576  000002                  RTI                     ;/CAUSE A RETURN (PUTS NEW STATUS
006600                  64$:                            ;/IN STATUS REG.)
006600  052777  000004  172560  BIS     #BIT2,@IBS2     ;GENERATE AN ERROR 1 AS PER LAST TEST.
006606  000240                  NOP
```

```
                                        NOP

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

                                        ERROR   1               ;/MODULE FAULT DETECTED:
                                                                ;ERROR 1 FAILED TO GENERATE AN INTR.

                                        ;;$$$$$$$$$$††† ERROR †††$$$$$$$$$$
                                        BR      2$
                                1$:
                                        ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
                                2$:
                                                                ;/-RESV-
                                        MOV     PRA,@VECTA ;/RESTORE VECTOR FOR
                                        MOV     #4700,@PRA      ;/ILLEGAL INTRO.
                                        CLR     @IBS2           ;CLEAR CSRS.
                                        CLR     @IBS

                                ;;**************************************************************
                                ;*TEST 51        *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.
                                        ;*
                                ;;**************************************************************
                                †ST51:  SCOPE

                                        MOV     #1$,$LPERR      ;SET ERROR LOOP.
                                        MOV     #0,$GDDAT       ;START PATTERN.
                                        CLR     $BDDAT

                                1$:     CLR     @IBS            ;CLEAR CSRS.
                                        CLR     @IBS2
                                        BIS     #BIT5!BIT0,@IBS ;SET TON AND TCS.
                                        BIS     #BIT4,@IBS2     ;SET LON ON KGM.
                                        MOV     $GDDAT,@IBD     ;SEND PATTERN.
                                        MOVB    @IBD2,$BDDAT    ;READ ATA FROM KGM.
                                        CMPB    $GDDAT,$BDDAT   ;DATA SENT = DATA RECEIVED?
                                        BEQ     2$              ;YES, CONTINUE

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

                                        ERROR   4               ;/MODULE FAULT DETECTED:
                                                                ;ERROR - BAD DATA PASSED BETWEEN
```

```
                                                        ;MODULE UNDER TEST AND KGM.

                                        ;;$$$$$$$$$$↑↑↑ ERROR ↑↑↑$$$$$$$$$$
006742  000407                          BR      TST2            ;;
006744  105237  001124        2$:       INCB    $GDDAT          ;CHANGE PATIERN.
006750  001347                          BNE     1$              ;IF NOT DONE, CONTINUE.
006752  005077  172370                  CLR     @IBS            ;CLEAR CSR'S
006756  005077  172404                  CLR     @IBS2

                                ;;**********************************************************************
                                ;*TEST 52        *TEMP END OF TESTS
                                ;;**********************************************************************
006762  000004                  TST52:  SCOPE

006764                          EOP:

                                        .SBTTL  SYSMAC ROUTINES:

                                .SBTTL  END OF PASS ROUTINE

                                ;;**********************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO RSTART

006764                          $EOP:
006764  000004                          SCOPE
006766  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
006772  005037  001160                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
006776  005237  001202                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
007002  042737  100000  001202          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
007010  005327                          DEC     (PC)+           ;;LOOP?
007012  000001          $EOPCT: .WORD   1
007014  003022                          BGT     $DOAGN          ;;YES
007016  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
007020  000001          $ENDCT: .WORD   1
007022  007012                          $EOPCT
007024  104401  007071                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
007030  013746  001202                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
007034  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
007036  104401  007066                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
007042  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
007046  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
007050  000005                          RESET                   ;;CLEAR THE WORLD
007052  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
007054  000240                          NOP                     ;;SAVE ROOM
007056  000240                          NOP                     ;;FOR
007060  000240                          NOP                     ;;ACT11
007062                  $DOAGN:
007062  000137                          JMP     @(PC)+          ;;RETURN
007064  002266          $RTNAD: .WORD   RSTART
```

```
2137  007066  377     377     000     $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
2138  007071  015     042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
2139  007076  050040  051501  020123
2140  007104  000043
2141                                                  ;/DELMA
2142
2143                                          ;/ROUTINE TO PROVIDE DELAYS IN INCREMENTS OF 25 US
2144                                          ;/
2145                                          ;/ CALL=        JSR     R5,DEL50
2146                                          ;/              .WORD   X       (# OF 25 US TO DELAY)
2147                                          ;/              ;RETURNS HERE
2148                                          ;/
2149
2150  007106  012500                  DEL50:  MOV     (5)+,R0         ;/GET # OF 25 US DELAYS
2151  007110  012701  000002          1$:     MOV     #2.,R1          ;/# FOR LOOP TO DO 50 US.
2152  007114  005301                  2$:     DEC     R1              ;/DEC IT
2153  007116  001376                          BNE     2$              ;/WAITED 25. TIMES?
2154  007120  005300                          DEC     R0              ;/DONE # OF 50 US DELAY DESIRED?
2155  007122  001372                          BNE     1$              ;/NO - NEXT ONE.
2156  007124  000205                          RTS     R5              ;/YES - EXIT.
2157
2158                          .SBTTL  ERROR HANDLER ROUTINE
2159
2160                          ;;**********************************************************************
2161                          ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
2162                          ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
2163                          ;*AND GO TO $ERRTYP ON ERROR
2164                          ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
2165                          ;*SW15=1         HALT ON ERROR
2166                          ;*SW13=1         INHIBIT ERROR TYPEOUTS
2167                          ;*SW10=1         BELL ON ERROR
2168                          ;*SW09=1         LOOP ON ERROR
2169                          ;*CALL
2170                          ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
2171
2172  007126                  $ERROR:
2173  007126  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
2174  007130  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
2175  007134  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
2176  007136  013777  001102  171776          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
2177  007144  032777  002000  171766          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
2178  007152  001402                          BEQ     1$              ;;NO - SKIP
2179  007154  104401  001164                  TYPE    ,$BELL          ;;RING BELL
2180  007160  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
2181  007164  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
2182  007170  162737  000002  001116          SUB     #2,$ERRPC
2183  007176  117737  171714  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
2184  007204  032777  020000  171726          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
2185  007212  001004                          BNE     20$             ;;SKIP TYPEOUTS
2186  007214  004737  007314                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
2187  007220  104401  001171                  TYPE    ,$CRLF
2188  007224                          20$:
2189  007224  122737  000001  001214          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
2190  007232  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
```

```
2191  007234  113737  001114  007246          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
2192  007242  004737  011460                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
2193  007246     000                  21$:    .BYTE   0
2194  007247     000                          .BYTE   0
2195  007250  000777                  22$:    BR      22$             ;;APT ERROR LOOP
2196  007252  005777  171662          2$:     TST     @SWR            ;;HALT ON ERROR
2197  007256  100002                          BPL     3$              ;;SKIP IF CONTINUE
2198  007260  000000                          HALT                    ;;HALT ON ERROR!
2199  007262  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
2200  007264  032777  001000  171646  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
2201  007272  001402                          BEQ     4$              ;;BR IF NO
2202  007274  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
2203  007300  005737  001162          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
2204  007304  001402                          BEQ     5$              ;;BR IF NONE
2205  007306  013716  001162                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
2206  007312                          5$:
2207  007312  000002                          RTI                     ;;RETURN
2208                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
2209
2210                                  ;;**************************************************************
2211                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
2212                                  ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
2213                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
2214
2215  007314                          $ERRTYP:
2216  007314  104401  001171                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
2217  007320  010046                          MOV     R0,-(SP)        ;;SAVE R0
2218  007322  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
2219  007324  153700  001114                  BISB    @#$ITEMB,R0
2220  007330  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
2221                                                                  ;;TYPE THE PC OF THE ERROR
2222  007332  013746  001116                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
2223                                                                  ;;ERROR ADDRESS
2224  007336  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2225  007340  000426                          BR      6$              ;;GET OUT
2226  007342  005300                  1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
2227  007344  006300                          ASL     R0              ;;       WORK FOR THE ERROR TABLE
2228  007346  006300                          ASL     R0
2229  007350  006300                          ASL     R0
2230  007352  062700  001256                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
2231  007356  012037  007366                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
2232  007362  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
2233  007364  104401                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
2234  007366  000000                  2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
2235  007370  104401  001171                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
2236  007374  012037  007404          3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
2237  007400  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
2238  007402  104401                          TYPE                    ;;TYPE THE "DATA HEADER"
2239  007404  000000                  4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
2240  007406  104401  001171                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
2241  007412  011000                  5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
2242  007414  001004                          BNE     7$              ;;GO TYPE THE DATA
2243  007416  012600                  6$:     MOV     (SP)+,R0        ;;RESTORE R0
2244  007420  104401  001171                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
```

```
2245  007424  000207                          RTS     PC              ;;RETURN
2246  007426                          7$:
2247  007426  013046                          MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
2248  007430  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2249  007432  005710                          TST     (R0)            ;;IS THERE ANOTHER NUMBER?
2250  007434  001770                          BEQ     6$              ;;BR IF NO
2251  007436  104401  007444                  TYPE    ,8$             ;;TYPE TWO(2) SPACES
2252  007442  000771                          BR      7$              ;;LOOP
2253  007444  020040      000       8$:       .ASCIZ  /  /            ;;TWO(2) SPACES
2254          007450                          .EVEN
2255                                  .SBTTL  SCOPE HANDLER ROUTINE
2256
2257                                  ;;**************************************************************
2258                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
2259                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
2260                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
2261                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
2262                                  ;*SW14=1         LOOP ON TEST
2263                                  ;*SW11=1         INHIBIT ITERATIONS
2264                                  ;*SW09=1         LOOP ON ERROR
2265                                  ;*SW08=1         LOOP ON TEST IN SWR<7:0>
2266                                  ;*CALL
2267                                  ;*      SCOPE           ;;SCOPE=IOT
2268
2269  007450                          $SCOPE:
2270  007450  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
2271  007452  104407                          CKSWR
2272  007454  032777  040000  171456  1$:      BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
2273  007462  001114                          BNE     $OVER           ;;YES IF SW14=1
2274                                  ;#####START OF CODE FOR THE XOR TESTER#####
2275  007464  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
2276                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
2277  007466  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
2278  007472  012737  007512  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
2279  007500  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
2280  007504  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
2281  007510  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
2282  007512  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
2283  007514  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
2284  007520  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
2285  007522                          6$:;#####END OF CODE FOR THE XOR TESTER#####
2286  007522  032777  000400  171410          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
2287  007530  001404                          BEQ     2$              ;;BR IF NO
2288  007532  127737  171402  001102          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
2289  007540  001465                          BEQ     $OVER           ;;BR IF YES
2290  007542  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
2291  007546  001421                          BEQ     3$              ;;BR IF NO
2292  007550  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
2293  007556  101015                          BHI     3$              ;;BR IF NO
2294  007560  032777  001000  171352          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
2295  007566  001404                          BEQ     4$              ;;BR IF NO
2296  007570  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
2297  007576  000446                          BR      $OVER
2298  007600  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
```

```
2299  007604  005037  001160                    CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
2300  007610  000415                            BR      1$              ;;ESCAPE TO THE NEXT TEST
2301  007612  032777  004000  171320  3$:       BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
2302  007620  001011                            BNE     1$              ;;BR IF YES
2303  007622  005737  001202                    TST     $PASS           ;;IF FIRST PASS OF PROGRAM
2304  007626  001406                            BEQ     1$              ;;      INHIBIT ITERATIONS
2305  007630  005237  001104                    INC     $ICNT           ;;INCREMENT ITERATION COUNT
2306  007634  023737  001160  001104            CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
2307  007642  002024                            BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
2308  007644  012737  000001  001104  1$:       MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
2309  007652  013737  007730  001160            MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
2310  007660  105237  001102          $SVLAD:   INCB    $TSTNM          ;;COUNT TEST NUMBERS
2311  007664  113737  001102  001200            MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
2312  007672  011637  001106                    MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
2313  007676  011637  001110                    MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
2314  007702  005037  001162                    CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
2315  007706  112737  000001  001115            MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
2316  007714  013777  001102  171220  $OVER:    MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
2317  007722  013716  001106                    MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
2318  007726  000002                            RTI                     ;;FIXES PS
2319  007730  003720                  $MXCNT:   2000.                   ;;MAX. NUMBER OF ITERATIONS
2320                                  .SBTTL  TTY INPUT ROUTINE
2321
2322                                  ;;**************************************************************
2323                                  .ENABL  LSB
2324
2325                                  ;;**************************************************************
2326                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
2327                                  ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
2328                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
2329                                  ;*WHEN OPERATING IN TTY FLAG MODE.
2330  007732  022737  000176  001140  $CKSWR:  CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
2331  007740  001074                            BNE     15$             ;;BRANCH IF NO
2332  007742  105777  171176                    TSTB    @$TKS           ;;CHAR THERE?
2333  007746  100071                            BPL     15$             ;;IF NO, DON'T WAIT AROUND
2334  007750  117746  171172                    MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
2335  007754  042716  177600                    BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
2336  007760  022726  000007                    CMP     #7,(SP)+        ;;IS IT A CONTROL G?
2337  007764  001062                            BNE     15$             ;;NO, RETURN TO USER
2338  007766  123727  001134  000001            CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
2339  007774  001456                            BEQ     15$             ;;BRANCH IF YES
2340
2341  007776  104401  010457          $GTSWR:   TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
2342  010002  104401  010464                    TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
2343  010006  013746  000176                    MOV     $SWREG,-(SP)    ;;SAVE SWREG FOR TYPEOUT
2344  010012  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2345  010014  104401  010475                    TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
2346  010020  005046                  19$:      CLR     -(SP)           ;;CLEAR COUNTER
2347  010022  005046                            CLR     -(SP)           ;;THE NEW SWR
2348  010024  105777  171114          7$:       TSTB    @$TKS           ;;CHAR THERE?
2349  010030  100375                            BPL     7$              ;;IF NOT TRY AGAIN
2350
2351  010032  117746  171110                    MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
2352  010036  042716  177600                    BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
```

```
010042 021627 000025          9$:     CMP     (SP),#25           ;;IS IT A CONTROL-U?
010046 001005                         BNE     10$                ;;BRANCH IF NOT
010050 104401 010452                  TYPE    ,$CNTLU            ;;YES, ECHO CONTROL-U (↑U)
010054 062706 000006          20$:    ADD     #6,SP              ;;IGNORE PREVIOUS INPUT
010060 000757                         BR      19$                ;;LET'S TRY IT AGAIN

010062 021627 000015          10$:    CMP     (SP),#15           ;;IS IT A <CR>?
010066 001022                         BNE     16$                ;;BRANCH IF NO
010070 005766 000004                  TST     4(SP)              ;;YES, IS IT THE FIRST CHAR?
010074 001403                         BEQ     11$                ;;BRANCH IF YES
010076 016677 000002 171034           MOV     2(SP),@SWR         ;;SAVE NEW SWR
010104 062706 000006          11$:    ADD     #6,SP              ;;CLEAR UP STACK
010110 104401 001171          14$:    TYPE    ,$CRLF             ;;ECHO <CR> AND <LF>
010114 123727 001135 000001           CMPB    $INTAG,#1          ;;RE-ENABLE TTY KBD INTERRUPTS?
010122 001003                         BNE     15$                ;;BRANCH IF NOT
010124 012777 000100 171012           MOV     #100,@$TKS         ;;RE-ENABLE TTY KBD INTERRUPTS
010132 000002                 15$:    RTI                        ;;RETURN
010134 004737 011372          16$:    JSR     PC,$TYPEC          ;;ECHO CHAR
010140 021627 000060                  CMP     (SP),#60           ;;CHAR < 0?
010144 002420                         BLT     18$                ;;BRANCH IF YES
010146 021627 000067                  CMP     (SP),#67           ;;CHAR > 7?
010152 003015                         BGT     18$                ;;BRANCH IF YES
010154 042726 000060                  BIC     #60,(SP)+          ;;STRIP-OFF ASCII
010160 005766 000002                  TST     2(SP)              ;;IS THIS THE FIRST CHAR
010164 001403                         BEQ     17$                ;;BRANCH IF YES
010166 006316                         ASL     (SP)               ;;NO, SHIFT PRESENT
010170 006316                         ASL     (SP)               ;;   CHAR OVER TO MAKE
010172 006316                         ASL     (SP)               ;;   ROOM FOR NEW ONE.
010174 005266 000002          17$:    INC     2(SP)              ;;KEEP COUNT OF CHAR
010200 056616 177776                  BIS     -2(SP),(SP)        ;;SET IN NEW CHAR
010204 000707                         BR      7$                 ;;GET THE NEXT ONE
010206 104401 001170          18$:    TYPE    ,$QUES             ;;TYPE ?<CR><LF>
010212 000720                         BR      20$                ;;SIMULATE CONTROL-U
                              .DSABL  LSB



                              ;;*************************************************************
                              ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                              ;*CALL:
                              ;*       RDCHR                      ;;INPUT A SINGLE CHARACTER FROM THE TTY
                              ;*       RETURN HERE                ;;CHARACTER IS ON THE STACK
                              ;*                                  ;;WITH PARITY BIT STRIPPED OFF
                              ;

010214 011646                 $RDCHR: MOV     (SP),-(SP)         ;;PUSH DOWN THE PC
010216 016666 000004 000002           MOV     4(SP),2(SP)        ;;SAVE THE PS
010224 105777 170714          1$:     TSTB    @$TKS              ;;WAIT FOR
010230 100375                         BPL     1$                 ;;A CHARACTER
010232 117766 170710 000004           MOVB    @$TKB,4(SP)        ;;READ THE TTY
010240 042766 177600 000004           BIC     #↑C<177>,4(SP)     ;;GET RID OF JUNK IF ANY
```

```
2407  010246  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
2408  010254  001013                          BNE     3$              ;;BRANCH IF NO
2409  010256  105777  170662          2$:     TSTB    @#TKS           ;;WAIT FOR A CHARACTER
2410  010262  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
2411  010264  117746  170656                  MOVB    @#TKB,-(SP)     ;;GET CHARACTER
2412  010270  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
2413  010274  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
2414  010300  001366                          BNE     2$              ;;IF NOT DISCARD IT
2415  010302  000750                          BR      1$              ;;YES, RESUME
2416  010304  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
2417  010312  002407                          BLT     4$              ;;BRANCH IF YES
2418  010314  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
2419  010322  003003                          BGT     4$              ;;BRANCH IF YES
2420  010324  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
2421  010332  000002                  4$:     RTI                     ;;GO BACK TO USER
2422                                          ;;**************************************************************
2423                                          ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
2424                                          ;*CALL:
2425                                          ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
2426                                          ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
2427                                          ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
2428
2429  010334  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
2430  010336  012703  010442          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
2431  010342  022703  010452          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
2432  010346  101405                          BLOS    4$              ;;BR IF YES
2433  010350  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
2434  010352  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
2435  010354  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
2436  010360  001003                          BNE     3$              ;;SKIP IF NOT
2437  010362  104401  001170          4$:     TYPE    ,$QUES          ;;TYPE A '?'
2438  010366  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
2439  010370  111337  010440          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
2440  010374  104401  010440                  TYPE    ,9$
2441  010400  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
2442  010404  001356                          BNE     2$              ;;LOOP IF NOT RETURN
2443  010406  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
2444  010412  104401  001172                  TYPE    ,$LF            ;;TYPE A LINE FEED
2445  010416  012603                          MOV     (SP)+,R3        ;;RESTORE R3
2446  010420  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
2447  010422  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
2448  010430  012766  010442  000004          MOV     #$TTYIN,4(SP)
2449  010436  000002                          RTI                     ;;RETURN
2450  010440     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
2451  010441     000                          .BYTE   0               ;;TERMINATOR
2452  010442  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
2453  010452  052536  005015     000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
2454  010457     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
2455  010464  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
2456  010472  020075     000
2457  010475     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
2458  010502  036440  000040
2459                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
2460
```

```
                                ;;*****************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT
010506  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
010512  116637  000001  010731          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
010520  112637  010733          MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
010524  062716  000002          ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
010530  000406                  BR      $TYPON
010532  112737  000001  010731  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
010540  112737  000006  010733          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
010546  112737  000005  010730  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
010554  010346                  MOV     R3,-(SP)        ;;SAVE R3
010556  010446                  MOV     R4,-(SP)        ;;SAVE R4
010560  010546                  MOV     R5,-(SP)        ;;SAVE R5
010562  113704  010733          MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
010566  005404                  NEG     R4
010570  062704  000006          ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
010574  110437  010732          MOVB    R4,$OMODE       ;;SAVE IT FOR USE
010600  113704  010731          MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
010604  016605  000012          MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
010610  005003                  CLR     R3              ;;CLEAR THE OUTPUT WORD
010612  006105          1$:     ROL     R5              ;;ROTATE MSB INTO "C"
010614  000404                  BR      3$              ;;GO DO MSB
010616  006105          2$:     ROL     R5              ;;FORM THIS DIGIT
010620  006105                  ROL     R5
010622  006105                  ROL     R5
010624  010503                  MOV     R5,R3
010626  006103          3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
010630  105337  010732          DECB    $OMODE          ;;TYPE THIS DIGIT?
010634  100016                  BPL     7$              ;;BR IF NO
010636  042703  177770          BIC     #177770,R3      ;;GET RID OF JUNK
010642  001002                  BNE     4$              ;;TEST FOR 0
010644  005704                  TST     R4              ;;SUPPRESS THIS 0?
010646  001403                  BEQ     5$              ;;BR IF YES
```

```
2515  010650  005204                    4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
2516  010652  052703  000060            5$:     BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
2517  010656  052703  000040                    BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
2518  010662  110337  010726                    MOVB    R3,8$           ;;SAVE FOR TYPING
2519  010666  104401  010726                    TYPE    ,8$             ;;GO TYPE THIS DIGIT
2520  010672  105337  010730            7$:     DECB    $OCNT           ;;COUNT BY 1
2521  010676  003347                            BGT     2$              ;;BR IF MORE TO DO
2522  010700  002402                            BLT     6$              ;;BR IF DONE
2523  010702  005204                            INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
2524  010704  000744                            BR      2$              ;;GO DO THE LAST DIGIT
2525  010706  012605                    6$:     MOV     (SP)+,R5        ;;RESTORE R5
2526  010710  012604                            MOV     (SP)+,R4        ;;RESTORE R4
2527  010712  012603                            MOV     (SP)+,R3        ;;RESTORE R3
2528  010714  016666  000002  000004            MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
2529  010722  012616                            MOV     (SP)+,(SP)
2530  010724  000002                            RTI                     ;;RETURN
2531  010726     000            8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
2532  010727     000                    .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
2533  010730     000            $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
2534  010731     000            $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
2535  010732  000000            $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                ;;*****************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                ;*REPLACED WITH SPACES.
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                ;*      TYPDS                   ;;GO TO THE ROUTINE

010734                          $TYPDS:
010734  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
010736  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
010740  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
010742  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
010744  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
010746  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
010752  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
010756  100004                          BPL     1$              ;;BR IF INPUT IS POS.
010760  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
010762  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
010770  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
010772  012703  011150                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
010776  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
011002  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
011004  016001  011140                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
011010  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
011012  002402                          BLT     4$              ;;BR IF DONE
011014  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
011016  000774                          BR      3$
011020  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
011022  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
011024  001002                          BNE     5$              ;;FALL THROUGH IF 0
011026  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
011030  100407                          BMI     7$              ;;BR IF YES
011032  106316                  5$:     ASLB    (SP)            ;;MSD?
011034  103003                          BCC     6$              ;;BR IF NO
011036  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
011044  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
011050  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
011054  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
011056  005720                          TST     (R0)+           ;;JUST INCREMENTING
011060  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
011064  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
011066  003002                          BGT     8$              ;;GO TO EXIT
011070  010502                          MOV     R5,R2           ;;GET THE LSD
011072  000764                          BR      6$              ;;GO CHANGE TO ASCII
011074  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
011076  100003                          BPL     9$              ;;BR IF NO
011100  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
011106  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
011110  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
```

```
                                        MOV     (SP)+,R3        ;;POP STACK INTO R3
                                        MOV     (SP)+,R2        ;;POP STACK INTO R2
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
011122  104401  011150                  TYPE    $DBLK           ;;NOW TYPE THE NUMBER
011126  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
011134  012616                          MOV     (SP)+,(SP)
011136  000002                          RTI                     ;;RETURN TO USER
011140  023420                  $DTBL:  10000.
011142  001750                          1000.
011144  000144                          100.
011146  000012                          10.
011150  000004                  $DBLK:  .BLKW   4
                                .SBTTL  TYPE ROUTINE

                                ;;*****************************************************************
                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*

000157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
                        BPL     1$              ;;BR IF YES
                        HALT                    ;;HALT HERE IF NO TERMINAL
                        BR      3$              ;;LEAVE
                1$:     MOV     R0,-(SP)        ;;SAVE R0
000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
000001  001214          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
                        BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
000100  001215          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
011230                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
011450                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
000000          61$:    .WORD   0               ;;MESSAGE ADDRESS
000040  001215  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
                        BNE     60$             ;;YES,SKIP TYPE OUT
                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
                60$:    MOV     (SP)+,R0        ;;RESTORE R0
000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
                        RTI                     ;;RETURN
000011                  CMPB    #HT,(SP)        ;;BRANCH IF <HT>
                        BEQ     8$
000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
```

```
2644  011272  001006                        BNE     5$
2645  011274  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
2646  011276  104401                        TYPE                    ;;TYPE A CR AND LF
2647  011300  001171                        $CRLF
2648  011302  105037  011436                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
2649  011306  000755                        BR      2$              ;;GET NEXT CHARACTER
2650  011310  004737  011372        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
2651  011314  123726  001156        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
2652  011320  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
2653  011322  013746  001154                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
2654                                                                ;;AND THE NULL CHAR.
2655  011326  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
2656  011332  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
2657  011334  004737  011372                JSR     PC,$TYPEC       ;;GO TYPE A NULL
2658  011340  105337  011436                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
2659  011344  000770                        BR      7$              ;;LOOP
2660
2661                                ;HORIZONTAL TAB PROCESSOR
2662
2663  011346  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
2664  011352  004737  011372        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
2665  011356  132737  000007  011436        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
2666  011364  001372                        BNE     9$              ;;TAB STOP
2667  011366  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
2668  011370  000724                        BR      2$              ;;GET NEXT CHARACTER
2669  011372  105777  167552        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
2670  011376  100375                        BPL     $TYPEC
2671  011400  116677  000002  167544        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
2672  011406  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
2673  011414  001003                        BNE     1$              ;;BRANCH IF NO
2674  011416  105037  011436                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
2675  011422  000406                        BR      $TYPEX          ;;EXIT
2676  011424  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
2677  011432  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
2678  011434  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
2679  011436  000000        $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
2680  011440  000207        $TYPEX: RTS     PC
2681
2682                                .SBTTL  APT COMMUNICATIONS ROUTINE
2683
2684                                ;;**************************************************************************
2685  011442  112737  000001  011706  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
2686  011450  112737  000001  011704  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
2687  011456  000403                        BR      $ATYC
2688  011460  112737  000001  011706  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
2689  011466                        $ATYC:
2690  011466  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
2691  011470  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2692  011472  105737  011704                TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
2693  011476  001450                        BEQ     5$              ;;IF NOT:  BR
2694  011500  122737  000001  001214        CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
2695  011506  001031                        BNE     3$              ;;IF NOT:  BR
2696  011510  132737  000100  001215        BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
2697  011516  001425                        BEQ     3$              ;;IF NOT:  BR
```

```
2698  011520  017600  000004              MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
2699  011524  062766  000002  000004      ADD     #2,4(SP)        ;;BUMP RETURN ADDR.
2700  011532  005737  001174        1$:   TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
2701  011536  001375                      BNE     1$              ;;IF NOT:  WAIT
2702  011540  010037  001210              MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
2703  011544  105720                2$:   TSTB    (R0)+           ;;FIND END OF MESSAGE
2704  011546  001376                      BNE     2$
2705  011550  163700  001210              SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
2706  011554  006200                      ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
2707  011556  010037  001212              MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
2708  011562  012737  000004  001174      MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
2709  011570  000413                      BR      5$
2710  011572  017637  000004  011616  3$:  MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
2711  011600  062766  000002  000004      ADD     #2,4(SP)        ;;BUMP RETURN ADDRESS
2712  011606  013746  177776              MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
2713  011612  004737  011160              JSR     PC,$TYPE        ;;CALL TYPE MACRO
2714  011616  000000                4$:   .WORD   0
2715  011620                        5$:
2716  011620  105737  011706       10$:   TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
2717  011624  001416                      BEQ     12$             ;;IF NOT:  BR
2718  011626  005737  001214              TST     $ENV            ;;RUNNING UNDER APT?
2719  011632  001413                      BEQ     12$             ;;IF NOT:  BR
2720  011634  005737  001174       11$:   TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
2721  011640  001375                      BNE     11$             ;;IF NOT:  WAIT
2722  011642  017637  000004  001176      MOV     @4(SP),$FATAL   ;;GET ERROR #
2723  011650  062766  000002  000004      ADD     #2,4(SP)        ;;BUMP RETURN ADDR.
2724  011656  005237  001174              INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
2725  011662  105037  011706       12$:   CLRB    $FFLG           ;;CLEAR FATAL FLAG
2726  011666  105037  011705              CLRB    $LFLG           ;;CLEAR LOG FLAG
2727  011672  105037  011704              CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
2728  011676  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
2729  011700  012600                      MOV     (SP)+,R0        ;;POP STACK INTO R0
2730  011702  000207                      RTS     PC              ;;RETURN
2731  011704     000          $MFLG:   .BYTE   0                  ;;MESSG. FLAG
2732  011705     000          $LFLG:   .BYTE   0                  ;;LOG FLAG
2733  011706     000          $FFLG:   .BYTE   0                  ;;FATAL FLAG
2734              011710                .EVEN
2735              000200          APTSIZE=200
2736              000001          APTENV=001
2737              000100          APTSPOOL=100
2738              000040          APTCSUP=040
2739                              .SBTTL  POWER DOWN AND UP ROUTINES
2740
2741                              ;;*****************************************************************
2742                              ;POWER DOWN ROUTINE
2743  011710  012737  012050  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
2744  011716  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
2745  011724  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
2746  011726  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2747  011730  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
2748  011732  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
2749  011734  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
2750  011736  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
2751  011740  017746  167174                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
```

```
011744  010637  012054                    MOV     SP,$SAVR6           ;;SAVE SP
011750  012737  011762  000024            MOV     #$PWRUP,@#PWRVEC    ;;SET UP VECTOR
011756  000000                            HALT
011760  000776                            BR      .-2                 ;;HANG UP

                                  ;;*****************************************************
                                  ;POWER UP ROUTINE
011762  012737  012050  000024  $PWRUP:   MOV     #$ILLUP,@#PWRVEC    ;;SET FOR FAST DOWN
011770  013706  012054                    MOV     $SAVR6,SP           ;;GET SP
011774  005037  012054                    CLR     $SAVR6              ;;WAIT LOOP FOR THE TTY
012000  005237  012054          1$:       INC     $SAVR6              ;;WAIT FOR THE INC
012004  001375                            BNE     1$                  ;;OF WORD
012006  012677  167126                    MOV     (SP)+,@SWR          ;;POP STACK INTO @SWR
012012  012605                            MOV     (SP)+,R5            ;;POP STACK INTO R5
012014  012604                            MOV     (SP)+,R4            ;;POP STACK INTO R4
012016  012603                            MOV     (SP)+,R3            ;;POP STACK INTO R3
012020  012602                            MOV     (SP)+,R2            ;;POP STACK INTO R2
012022  012601                            MOV     (SP)+,R1            ;;POP STACK INTO R1
012024  012600                            MOV     (SP)+,R0            ;;POP STACK INTO R0
012026  012737  011710  000024            MOV     #$PWRDN,@#PWRVEC    ;;SET UP THE POWER DOWN VECTOR
012034  012737  000340  000026            MOV     #340,@#PWRVEC+2     ;;PRIO:7
012042  104401                            TYPE                        ;REPORT THE POWER FAILURE
012044  012056                  $PWRMG:   .WORD   $POWER              ;;POWER FAIL MESSAGE POINTER
012046  000002                            RTI
012050  000000                  $ILLUP:   HALT                        ;;THE POWER UP SEQUENCE WAS STARTED
012052  000776                            BR      .-2                 ;; BEFORE THE POWER DOWN WAS COMPLETE
012054  000000                  $SAVR6:   0                           ;;PUT THE SP HERE
012056  005015  047520  042527  $POWER:   .ASCIZ  <15><12>"POWER"
012064  000122
                                          .EVEN

                                          ;*
                                          ;*THIS ROUTINE WILL PROTECT THE PROGRAM
                                          ;*FROM INTERRUPTS.
                                          ;*
                                          ;*THE TRAP CATCHER IS SET UP FOR
                                          ;*        .WORD   .+2
                                          ;*        .WORD   JSR     PC,R0
                                          ;*
                                          ;*ILLEGAL INTERRUPTS OR INTERRUPTS TO THE WRONG VECTOR
                                          ;*GOTO THE VECTOR AND PICK UP THE ".+2" AS AN ADDRESS
                                          ;*AND "4700" AS NEW STATUS.
                                          ;*THE .+2 AS A PC WILL CAUSE EXECUTION OF THE "JSR PC,R0" (AN ILLEGAL INSTR).
                                          ;*AND TRAP TO LOCATION "4". IN LOCATION 10 WE HAVE A
                                          ;*POINTER HERE. IF THIS CONDITION CAUSES A TRAP TO LOC. 4
                                          ;*WE WILL REPORT IT IN THE SAME MANNER THAT WE WOULD
                                          ;*REPORT ANY OTHER ERROR.

                                          ;*IF A BUSS ERROR TRAP DID OCCUR AND CAUSE A TRAP TO 4,
                                          ;*WE WILL HALT.

012066  011637  012132          IOTRD:    MOV     (6),TRTO            ;GET WHERE WE CAME TO.
012072  162737  000004  012132            SUB     #4,TRTO             ;FORM REAL ADDR.
```

```
2806  012100  023727  012132  001000          CMP     TRTO,#1000      ;DID TRAP COME FROM LESS THAN ADDR. 1000?
2807  012106  003402                          BLE     2$
2808
2809  012110  000000                  1$:     HALT                    ;NO!  MUST BE A BUSS ILLEGAL ADDR. TIME OUT.
2810                                                                  ; ADDRESS CONTAINED IN TRTO.
2811
2812  012112  000776                          BR      1$              ;DON'T ALLOW A CONTINUE.
2813  012114                          2$:
2814
2815  012114  016637  000004  012134          MOV     4(6),TRFRO      ;GET TRAPPED FROM ADDR.
2816
2817  012122  062706  000004                  ADD     #4,SP           ;/ADD #4 TO STACK POINTER.
2818
2819

                                              ;;SSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSS

2823
2824  012126  104007                          ERROR   7               ;/MODULE FAULT DETECTED:
2825                                                                  ;ERROR! ILLEGAL INTERRUPT
2826                                                                  ;OR INTERRUPT TO WRONG
2827                                                                  ;VECTOR - IF TEST NUMBER
2828                                                                  ;IS LESS THAN 10, ITS LIKELY
2829                                                                  ;(BUT NOT EXCLUSIVELY) TO BE A
2830                                                                  ;DEVICE OTHER THAN THE IBV-11
2831                                                                  ;TO BLAME.
2832                                                                  ;IF THE INTERRUPT OCCURRED
2833                                                                  ;DURING AN INTERRUPT TEST, I'D
2834                                                                  ;SUSPECT A PROBLEM WITH THE
2835                                                                  ;IBV-11.
2836                                                                  ;IF THE ADDRESS THE INTERRUPT
2837                                                                  ;VECTOR TO IS WITHIN THE RANGE
2838                                                                  ;OF VECTORS ASSIGNED TO THE IBV-11,
2839                                                                  ;THEN I'D SUSPECT THE IBV-11
2840                                                                  ;INTERRUPTED ILLEGALLY.
2841                                                                  ;IF THE ADDRESS THE INTERRUPT
2842                                                                  ;VECTORED TO IS OUTSIDE OF THE
2843                                                                  ;RANGE ASSIGNED TO THE IBV-11,
2844                                                                  ;I'D SUSPECT THAT THE
2845                                                                  ;IBV-11 PUT THE WRONG VECTOR ON
2846                                                                  ;THE BUSS DURING THE INTERRUPT
2847                                                                  ;PROCESS.
2848                                                                  ;FOR THIS ERROR - DON'T
2849                                                                  ;USE "LOOP ON ERROR" OPTION.
2850                                                                  ;ALSO EXPECT THE INTERRUPT TEST TO
2851                                                                  ;REPORT THAT THE IBV-11 DIDN'T
2852                                                                  ;INTERRUPT.
2853                                                                  ;FOLLOW RECOMMENDED PROCEDURE
2854                                                                  ;IN THE DOCUMENT (ON THIS DIAGNOSTIC)
2855                                                                  ;FOR LOOPING ON ERROR

                                              ;;SSSSSSSSSS††† ERROR †††SSSSSSSSSS
2858  012130  000002                          RTI
2859
```

```
012132  000000                  TRTO:   .WORD   0               ;ADDR THAT WE INTERRUPTED TO
012134  000000                  TRFRO:  .WORD   0               ;ADDR THAT WE INTERRUPTED FROM.

                                .SBTTL  TRAP DECODER

                                ;;**************************************************************
                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                ;*GO TO THAT ROUTINE.

012136  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
012140  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
012144  005740                          TST     -(R0)           ;;BACKUP BY 2
012146  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
012150  006300                          ASL     R0              ;;POSITION FOR INDEXING
012152  016000  012172                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
012156  000200                          RTS     R0              ;;GO TO ROUTINE

                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

012160  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
012162  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
012170  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                ;*BY THE "TRAP" INSTRUCTION.

                                ;       ROUTINE
                                ;       -------
012172  012160                  $TRPAD: .WORD   $TRAP2
012174  011160                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
012176  010532                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
012200  010506                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
012202  010546                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
012204  010734                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

012206  010002                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

012210  007732                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
012212  010214                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
012214  010334                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE

                                .SBTTL MESSAGES AND TABLES

012216  005007  044415  051502  EM1:    .ASCIZ<7><12><15>#IBS FUNCTION ERROR#
012224  043040  047125  052103
012232  047511  020116  051105
012240  047522  000122

012244  005007  044415  042102  EM2:    .ASCIZ<7><12><15>#IBD FUNCTION ERROR#
```

```
2914  012252  043040  047125  052103
2915  012260  047511  020116  051105
2916  012266  047522  000122
2917
2918  012272  005007  044415  051502  EM3:    .ASCIZ<7><12><15>#IBS DATA ERROR#
2919  012300  042040  052101  020101
2920  012306  051105  047522  000122
2921
2922  012314  005007  044415  042102  EM4:    .ASCIZ<7><12><15>#IBD DATA ERROR#
2923  012322  042040  052101  020101
2924  012330  051105  047522  000122
2925
2926  012336  100007  041111  027523  EM5:    .ASCIZ<7><200>#IBS/IBD ADDRESS ERROR#
2927  012344  041111  020104  042101
2928  012352  051104  051505  020123
2929  012360  051105  047522  000122
2930  012366  100007  041111  041527  EM6:    .ASCIZ  <7><200>#IBWC/IBCA DATA ERROR#
2931  012374  044457  041502  020101
2932  012402  040504  040524  042440
2933  012410  051122  051117     000
2934
2935  012415     007  044600  052116  EM7:    .ASCIZ  <7><200>#INTERRUPT ERROR#
2936  012422  051105  052522  052120
2937  012430  042440  051122  051117
2938  012436     000
2939
2940  012437     200  042524  052123  DH1:    .ASCIZ<CRLF>#TEST    ERRPC   IB ADR   IBS     IBD#
2941  012444  020040  020040  051105
2942  012452  050122  020103  020040
2943  012460  041111  040440  051104
2944  012466  020040  044440  051502
2945  012474  020040  020040  044440
2946  012502  042102     000
2947
2948  012505     200  042524  052123  DH3:    .ASCIZ<CRLF>#TEST    ERRPC   GOOD    BAD#
2949  012512  020040  020040  051105
2950  012520  050122  020103  020040
2951  012526  047507  042117  020040
2952  012534  020040  040502  000104
2953
2954  012542  052200  051505  020124  DH5:    .ASCIZ  <CRLF>#TEST    ERRPC   IB ADDR#
2955  012550  020040  042440  051122
2956  012556  041520  020040  044440
2957  012564  020102  042101  051104
2958  012572     000
2959
2960  012573     200  042524  052123  DH7:    .ASCIZ  <CRLF>#TEST    ERRPC   TO      FROM ADDR.#
2961  012600  020040  020040  051105
2962  012606  050122  020103  020040
2963  012614  047524  020040  020040
2964  012622  020040  051106  046517
2965  012630  040440  042104  027122
2966  012636     000
2967
```

```
2968            012640                                  .EVEN
2969
2970  012640    001200  001116  001346  DT1:    .WORD   $TESTN,$ERRPC,IBS
2971
2972  012646    000000                  IBSA:   .WORD   0
2973
2974  012650    000000  000000          IBDA:   .WORD   0,0
2975
2976  012654    001200  001116  001124  DT3:    .WORD   $TESTN,$ERRPC,$GDDAT,$BDDAT,0
2977  012662    001126  000000
2978
2979  012666    001200  001116  001346  DT5:    .WORD   $TESTN,$ERRPC,IBS,0
2980  012674    000000
2981
2982  012676    001200  001116  012132  DT7:    .WORD   $TESTN,$ERRPC,TRTO,TRFRO,0
2983  012704    012134  000000
2984
2985  012710    000000  000000          DF0:    .WORD   0,0
2986
2987            000001                          .END
```

```
ABASE = 160150          167#    255     296     376     377     378     379     384     385
ACDW1 = 000000          255     298
ACDW2 = 000000          255
ACPUOP= 000000          255     270
ADDW0 = 000000          255
ADDW1 = 000000          255
ADDW10= 000000          255
ADDW11= 000000          255
ADDW12= 000000          255
ADDW13= 000000          255
ADDW14= 000000          255
ADDW15= 000000          255
ADDW2 = 000000          255
ADDW3 = 000000          255
ADDW4 = 000000          255
ADDW5 = 000000          255
ADDW6 = 000000          255
ADDW7 = 000000          255
ADDW8 = 000000          255
ADDW9 = 000000          255
ADEVCT= 000000          255     261
ADEVM = 000000          255     297
AENV  = 000000          255     266
AENVM = 000000          255     267
AFATAL= 000000          255     258
AMADR1= 000000          255     283
AMADR2= 000000          255     287
AMADR3= 000000          255     290
AMADR4= 000000          255     293
AMAMS1= 000000          255     277
AMAMS2= 000000          255     285
AMAMS3= 000000          255     288
AMAMS4= 000000          255     291
AMSGAD= 000000          255     263
AMSGLG= 000000          255     264
AMSGTY= 000000          255     257
AMTYP1= 000000          255     278
AMTYP2= 000000          255     286
AMTYP3= 000000          255     289
AMTYP4= 000000          255     292
APASS = 000000          255     260
APRIOR= 000200          169#    255
APTCSU= 000040         2633    2738#
APTENV= 000001         2189    2626    2694    2736#
APTSIZ= 000200          446    2735#
APTSPO= 000100         2628    2696    2737#
ASWREG= 000000          255     268
ATESTN= 000000          255     259
AUNIT = 000000          255     262
AUSWR = 000000          255     269
AVECT1= 000640          168#    255     294     380     381     382     383     386     387     388     389     393     394
                        395     396     398     399     400     401
AVECT2= 000000          255     295
BIT0  = 000001          151#    722     723     738     853     991    1009    1472    1475    1490    1583    1677    2070
```

```
BIT00 = 000001          141#    151
BIT01 = 000002          140#    150
BIT02 = 000004          139#    149     794     795     810
BIT03 = 000010          138#    148     830     831     835
BIT04 = 000020          137#    147
BIT05 = 000040          136#    146
BIT06 = 000100          135#    145    2021
BIT07 = 000200          134#    144
BIT08 = 000400          133#    143    2286
BIT09 = 001000          132#    142    2200    2294
BIT1  = 000002          150#    758     759     774    1030    1048    1512    1515    1530
BIT10 = 002000          131#    723     853    1550    1564    2177
BIT11 = 004000          130#   2301
BIT12 = 010000          129#   1390    1395    1408
BIT13 = 020000          128#   1430    1435    1451    1929    1955    1959    1990    2184
BIT14 = 040000          127#   1473    1478    1491    1587    2272
BIT15 = 100000          126#   1513    1518    1531    1884
BIT2  = 000004          149#   1069    1087    1389    1392    1407    1988    2027
BIT3  = 000010          148#   1108    1126    1429    1432    1448    1627    1628    1954    1958
BIT4  = 000020          147#    990    1029    1068    1107    1146    1147    1165    1185    1224    1263    1339    1355
                       1649    1708    2071
BIT5  = 000040          146#    872     873     889     990    1029    1068    1107    1146    1185    1186    1204    1224
                       1263    1339    1355    1583    1649    1708    1779    1839    2070
BIT6  = 000100          145#    916     917     933    1225    1243    1677    1709    1779    1839
BIT7  = 000200          144#    952     953     968    1264    1282    1339    1563    1649
BIT8  = 000400          143#   1326    1357
BIT9  = 001000          142#    873    1342    1369
BPTVEC= 000014          158#
CKSWR = 104407         2173    2199    2270    2271    2902#
CR    = 000015            66#  2672    2682
CRLF  = 000200            67#   526    2643    2682    2940    2948    2954    2960
DDISP = 177570            73#   236     434
DEL50   007106           850   2150#
DF0     012710           323    330     337     344     351     358     364    2985#
DH1     012437           321    328    2940#
DH3     012505           335    342     356    2948#
DH5     012542           349   2954#
DH7     012573           362   2960#
DISPLA  001142           236#   434*    442*   2176*   2316*
DISPRE  000174            48#   442
DSWR  = 177570            72#   235     433
DT1     012640           322    329    2970#
DT3     012654           336    343     357    2976#
DT5     012666           350   2979#
DT7     012676           363   2982#
EMTVEC= 000030           161#   418*    419*
EM1     012216           320   2908#
EM2     012244           327   2913#
EM3     012272           334   2918#
EM4     012314           341   2922#
EM5     012336           348   2926#
EM6     012366           355   2930#
EM7     012415           361   2935#
EOP     006764          1827   2099#
```

K06


```
ERRVEC= 000004   154#   431    432*   443*   450*   451*   540    541*   547*   575*   594    595*   601*
                 633*   2277   2278*  2280*  2283*
GNS   = ****** U 525    2894   2895   2896   2897   2898   2900   2902   2903   2904
GTSWR = 104406   520    2900#
HT    = 000011    64#   2641   2682
IBCA    001354   379#   458*   459*   603    703
IBD     001350   377#   454*   455*   456    463    549    665    992*   994    1009*  1010   1031*  1033
                1048*  1049   1070*  1072   1087*  1088   1109*  1111   1126*  1127   1148*  1150   1165*
                1166   1187*  1189   1204*  1205   1226*  1228   1243*  1244   1265*  1267   1282*  1283
                1302*  1304   1325*  1326   1340*  1341   1356   1357   1369   1390   1395   1408   1430
                1435   1451   1473   1478   1491   1513   1518   1531   1550   1564   1585*  1650*  1652
                1744*  1780*  2072*
IBDA    012650   463*  2974#
IBD2    001370   385#   464*   465*   2073
IBS     001346   376#   453*   454    462    545    644    721*   722*   724    738*   740    757*   758*
                 760    774*   776    793*   794*   796    810*   812    829*   830*   832    835*   836
                 854    871*   872*   874    889*   891    914*   916*   918    933*   934    951*   952*
                 954    968*   970    990*   1029*  1068*  1107*  1146*  1185*  1224*  1263*  1301*  1324*
                1339*  1355*  1388*  1389*  1392*  1407*  1428*  1429*  1432*  1448   1471*  1472*  1475*
                1490*  1511*  1512*  1515*  1530*  1549*  1563*  1582*  1583*  1587   1605*  1607   1626*
                1627*  1628   1630   1649*  1674*  1677*  1697*  1708*  1709*  1726*  1761*  1771*  1779*
                1802*  1830*  1873*  1883*  1908*  1929   1953*  1955   1959   1986*  1990   2016*  2021*
                2048*  2068*  2070*  2091*  2970   2979
IBSA    012646   462*  2972#
IBS2    001366   384#   464    1831*  1839*  1856*  1874*  1884*  1909*  1924*  1925*  1954*  1958*  1987*
                1988*  2015*  2027*  2047*  2069*  2071*  2092*
IBWC    001352   378#   456*   457*   458    598    689
IOTRD   012066    46    450    2803#
IOTVEC= 000020   159#   416*   417*
LF    = 000012    65#   2676   2682
PC    =%000007    85#   2117*  2120*  2130*  2135   2186*  2192*  2245*  2374*  2631*  2650*  2657*  2664*
                2678*  2680*  2713*  2730*
PIRQ  = 177772    71#
PIRQVE= 000240   165#
PRA     001402   393#   479*   480*   481    497    498*   1772*  1804   1805*  2018*  2045   2046*
PRA2    001412   398#   487*   488*   489
PRB     001404   394#   481*   482*   483    500    501*   1876*  1906   1907*
PRB2    001414   399#   489*   490*   491
PRC     001406   395#   483*   484*   485    503    504*   1675*  1699   1700*  1706*  1728   1729*  1735
                1736*
PRC2    001416   400#   491*   492*   493    1832*  1858   1859*
PRD     001410   396#   485*   486*   506    507*   1737*  1763   1764*
PRD2    001420   401#   493*   494*
PR0   = 000000    88#
PR1   = 000040    89#
PR2   = 000100    90#
PR3   = 000140    91#
PR4   = 000200    92#
PR5   = 000240    93#
PR6   = 000300    94#
PR7   = 000340    95#
PS    = 177776    68#    69
PSW   = 177776    69#
PWRVEC= 000024   160#   422*   423*   2743*  2744*  2753*  2759*  2771*  2772*
```

```
RDCHR = 104410          2433    2903#
RDLIN = 104411          2904#
RESVEC= 000010           155#
RSTART  002266           495#  2136
R0     =%000000            76#  2127*  2130   2150*  2154*  2217   2218*  2219*  2226*  2227*  2228*  2229*  2230*
                         2231    2236   2241*  2243*  2247   2249   2549   2559*  2563   2579   2580   2593*  2624
                         2625*   2630   2635   2638*  2690   2698*  2702   2703   2705*  2706*  2707   2729*  2745
                         2770*   2871   2872*  2873   2874*  2875*  2876*  2877*
R1     =%000001            77#  2151*  2152*  2550   2563*  2564   2568   2592*  2691   2728*  2746   2769*
R2     =%000002            78#  2551   2562*  2566*  2569   2576*  2577*  2578   2583*  2591*  2747   2768*
R3     =%000003            79#  2429   2430*  2431   2434*  2435   2439   2441   2443*  2445*  2492   2501*  2507*
                         2508*   2511*  2516*  2517*  2518   2527*  2552   2560*  2561*  2575*  2578*  2587*  2588*
                         2590*   2748   2767*
R4     =%000004            80#  2493   2495*  2496*  2497*  2498   2499*  2513   2515*  2523*  2526*  2749   2766*
R5     =%000005            81#   850*  2156*  2494   2500*  2502*  2504*  2505*  2506*  2507   2525*  2553   2555*
                         2557*   2564*  2568*  2583   2589*  2750   2765*
R6     =%000006            82#   410*   411*   412
R7     =%000007            83#
SP     =%000006            84#   414*   431*   439*   443    540*   553*   565*   575    577*   578*   594*   607*
                          621*   633    909*   910*  1679*  1680*  1696*  1711*  1712*  1733*  1740*  1741*  1760*
                         1775*  1776*  1784*  1785*  1801*  1835*  1836*  1855*  1879*  1880*  1902*  2023*  2024*
                         2042*  2124*  2181   2202*  2205*  2217*  2222*  2243   2247*  2277*  2280   2282   2283
                         2312   2313   2317*  2334*  2335*  2336   2343*  2346*  2347*  2351*  2352*  2356   2359*
                         2363   2365   2367   2368*  2375   2377   2379*  2380   2382*  2383*  2384*  2385*  2386*
                         2401*  2402*  2405*  2406*  2407   2411*  2412*  2413   2416   2418   2420*  2429*  2434
                         2445   2446*  2447*  2448*  2484*  2485   2486   2487*  2492*  2493*  2494*  2500   2525
                         2526   2527   2528*  2529*  2549*  2550*  2551*  2552*  2553*  2554*  2555   2558*  2571
                         2573*  2575   2585   2587   2589   2590   2591   2592   2593   2595*  2596*  2624*  2625
                         2635*  2637   2638   2639*  2641   2643   2645   2651   2653*  2655*  2663*  2667   2671
                         2672   2676   2690*  2691*  2698   2699*  2710   2711*  2712*  2722   2723*  2728   2729
                         2745*  2746*  2747*  2748*  2749*  2750*  2751*  2752   2760*  2764   2765   2766   2767
                         2768   2769   2770   2817*  2871*  2872   2882*  2883*
STACK  = 001100           59#   414
START   001422            54    407#
STKLMT= 177774            70#
SWR     001140          2358#   412    433*   435    441*   448*   518    2177   2184   2196   2200   2272   2286
                         2288    2294   2301   2330   2367*  2751   2764*
SWREG   000176            49#   441    518    2330   2343
SW0    = 000001          123#
SW00   = 000001          113#   123
SW01   = 000002          112#   122
SW02   = 000004          111#   121
SW03   = 000010          110#   120
SW04   = 000020          109#   119
SW05   = 000040          108#   118
SW06   = 000100          107#   117
SW07   = 000200          106#   116
SW08   = 000400          105#   115
SW09   = 001000          104#   114
SW1    = 000002          122#
SW10   = 002000          103#
SW11   = 004000          102#
SW12   = 010000          101#
SW13   = 020000          100#
```

```
SW14   = 040000            99#
SW15   = 100000            98#
SW2    = 000004           121#
SW3    = 000010           120#
SW4    = 000020           119#
SW5    = 000040           118#
SW6    = 000100           117#
SW7    = 000200           116#
SW8    = 000400           115#
SW9    = 001000           114#
TBITVE= 000014           156#
TKVEC  = 000060           163#
TPVEC  = 000064           164#
TRAPVE= 000034           162#   420*   421*
TRFRO   012134          2815*  2861#  2982
TRTO    012132          2803*  2804*  2806   2860#  2982
TRTVEC= 000014           157#
TST1    002444           532#
TST10   003156           772    777    791#
TST11   003246           808    813    827#
TST12   003372           848    856    869#
TST13   003462           887    892    906#
TST14   003564           930    935    949#
TST15   003654           966    971    987#
TST16   003746          1007   1011   1026#
TST17   004040          1046   1050   1065#
TST2    002570           591#
TST20   004132          1085   1089   1104#
TST21   004224          1124   1128   1143#
TST22   004316          1163   1167   1182#
TST23   004410          1202   1206   1221#
TST24   004502          1241   1245   1260#
TST25   004574          1280   1284   1299#
TST26   004626          1305   1322#
TST27   004752          1337   1353   1368   1370   1386#
TST3    002654           638#
TST30   005040          1406   1409   1426#
TST31   005130          1446   1452   1469#
TST32   005216          1489   1492   1509#
TST33   005304          1529   1532   1547#
TST34   005346          1561   1565   1580#
TST35   005400          1588   1602#
TST36   005430          1608   1623#
TST37   005476          1631   1646#
TST4    002704           645    659#
TST40   005534          1653   1672#
TST41   005632          1705#
TST42   006034          1730   1769#
TST43   006152          1814#
TST44   006266          1864#
TST45   006402          1914#
TST46   006426          1930   1944#
TST47   006500          1957   1960   1976#
TST5    002734           666    685#
```

```
TST50   006532          1991    2005#
TST51   006546          2062#
TST52   006762          2086    2097#
TST6    002776           701     704    719#
TST7    003066           736     741    755#
TYPOS = 104405          2125    2898#
TYPE  = 104401           512    2123   2126   2179   2187   2216   2233   2235   2238   2240   2244   2251   2341
                        2342    2345   2358   2369   2388   2437   2440   2444   2519   2594   2646   2773   2894#
TYPOC = 104402          2224    2248   2344   2895#
TYPON = 104404          2897#
TYPOS = 104403          2896#
VECTA   001356           380#    460*   461*   466    479    497*  1773*  1804*  2019*  2045*
VECTA2  001372           386#    472    487
VECTB   001360           381#    466*   467*   468    500*  1877*  1906*
VECTB2  001374           387#    472*   473*   474
VECTC   001362           382#    468*   469*   470    503*  1676*  1699*  1707*  1728*  1735*
VECTC2  001376           388#    474*   475*   476   1833*  1858*
VECTD   001364           383#    470*   471*   506*  1738*  1763*
VECTD2  001400           389#    476*   477*
$APTHD  001000           194     200#
$ASTAT= ****** U        2716    2731
$ATYC   011466          2687    2689#
$ATY1   011442          2685#
$ATY3   011450          2631    2686#
$ATY4   011460          2192    2688#
$AUTOB  001134           232#    522*   2338   2459
$BASE   001250           296#    453
$BDADR  001122           227#
$BDDAT  001126           229#    644*   665*   689*   703*   724*   725    740*   760*   761    776*   796*   797
                         812*   832*   833    836*   837    854*   855    874*   875    891*   918*   919    934*
                         954*   955    970*   994*   995   1010*  1033*  1034   1049*  1072*  1073   1088*  1111*
                        1112   1127*  1150*  1151   1166*  1189*  1190   1205*  1228*  1229   1244*  1267*  1268
                        1283*  1304*  1341*  1342   2066*  2073*  2074   2976
$BELL   001164           247#    2179   2208
$CDW1   001254           298#    1825
$CHARC  011436          2648*   2658*  2665   2674*  2679#
$CKSWR  007732          2330#    2902
$CMTAG  001100           215#    409    410    418    424    425
$CM3  = 000000           245#
$CNTLC  010457          2341    2454#
$CNTLU  010452          2358    2453#
$CPUOP  001222           270#
$CRLF   001171           249#    2187   2208   2216   2235   2240   2244   2369   2453   2647   2682
$DBLK   011150          2560    2594   2602#
$DEVCT  001204           261#
$DEVM   001252           297#
$DOAGN  007062          2119    2128   2134#
$DTBL   011140          2563    2598#
$ENDAD  007052           178    2130#
$ENDCT  007020          2121#
$ENDMG  007071          2123    2138#
$ENULL  007066          2126    2137#
$ENV    001214           266#    516    2189   2626   2694   2718
$ENVM   001215           267#    446    2628   2633   2696
```

```
SEOP    006764          2111#
SEOPCT  007012          2118#  2122
SERFLG  001103           218#  2174*  2208   2260   2290   2292   2298*  2320
SERMAX  001115           224#   426*  2292   2315*  2320
SERROR  007126           418   2172#
SERRPC  001116           225#  2181*  2182*  2183   2208   2222   2970   2976   2979   2982
SERRTB  001256           315#  2230
SERRTY  007314          2186   2215#
SERTTL  001112           222#  2180*  2208
SESCAP  001162           246#   425*  2203   2205   2208   2314*
SETABL  001214           265#
SETEND  001256           206    299#
SFATAL  001176           258#  2722*
SFFLG   011706          2685*  2688*  2716   2725*  2733#
SFILLC  001156           243#  2651   2682
SFILLS  001155           242#  2682
SGOADR  001120           226#
SGODAT  001124           228#   643*   664*   688*   723*   725    739*   759*   761    775*   795*   797    811*
                         831*   833    837    853*   855    873*   875    890*   917*   919    932*   953*   955
                         969*   991*   992    995   1008*  1030*  1031   1034   1047*  1069*  1070   1073   1086*
                        1108*  1109   1112   1125*  1147*  1148   1151   1164*  1186*  1187   1190   1203*  1225*
                        1226   1229   1242*  1264*  1265   1268   1281*  1303*  2065*  2072   2074   2088*  2976
SGET42  007042          2127#
SGTSWR  010002          2342#  2900
SHD     =  000001         22    23
SHIBTS  001000           201#
SICNT   001104           219#  2305*  2306   2308*  2319
SILLUP  012050          2743   2759   2776#
SINTAG  001135           233#  2370   2459
SITEMB  001114           223#  2183*  2191   2208   2219
SLF     001172           250#  2208   2444   2453   2682
SLFLG   011705          2726*  2732#
SLPADR  001106           220#   427*   534*  2296*  2312*  2317   2319
SLPERR  001110           221#   428*   538*  2064*  2202   2296   2313*  2319
SMAOR1  001226           283#
SMAOR2  001232           287#
SMAOR3  001236           290#
SMAOR4  001242           293#
SMAIL   001174           202    206    256#   445    516   2189   2311   2626
SMAMS1  001224           277#
SMAMS2  001230           285#
SMAMS3  001234           288#
SMAMS4  001240           291#
SMBAOR  001002           202#
SMFLG   011704          2686*  2692   2727*  2731#
SMNEW   010475          2345   2457#
SMSGAD  001210           263#  2702*  2705
SMSGLG  001212           264#  2707*
SMSGTY  001174           257#  2700   2708*  2720   2724*
SMSWR   010464          2342   2455#
SMTYP1  001225           278#
SMTYP2  001231           286#
SMTYP3  001235           289#
SMTYP4  001241           292#
```

```
$MXCNT  007730          2309    2319#
$NULL   001154           241#   2653    2682
$NWTST= 000001           529#    583#     585     635#    656#    678#    680     716#    752#    788#    824#    866#    903#
                         946#    984#    1023#   1062#   1101#   1140#   1179#   1218#   1257#   1296#   1319#   1383#   1423#
                        1466#   1506#   1544#   1577#   1599#   1620#   1643#   1669#   1702#   1766#   1811#   1861#   1911#
                        1941#   1973#   2002#   2051#   2053    2094#
$OCNT   010730          2491*   2520*   2533#
$OMODE  010732          2486*   2490*   2495    2498*   2509*   2535#
$OVER   007714          2273    2289    2297    2307    2316#
$PASS   001202           260#    445*   2115*   2116*   2124    2137    2303    2320
$PASTM  001006           204#
$POWER  012056          2774    2779#
$PWRDN  011710           422    2743#   2771
$PWRMG  012044          2774#
$PWRUP  011762          2753    2759#
$QUES   001170           248#   2208    2388    2437    2453    2682
$RDCHR  010214          2401#   2903
$RDDEC= ****** U        2905
$RDLIN  010334          2429#   2904
$RDOCT= ****** U        2905
$RDSZ = 000010          2422#
$RTNAD  007064          2136#
$R2A  = ****** U        2905
$SAVRE= ****** U        2905
$SAVR6  012054          2752*   2760    2761*   2762*   2778#
$SCOPE  007450           416    2269#
$SETUP= 000117           406#    415     416     418     420     422     424     425     427     512     513    2113    2173
                        2199    2207    2270    2325    2459
$STUP = 177777           406#
$SVLAD  007660          2281    2310#
$SVPC = 000204           176#    181
$SWR  = 167400            10#     22      29      30      31      32      33      34      35     245     246     247     424
                          425     427     428     533     592     639     660     686     720     756     792     828     870
                          907     950     988    1027    1066    1105    1144    1183    1222    1261    1300    1323    1387
                         1427    1470    1510    1548    1581    1603    1624    1647    1673    1706    1770    1815    1865
                         1915    1945    1977    2006    2063    2098    2108    2114    2129    2135    2137    2164    2165
                         2166    2167    2168    2177    2184    2196    2200    2208    2261    2262    2263    2264    2265
                         2272    2284    2286    2287    2290    2291    2292    2299    2300    2301    2313    2316    2319
                         2775
$SWREG  001216           268#    448
$SWRMK= 000000            35      36    2265    2266    2288
$TESTN  001200           259#    537*   2311*   2970    2976    2979    2982
$TIMES  001160           245#    424*    533*    592*    639*    660*    686*   1603*   1624*   1647*   1945*   2114*   2299*
                        2306    2309*   2319
$TKB    001146           238#   2323    2334    2351    2405    2411
$TKS    001144           237#   2323    2332    2348    2372*   2403    2409
$TN   = 000053            22#    170#     529     533#    583     592#    635     639#    645     656     660#    666     678
                          686#    701     704     716     720#    736     741     752     756#    772     777     788     792#
                          808     813     824     828#    848     856     866     870#    887     892     903     907#    930
                          935     946     950#    966     971     984     988#   1007    1011    1023    1027#   1046    1050
                         1062    1066#   1085    1089    1101    1105#   1124    1128    1140    1144#   1163    1167    1179
                         1183#   1202    1206    1218    1222#   1241    1245    1257    1261#   1280    1284    1296    1300#
                         1305    1319    1323#   1337    1353    1368    1370    1383    1387#   1406    1409    1423    1427#
                         1446    1452    1466    1470#   1489    1492    1506    1510#   1529    1532    1544    1548#   1561
```

```
                    1565    1577    1581#   1588    1599    1603#   1608    1620    1624#   1631    1643    1647#   1653
                    1669    1673#   1702    1706#   1730    1766    1770#   1811    1815#   1861    1865#   1911    1915#
                    1930    1941    1945#   1957    1960    1973    1977#   1991    2002    2006#   2051    2063#   2086
                    2094    2098#
$TPB    001152       240#   2671*   2682
$TPFLG  001157       244#   2620    2682
$TPS    001150       239#   2669    2682
$TRAP   012136       420    2871#
$TRAP2  012160      2882#   2893
$TRP  = 000012      2886#   2895#   2896#   2897#   2898#   2899#   2900    2901#   2902    2903#   2904#   2905#
$TRPAD  012172      2876    2893#
$TSTM   001004       203#
$TSTNM  001102       217#    536*   2113*   2176    2208    2260    2288    2310*   2311    2316    2320
$TTYIN  010442      2430    2431    2448    2452#
$TYPBN= ****** U    2899
$TYPDS  010734      2548#   2898
$TYPE   011160      2620#   2713    2886    2894
$TYPEC  011372      2374    2650    2657    2664    2669#   2670
$TYPEX  011440      2675    2677    2680#
$TYPOC  010532      2489#   2895
$TYPON  010546      2488    2491#   2897
$TYPOS  010506      2484#   2896
$UNIT   001206       262#
$UNITM  001010       205#
$USWR   001220       269#
$VECT1  001244       294#    460
$VECT2  001246       295#
$XTSTR  007464      2275#
$$GET4= 000000      2129#
$OFILL  010731      2485*   2489*   2499    2534#
$40CAT= ****** U    2186    2272
.     = 012714        39#     45#     47#     50#     53#    176     177#    179#    181#    183#    190     191#    193#
                     195#    214#    251     413     427    428     526#   2137    2141    2208    2254#   2319    2320
                    2323    2452#   2453    2459    2602#  2682    2734#   2755    2777    2968#
.$ASTA= ****** U    2686    2689
.$X   = 001000       190#    195
```

```
BDT      982#   983   1022  1061  1100  1139  1178  1217  1256
COMMEN   166#
DELMA    371#  2141
DFC      317#   323   330   337   344   351   358   364
ECB      367#   562   573   616   630   653   674   699   713   734   749   770   785   806   821
         846    864   885   900   928   943   964   979   1005  1019  1044  1058  1083  1097  1122
         1136   1161  1175  1200  1214  1239  1253  1278  1292  1316  1335  1351  1366  1378  1404
         1417   1444  1460  1487  1500  1527  1540  1559  1573  1596  1617  1640  1662  1692  1724
         1755   1797  1851  1897  1938  1970  1999  2037  2084  2856
ENDCOM   166#
ERC      366#   554   566   608   622   646   667   691   705   727   742   763   778   799   814
         839    857   878   893   921   936   957   972   998   1012  1037  1051  1076  1090  1115
         1129   1154  1168  1193  1207  1232  1246  1271  1285  1306  1328  1344  1359  1371  1397
         1410   1437  1453  1480  1493  1520  1533  1552  1566  1589  1609  1632  1654  1685  1717
         1748   1790  1842  1889  1931  1961  1992  2030  2076  2819
ERROR    60#    559   571   613   627   651   672   696   710   732   747   768   783   804   819
         844    862   883   898   926   941   962   977   1003  1017  1042  1056  1081  1095  1120
         1134   1159  1173  1198  1212  1237  1251  1276  1290  1311  1333  1349  1364  1376  1402
         1415   1442  1458  1485  1498  1525  1538  1557  1571  1594  1614  1637  1659  1690  1722
         1753   1795  1847  1894  1936  1966  1997  2035  2081  2824
ESCAPE   166#
GETPRI   166#
GETSWR   166#   513#
MULT     166#
NEWTST   166#   529   583   635   656   678   716   752   788   824   866   903   946   984   1023
         1062   1101  1140  1179  1218  1257  1296  1319  1383  1423  1466  1506  1544  1577  1599
         1620   1643  1669  1702  1766  1811  1861  1911  1941  1973  2002  2051  2094
POP      166#   2589  2728  2729  2764  2765
POPSP2   368#   552   564   606   620   1695  1732  1759  1800  1854  1901  2041  2817
PR       369#   576   908   1678  1710  1739  1774  1783  1834  1878  2022
PUSH     166#   2548  2689  2691  2712  2745  2751
REPORT   166#
RESV     370#   496   499   502   505   1698  1727  1734  1762  1803  1857  1904  2043
SCOPE    61#    591   638   659   685   719   755   791   827   869   906   949   987   1026  1065
         1104   1143  1182  1221  1260  1299  1322  1386  1426  1469  1509  1547  1580  1602  1623
         1646   1672  1705  1769  1814  1864  1914  1944  1976  2005  2062  2097  2112
SETPRI   166#
SETTRA   2886#  2895  2896  2897  2898  2900  2902  2903  2904
SETUP    166#   407
SIGC     1380#  1381  1421  1464  1504
SKIP     166#   645   666   701   704   736   741   772   777   808   813   848   856   887   892
         930    935   966   971   1007  1011  1046  1050  1085  1089  1124  1128  1163  1167  1202
         1206   1241  1245  1280  1284  1305  1337  1353  1368  1370  1406  1409  1446  1452  1489
         1492   1529  1532  1561  1565  1588  1608  1631  1653  1730  1930  1957  1960  1991  2086
SLASH    166#
SPACE    166#
STARS    166#   174   187   189   196   210   251   254   529   531   583   590   635   637   656
         658    678   684   716   718   752   754   788   790   824   826   866   868   903   905
         946    948   984   986   1023  1025  1062  1064  1101  1103  1140  1142  1179  1181  1218
         1220   1257  1259  1296  1298  1319  1321  1383  1385  1423  1425  1466  1468  1506  1508
         1544   1546  1577  1579  1599  1601  1620  1622  1643  1645  1669  1671  1702  1704  1766
         1768   1811  1813  1861  1863  1911  1913  1941  1943  1973  1975  2002  2004  2051  2061
         2094   2096  2105  2160  2210  2257  2322  2325  2393  2422  2461  2538  2605  2684  2741
         2757   2865
```

```
                297   298   301   406   414   415   418   420   422   424   425   427   429   450   512
                518   524   526   530   531   532   533   534   535   584   585   590   591   592   593
                636   637   638   639   640   646   657   658   659   660   661   667   679   680   684
                685   686   687   702   705   717   718   719   720   737   742   753   754   755   756
                773   778   789   790   791   792   809   814   825   826   827   828   849   857   867
                868   869   870   888   893   904   905   906   907   931   936   947   948   949   950
                967   972   985   986   987   988  1008  1012  1024  1025  1026  1027  1047  1051  1063
               1064  1065  1066  1086  1090  1102  1103  1104  1105  1125  1129  1141  1142  1143  1144
               1164  1168  1180  1181  1182  1183  1203  1207  1219  1220  1221  1222  1242  1246  1258
               1259  1260  1261  1281  1285  1297  1298  1299  1300  1306  1320  1321  1322  1323  1338
               1354  1369  1371  1384  1385  1386  1387  1397  1407  1410  1420  1424  1425  1426  1427
               1437  1447  1453  1463  1467  1468  1469  1470  1480  1490  1493  1503  1507  1508  1509
               1510  1520  1530  1533  1543  1545  1546  1547  1548  1562  1566  1578  1579  1580  1581
               1589  1600  1601  1602  1603  1604  1609  1621  1622  1623  1624  1625  1632  1644  1645
               1646  1647  1648  1654  1670  1671  1672  1673  1703  1704  1705  1706  1731  1767  1768
               1769  1770  1812  1813  1814  1815  1862  1863  1864  1865  1912  1913  1914  1915  1931
               1942  1943  1944  1945  1946  1958  1961  1974  1975  1976  1977  1992  2003  2004  2005
               2006  2052  2053  2061  2062  2063  2087  2095  2096  2097  2098  2106  2107  2108  2110
               2113  2119  2122  2123  2127  2129  2135  2137  2138  2141  2161  2164  2174  2181  2186
               2187  2188  2196  2207  2208  2211  2226  2255  2258  2261  2266  2272  2274  2285  2288
               2289  2290  2292  2294  2301  2305  2310  2312  2316  2319  2320  2323  2324  2326  2354
               2390  2394  2422  2423  2430  2432  2435  2437  2453  2459  2462  2539  2606  2635  2685
               2686  2689  2716  2731  2742  2751  2752  2758  2764  2765  2775  2782  2866  2872  2875
               2894  2895  2896  2897  2898  2899  2900  2901  2902  2903  2904  2905
.EQUIV           60    61    69   114   115   116   117   118   119   120   121   122   123   142   143
                144   145   146   147   148   149   150   151
.EVEN           255   526  2254  2734  2781  2968
.IF              13    32    33    34    35    36    58   124   152   174   177   179   187   189   196
                210   214   216   245   246   247   251   252   254   277   285   288   291   294   295
                296   297   298   299   301   406   409   414   416   418   420   422   424   425   427
                445   511   512   513   516   525   529   531   533   534   535   583   585   590   592
                593   635   637   639   640   645   656   658   660   661   666   678   680   684   686
                687   701   704   716   718   720   736   741   752   754   756   772   777   788   790
                792   808   813   824   826   828   848   856   866   868   870   887   892   903   905
                907   930   935   946   948   950   966   971   984   986   988  1007  1011  1023  1025
               1027  1046  1050  1062  1064  1066  1085  1089  1101  1103  1105  1124  1128  1140  1142
               1144  1163  1167  1179  1181  1183  1202  1206  1218  1220  1222  1241  1245  1257  1259
               1261  1280  1284  1296  1298  1300  1305  1319  1321  1323  1337  1353  1368  1370  1383
               1385  1387  1392  1406  1407  1409  1423  1425  1427  1432  1446  1447  1452  1466  1468
               1470  1475  1489  1490  1492  1506  1508  1510  1515  1529  1530  1532  1544  1546  1548
               1561  1565  1577  1579  1581  1588  1599  1601  1603  1604  1608  1620  1622  1624  1625
               1631  1643  1645  1647  1648  1653  1669  1671  1673  1702  1704  1706  1730  1766  1768
               1770  1811  1813  1815  1861  1863  1865  1911  1913  1915  1930  1941  1943  1945  1946
               1957  1960  1973  1975  1977  1991  2002  2004  2006  2051  2053  2061  2063  2086  2094
               2096  2098  2105  2106  2107  2108  2109  2110  2112  2118  2121  2123  2127  2129  2135
               2137  2138  2160  2163  2174  2177  2184  2186  2187  2189  2196  2200  2207  2208  2210
               2225  2241  2257  2260  2265  2271  2272  2284  2286  2287  2288  2290  2291  2292  2301
               2303  2311  2313  2318  2319  2320  2322  2324  2325  2326  2354  2393  2394  2422  2430
               2431  2435  2436  2452  2453  2459  2461  2538  2605  2626  2684  2686  2689  2716  2731
               2741  2751  2752  2757  2764  2765  2773  2775  2779  2865  2871  2875  2886  2895  2896
               2897  2898  2899  2900  2902  2903  2904  2905
.IFF             32    34    35    36    58   175   179   181   188   190   197   211   214   217   245
                252   255   414   511   512   529   530   531   532   533   584   585   591   592   636
                637   638   639   640   646   657   658   659   660   661   667   679   680   685   686
```

```
                 702   705   717   718   719   720   737   742   753   754   755   756   773   778   789
                 790   791   792   809   814   825   826   827   828   849   857   867   868   869   870
                 888   893   904   905   906   907   931   936   947   948   949   950   967   972   985
                 986   987   988  1008  1012  1024  1025  1026  1027  1047  1051  1063  1064  1065  1066
                1086  1090  1102  1103  1104  1105  1125  1129  1141  1142  1143  1144  1164  1168  1180
                1181  1182  1183  1203  1207  1219  1220  1221  1222  1242  1246  1258  1259  1260  1261
                1281  1285  1297  1298  1299  1300  1306  1320  1321  1322  1323  1338  1354  1369  1371
                1384  1385  1386  1387  1407  1410  1420  1424  1425  1426  1427  1447  1453  1467  1468
                1469  1470  1490  1493  1503  1507  1508  1509  1510  1530  1533  1543  1545  1546  1547
                1548  1562  1566  1578  1579  1580  1581  1589  1600  1601  1602  1603  1609  1621  1622
                1623  1624  1632  1644  1645  1646  1647  1654  1670  1671  1672  1673  1703  1704  1705
                1706  1731  1767  1768  1769  1770  1812  1813  1814  1815  1862  1863  1864  1865  1912
                1913  1914  1915  1931  1942  1943  1944  1945  1958  1961  1974  1975  1976  1977  1992
                2003  2004  2005  2006  2052  2053  2062  2063  2087  2095  2096  2097  2098  2106  2109
                2113  2119  2122  2137  2161  2163  2177  2207  2208  2211  2226  2255  2258  2285  2288
                2289  2292  2319  2320  2323  2326  2394  2396  2401  2422  2423  2432  2436  2453  2462
                2539  2606  2685  2742  2758  2775  2866  2872
.IFT             526  2187  2300  2396  2401
.IFTF            526  2186  2298  2341  2394  2397
.IIF              12    17    22    23    29    30    31    32    35    36   251   255   415   418   424
                 425   427   428   512  2107  2113  2114  2125  2137  2141  2164  2165  2166  2167  2168
                2173  2199  2207  2208  2223  2248  2261  2262  2263  2264  2265  2266  2270  2299  2300
                2316  2319  2320  2323  2344  2445  2453  2459  2682  2894  2895  2896  2897  2898  2900
                2902  2903  2904
.IRP             406   529   583   635   656   678   716   752   788   824   866   903   946   984  1023
                1062  1101  1140  1179  1218  1257  1296  1319  1383  1423  1466  1506  1544  1577  1599
                1620  1643  1669  1702  1766  1811  1861  1911  1941  1973  2002  2051  2094  2271  2549
                2589  2690  2691  2712  2728  2729  2745  2751  2764  2765
.LIST              2    35    45   166   245   252   255   406   429   512   513   526   529   533   555
                 558   562   564   567   570   573   575   583   592   609   612   616   618   623   626
                 630   632   635   639   647   650   653   655   656   660   668   671   674   676   678
                 686   692   695   699   701   706   709   713   715   716   720   728   731   734   736
                 743   746   749   751   752   756   764   767   770   772   779   782   785   787   788
                 792   800   803   806   808   815   818   821   823   824   828   840   843   846   848
                 858   861   864   866   870   879   882   885   887   894   897   900   902   903   907
                 922   925   928   930   937   940   943   945   946   950   958   961   964   966   973
                 976   979   981   984   988   999  1002  1005  1007  1013  1016  1019  1021  1023  1027
                1038  1041  1044  1046  1052  1055  1058  1060  1062  1066  1077  1080  1083  1085  1091
                1094  1097  1099  1101  1105  1116  1119  1122  1124  1130  1133  1136  1138  1140  1144
                1155  1158  1161  1163  1169  1172  1175  1177  1179  1183  1194  1197  1200  1202  1208
                1211  1214  1216  1218  1222  1233  1236  1239  1241  1247  1250  1253  1255  1257  1261
                1272  1275  1278  1280  1286  1289  1292  1294  1296  1300  1307  1310  1316  1318  1319
                1323  1329  1332  1335  1337  1345  1348  1351  1353  1360  1363  1366  1368  1372  1375
                1378  1380  1383  1387  1398  1401  1404  1406  1411  1414  1417  1419  1423  1427  1438
                1441  1444  1446  1454  1457  1460  1462  1466  1470  1481  1484  1487  1489  1494  1497
                1500  1502  1506  1510  1521  1524  1527  1529  1534  1537  1540  1542  1544  1548  1553
                1556  1559  1561  1567  1570  1573  1575  1577  1581  1590  1593  1596  1598  1599  1603
                1610  1613  1617  1619  1620  1624  1633  1636  1640  1642  1643  1647  1655  1658  1662
                1664  1669  1673  1686  1689  1692  1694  1702  1706  1718  1721  1724  1726  1749  1752
                1755  1757  1766  1770  1791  1794  1797  1799  1811  1815  1843  1846  1851  1853  1861
                1865  1890  1893  1897  1899  1911  1915  1932  1935  1938  1940  1941  1945  1962  1965
                1970  1972  1973  1977  1993  1996  1999  2001  2002  2006  2031  2034  2037  2039  2051
                2063  2077  2080  2084  2086  2094  2098  2113  2129  2207  2265  2422  2820  2823  2856
                2858  2886  2894  2895  2896  2897  2898  2899  2900  2901  2902  2903  2904  2905
```

```
.MACRO    36   208   317   366   367   368   369   370   371   445   582   677   982  1380  1816
         2050  2886
.MCALL     5     6     7     8     9   166   252   429   513
.MEXIT   300
.NLIST     1    35    45   166   245   252   255   406   429   512   513   526   529   533   555
         558   562   564   567   570   573   575   583   592   609   612   616   618   623   626
         630   632   635   639   647   650   653   655   656   660   668   671   674   676   678
         686   692   695   699   701   706   709   713   715   716   720   728   731   734   736
         743   746   749   751   752   756   764   767   770   772   779   782   785   787   788
         792   800   803   806   808   815   818   821   823   824   828   840   843   846   848
         858   861   864   866   870   879   882   885   887   894   897   900   902   903   907
         922   925   928   930   937   940   943   945   946   950   958   961   964   966   973
         976   979   981   984   988   999  1002  1005  1007  1013  1016  1019  1021  1023  1027
        1038  1041  1044  1046  1052  1055  1058  1060  1062  1066  1077  1080  1083  1085  1091
        1094  1097  1099  1101  1105  1116  1119  1122  1124  1130  1133  1136  1138  1140  1144
        1155  1158  1161  1163  1169  1172  1175  1177  1179  1183  1194  1197  1200  1202  1208
        1211  1214  1216  1218  1222  1233  1236  1239  1241  1247  1250  1253  1255  1257  1261
        1272  1275  1278  1280  1286  1289  1292  1294  1296  1300  1307  1310  1316  1318  1319
        1323  1329  1332  1335  1337  1345  1348  1351  1353  1360  1363  1366  1368  1372  1375
        1378  1380  1383  1387  1398  1401  1404  1406  1411  1414  1417  1419  1423  1427  1438
        1441  1444  1446  1454  1457  1460  1462  1466  1470  1481  1484  1487  1489  1494  1497
        1500  1502  1506  1510  1521  1524  1527  1529  1534  1537  1540  1542  1544  1548  1553
        1556  1559  1561  1567  1570  1573  1575  1577  1581  1590  1593  1596  1598  1599  1603
        1610  1613  1617  1619  1620  1624  1633  1636  1640  1642  1643  1647  1655  1658  1662
        1664  1669  1673  1686  1689  1692  1694  1702  1706  1718  1721  1724  1726  1749  1752
        1755  1757  1766  1770  1791  1794  1797  1799  1811  1815  1843  1846  1851  1853  1861
        1865  1890  1893  1897  1899  1911  1915  1932  1935  1938  1940  1941  1945  1962  1965
        1970  1972  1973  1977  1993  1996  1999  2001  2002  2006  2031  2034  2037  2039  2051
        2063  2077  2080  2084  2086  2094  2098  2113  2129  2207  2265  2422  2820  2823  2856
        2858  2886  2894  2895  2896  2897  2898  2899  2900  2901  2902  2903  2904  2905
.PAGE    208   301
.REPT     45
.SBTTL    25    37    56   172   185   208   252   301   372   404   408   508   513   529   583
         635   656   678   716   752   788   824   866   903   946   984  1023  1062  1101  1140
        1179  1218  1257  1296  1319  1383  1423  1466  1506  1544  1577  1599  1620  1643  1665
        1666  1667  1669  1702  1766  1807  1808  1809  1811  1861  1911  1941  1973  2002  2051
        2094  2101  2103  2158  2208  2255  2320  2459  2536  2603  2682  2739  2863  2886  2906
.TITLE    12
.WORD     45    46    48    49    51   180   201   202   203   204   205   206   216   219   220
         221   222   225   226   227   228   229   230   231   234   235   236   257   258   259
         260   261   262   263   264   268   269   270   283   287   290   293   294   295   296
         297   298   376   377   378   379   380   381   382   383   384   385   386   387   388
         389   393   394   395   396   398   399   400   401   851  2118  2121  2136  2234  2239
        2535  2632  2679  2714  2774  2860  2861  2893  2970  2972  2974  2976  2979  2982  2985


ERRORS DETECTED:  0

*DVIBA,DVIBA/SOL/CRF=DVIBA
RUN-TIME:  95 39 7 SECONDS
CORE USED:  25K
```

L07