LSI-11

SUITCASE TEST

MD-11-DWQAA-A

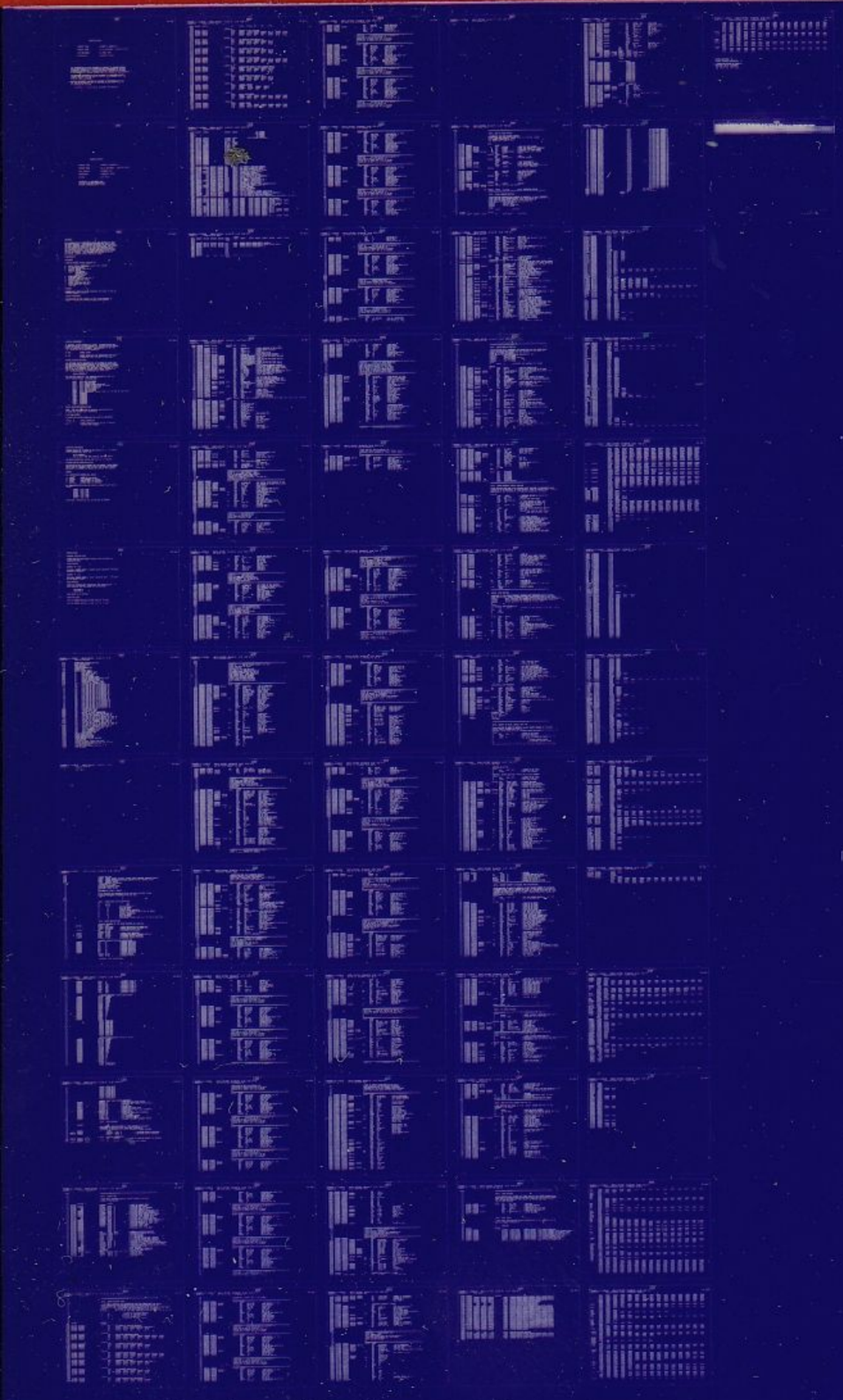
EP-DWQAA-A-DL-A

FICHE 1 OF 1

OCT 1976

digital

Made In U.S.A.



IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DWQAA-A-D
PRODUCT NAME:	LSI-11 SUITCASE 11W03-AA SYSTEM
DATE RELEASED:	21 APRIL 1976
MAINTAINER:	DIAGNOSTIC GROUP

PROGRAM HISTORY

PRODUCT CODE: MAINDEC-11-DWQAA-A-D
PRODUCT NAME: LSI-11 SUITCASE 11W03-AA SYSTEM
DATE CREATED: DECEMBER 1975
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: JOHN EGOLF

1. ABSTRACT

MD-11-DWQAA-A IS A DIAGNOSTIC WHICH WAS WRITTEN TO TEST THE M5911 MODULE. THIS PROGRAM VERIFIES THE DATA WITHIN THE ROM, VERIFIES FUNCTIONS OF THE 600HZ CLOCK, AND CHECKS THE BUS BUFFER BY MEANS OF THE TEST BOX PROVIDED WITH THE 11W03-AA SYSTEM. THIS PROGRAM WAS WRITTEN EXPRESSLY FOR THE 11W03-AA SYSTEM AND TAKES ADVANTAGE OF ALL UNITS WITHIN THE SYSTEM. THIS PROGRAM MUST BE RUN AFTER ALL CPU TESTS, MEMORY TESTS, AND DLV11 TESTS.

2. REQUIREMENTS

2.1 EQUIPMENT

11W03-AA SYSTEM, WHICH CONSISTS OF:

A. KD11-F LSI-11 PROCESSOR (WITH 4K OF MEMORY)
B. 12K MOS RAM MEMORY
C. 2 DLV11 SERIAL LINE UNITS
D. SPECIAL MODULE
   M5911 - BUS BUFFER
           BOOTSTRAP ROM
           600HZ CLOCK
E. H780 POWER SUPPLY
F. ALUMINUM CASE 17 X 21 X 7.5"
G. SPECIAL OPERATORS PANEL
H. POWER CORD
I. CABLE, EXTERNAL BUS -6FT.
J. M9270 BACKPLANE ASSEMBLY
K. TEST BOX

2.2 STORAGE

PROGRAM WILL LOAD IN 4K BUT RESERVES THE RIGHT TO USE 16K (ADDRESS 000000 - 077777).

3. LOADING PROCEDURE

THIS PROGRAM MAY BE LOADED LIKE ANY OTHER PROGRAM IS LOADED. THERE ARE NO SPECIAL LOADING PROCEDURES.

4. STARTING PROCEDURE

IF DESIRED, AFTER PROGRAM IS LOADED, SOFTWARE SWITCH REGISTER MAY BE SET AS PER SECTION 4.1 (USING LSI-ODT) BEFORE START OF EXECUTION OF PROGRAM. AS DISCUSSED IN SECTION 4.2, THERE ARE TWO STARTING ADDRESSES:

SA 200	NORMAL START
SA 210	NORMAL START BUT THE OPPORTUNITY TO "GOTO" A SELECTED TEST IS GIVEN TO THE USER.

4.1 CONTROL SWITCH SETTINGS

THE SOFTWARE SWITCH REGISTER MAY BE ALTERED BY USING LSI-ODT AND MODIFYING ADDRESS 000176. AN EASIER WAY TO CHANGE THE SWITCH REGISTER IS THAT AFTER THE DIAGNOSTIC IS STARTED AND WHILE IT IS RUNNING THE USER MAY TYPE "CONTROL G" (↑G) AND IF THE CPU PRIORITY ALLOWS (AND A RESET INSTRUCTION ISN'T BEING EXECUTED) THE PROGRAM WILL PRINT OUT:

(SWR)=/000000/=/

THE USER MAY THEN HIT (CR) CARRIAGE RETURN OR TYPE (IN OCTAL) NEW SETTINGS. SWITCHES ARE:

SW15	SET: HALT ON ERROR
SW14	SET: LOOP ON TEST
SW13	SET: INHIBIT ERROR PRINT-OUT.
SW12	SET: ESCAPE TO NEXT TEST ON ERROR.
SW11	SET: INHIBIT ITERATIONS
SW10	SET: BELL ON ERROR
SW09	SET: LOOP ON ERROR
SW08	SET: CONFIDENCE; -PRINT TEST # AT 1ST END OF EACH TEST.
SW07	RESERVED
SW06	RESERVED
SW05	RESERVED
SW04	RESERVED
SW03	RESERVED
SW02	RESERVED
SW01	RESERVED
SW00	RESERVED

4.1.2 SWITCH REGISTER RESTRICTIONS

NONE. FOR THE EXCEPTION OF SW12 ALL SWITCH REGISTER OPTIONS ARE "SYSMAC.SML" COMPATABLE.

4.2 STARTING ADDRESS

AS MENTIONED BEFORE THERE ARE TWO STARTING ADDRESSES:

SA 200	NORMAL OPERATION
SA 210	NORMAL OPERATION WITH OPTION TO SPECIFY STARTING TEST NUMBER.

5. OPERATING PROCEDURE

PROGRAM SHOULD BE STARTED AS PER SECTION 4.2. ON INITIAL START PROGRAM WILL PRINT OUT:

MD-11-DWOAA-A
LSI-11 SUIT-CASE ROM, 600 HZ, AND BUS TESTS

AND BEGIN EXECUTION (OR ASK FOR TEST NO. IF SA=210).

5.2 PROGRAM AND/OR OPERATOR ACTION

SWR SHOULD BE SET TO "100000" HALT ON ERROR. AFTER ERROR MODIFY SWR TO "060000" LOOP ON TEST AND INHIBIT PRINTOUT AND START AT ADDRESS 210 SPECIFING FAILING TEST NUMBER.

6. ERRORS

ALL CONTROLABLE ERRORS WILL PRINT:

1) ERRPC ERROR PC (IN OCTAL)
2) STSTNM TEST NUMBER (IN OCTAL)
3) SPASS PASSES MADE (IN DECIMAL)
4) SERTLL ERRORS MADE (IN DECIMAL)

ALL PROCESSOR REGISTERS ARE SAVED IN MEMORY LOCATIONS ON ERRORS.

SAVR0 LOC: 1420
SAVR1 LOC: 1422
SAVR2 LOC: 1424
SAVR3 LOC: 1426
SAVR4 LOC: 1430
SAVR5 LOC: 1432

ADDITIONAL INFORMATION WILL BE PRINTED AS NEEDED.

7. RESTRICTIONS

7.1 HARDWARE RESTRICTIONS

SYSTEM MUST BE CONFIGURED *EXACTLY* AS OUTLINED FOR 11WD3-AA SYSTEM.

8. MISCELLANEOUS

8.1 CONTROL "G" (↑G)

THIS WILL ENABLE USER TO CHANGE SWITCH REGISTER (PROVIDED PROGRAM IS RUNNING).

8.2 CONTROL "T" (↑T)

THIS WILL ENABLE USER TO "GOTO" SELECTED TEST. (CPU MUST NOT BE DOING A RESET.)

8.3 PASS COMPLETE

THIS IS A "SYSMAC.SML" COMPATIBLE END PASS ROUTINE. AT PASS COMPLETE THE FOLLOWING WILL BE PRINTED:

```
END PASS #      1
END PASS #      2
```

PASS COUNT IS IN DECIMAL.

8.4 EXECUTION TIME

WITH NO ERRORS AND SW11=0 PASS TIME IS <8 MINS.

WITH NO ERRORS AND SW11=1 PASS TIME IS <2 MINS.

```
4694
4701                                    .ENABLE ABS,AMA
4702                                    .MCALL  .SWREG,.EQUAT,.SCATN,.SCMTAG,.SEOP,.SERROR,.SERRTYP,.TYPNAM
4703                                    .MCALL  .SIXBP,.$TYPOCT,.$TYPDEC,.$SCOPE,.$TYPE,.HEADER,.SETUP
4704                                    .MCALL  .$PDOCT,.$READ
4713                                    .TITLE  MAINDEC-11-11WO3A-A
  (1)                                   ;*COPYRIGHT (C) 1975
  (1)                                   ;*DIGITAL EQUIPMENT CORP.
  (1)                                   ;*MAYNARD, MASS. 01754
  (1)                                   ;*
  (1)                                   ;*PROGRAM BY JOHN C. EGOLF
  (1)                                   ;*
  (1)                                   ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
  (1)                                   ;*PACKAGE (MAINDEC-11-DZQAC-B1),AUG 29,1975.
  (1)                                   ;*
4714
4715
  (1)                                   .SBTTL  OPERATIONAL SWITCH SETTINGS
  (1)                                   ;*
  (1)                                   ;*      SWITCH                  USE
  (1)                                   ;*      ------          --------------------
  (1)                                   ;*        15            HALT ON ERROR
  (1)                                   ;*        14            LOOP ON TEST
  (1)                                   ;*        13            INHIBIT ERROR TYPEOUTS
  (1)                                   ;*        12            SWITCH 12: ESCAPE TO NEXT TEST ON ERROR.
  (1)                                   ;*        11            INHIBIT ITERATIONS
  (1)                                   ;*        10            BELL ON ERROR
  (1)                                   ;*         9            LOOP ON ERROR
  (1)                                   ;*         8            CONFIDENCE: -PRINT TEST # AT 1ST END OF EACH TEST
4716
  (1)                                   .SBTTL  BASIC DEFINITIONS
  (1)
  (1)                                   ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
  (1)           001100                  STACK=  1100
  (1)                                   .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
  (1)                                   .EQUIV  TOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
  (1)           177776                  PS=     177776          ;;PROCESSOR STATUS WORD
  (1)                                   .EQUIV  PS,PSW
  (1)           177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
  (1)           177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
  (1)           177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
  (1)           177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
  (1)
  (1)                                   ;*GENERAL PURPOSE REGISTER DEFINITIONS
  (1)           000000                  R0=     %0              ;;GENERAL REGISTER
  (1)           000001                  R1=     %1              ;;GENERAL REGISTER
  (1)           000002                  R2=     %2              ;;GENERAL REGISTER
  (1)           000003                  R3=     %3              ;;GENERAL REGISTER
  (1)           000004                  R4=     %4              ;;GENERAL REGISTER
  (1)           000005                  R5=     %5              ;;GENERAL REGISTER
  (1)           000006                  R6=     %6              ;;GENERAL REGISTER
  (1)           000007                  R7=     %7              ;;GENERAL REGISTER
  (1)                                   .EQUIV  R6,SP           ;;STACK POINTER
  (1)                                   .EQUIV  R7,PC           ;;PROGRAM COUNTER
  (1)
  (1)                                   ;*PRIORITY LEVEL DEFINITIONS
```

```
(1)                     PR0=    0               ;;PRIORITY LEVEL 0
(1)                     PR1=    40              ;;PRIORITY LEVEL 1
(1)                     PR2=    100             ;;PRIORITY LEVEL 2
(1)                     PR3=    140             ;;PRIORITY LEVEL 3
(1)                     PR4=    200             ;;PRIORITY LEVEL 4
(1)                     PR5=    240             ;;PRIORITY LEVEL 5
(1)                     PR6=    300             ;;PRIORITY LEVEL 6
(1)                     PR7=    340             ;;PRIORITY LEVEL 7
(1)
(1)                     ;*"SWITCH REGISTER" SWITCH DEFINITIONS
(1)             100000  SW15=   100000
(1)             040000  SW14=   40000
(1)             020000  SW13=   20000
(1)             010000  SW12=   10000
(1)             004000  SW11=   4000
(1)             002000  SW10=   2000
(1)             001000  SW09=   1000
(1)             000400  SW08=   400
(1)             000200  SW07=   200
(1)             000100  SW06=   100
(1)             000040  SW05=   40
(1)             000020  SW04=   20
(1)             000010  SW03=   10
(1)             000004  SW02=   4
(1)             000002  SW01=   2
(1)             000001  SW00=   1
(1)                     .EQUIV  SW09,SW9
(1)                     .EQUIV  SW08,SW8
(1)                     .EQUIV  SW07,SW7
(1)                     .EQUIV  SW06,SW6
(1)                     .EQUIV  SW05,SW5
(1)                     .EQUIV  SW04,SW4
(1)                     .EQUIV  SW03,SW3
(1)                     .EQUIV  SW02,SW2
(1)                     .EQUIV  SW01,SW1
(1)                     .EQUIV  SW00,SW0
(1)
(1)                     ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)             100000  BIT15=  100000
(1)             040000  BIT14=  40000
(1)             020000  BIT13=  20000
(1)             010000  BIT12=  10000
(1)             004000  BIT11=  4000
(1)             002000  BIT10=  2000
(1)             001000  BIT09=  1000
(1)             000400  BIT08=  400
(1)             000200  BIT07=  200
(1)             000100  BIT06=  100
(1)             000040  BIT05=  40
(1)             000020  BIT04=  20
(1)             000010  BIT03=  10
(1)             000004  BIT02=  4
(1)             000002  BIT01=  2
(1)             000001  BIT00=  1
(1)                     .EQUIV  BIT09,BIT9
(1)                     .EQUIV  BIT08,BIT8
```

```
(1)                             .EQUIV  BIT07,BIT7
(1)                             .EQUIV  BIT06,BIT6
(1)                             .EQUIV  BIT05,BIT5
(1)                             .EQUIV  BIT04,BIT4
(1)                             .EQUIV  BIT03,BIT3
(1)                             .EQUIV  BIT02,BIT2
(1)                             .EQUIV  BIT01,BIT1
(1)                             .EQUIV  BIT00,BIT0
(1)
(1)                             .#BASIC "CPU" TRAP VECTOR ADDRESSES
(1)     000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
(1)     000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1)     000014                  TBITVEC=14              ;;"T" BIT
(1)     000014                  TRTVEC= 14              ;;TRACE TRAP
(1)     000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
(1)     000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)     000024                  PWRVEC= 24              ;;POWER FAIL
(1)     000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
(1)     000034                  TRAPVEC=34              ;;"TRAP" TRAP
(1)     000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
(1)     000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
(1)     000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
4717
(1)                             .SBTTL  TRAP CATCHER
(1)
(1)     000000                          .=0
(1)                             ;#ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
(1)                             ;#SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1)                             ;#LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
(1)     000174                          .=174
(1)  000174  000000             DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
(1)  000176  000000             SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
(1)
(1)                             .SBTTL  STARTING ADDRESS(ES)
(1)  000200  000137  003732             JMP     @#START         ;;JUMP TO STARTING ADDRESS OF PROGRAM
4718    000210
4719  000210  000137  003726   .=210
                                JMP     SETTST          ;ASK FOR TEST NUMBER.
```

```
4720                                    ;*****************************************************************
 (1)
 (1)                                    .SBTTL  COMMON TAGS
 (1)
 (1)                                    ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
 (1)                                    ;*USED IN THE PROGRAM.
 (1)
 (1)          001100                            .=1100
 (1)  001100                            $CMTAG:                 ;;START OF COMMON TAGS
 (1)  001100  000000                    $PASS:  .WORD   0       ;;CONTAINS PASS COUNT
 (1)  001102      000                   $TSTNM: .BYTE   0       ;;CONTAINS THE TEST NUMBER
 (1)  001103      000                   $ERFLG: .BYTE   0       ;;CONTAINS ERROR FLAG
 (1)  001104  000000                    $ICNT:  .WORD   0       ;;CONTAINS SUBTEST ITERATION COUNT
 (1)  001106  000000                    $LPADR: .WORD   0       ;;CONTAINS SCOPE LOOP ADDRESS
 (1)  001110  000000                    $LPERR: .WORD   0       ;;CONTAINS SCOPE RETURN FOR ERRORS
 (1)  001112  000000                    $ERTTL: .WORD   0       ;;CONTAINS TOTAL ERRORS DETECTED
 (1)  001114      000                   $ITEMB: .BYTE   0       ;;CONTAINS ITEM CONTROL BYTE
 (1)  001115      001                   $ERMAX: .BYTE   1       ;;CONTAINS MAX. ERRORS PER TEST
 (1)  001116  000000                    $ERRPC: .WORD   0       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
 (1)  001120  000000                    $GDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'GOOD' DATA
 (1)  001122  000000                    $BDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'BAD' DATA
 (1)  001124  000000                    $GDDAT: .WORD   0       ;;CONTAINS 'GOOD' DATA
 (1)  001126  000000                    $BDDAT: .WORD   0       ;;CONTAINS 'BAD' DATA
 (1)  001130  000000                            .WORD   0       ;;RESERVED--NOT TO BE USED
 (1)  001132  000000                            .WORD   0
 (1)  001134  000000                            .WORD   0
 (1)  001136  177570                    $SWR:   .WORD   DSWR    ;;ADDRESS OF SWITCH REGISTER
 (1)  001140  177570                    DISPLAY: .WORD  DDISP   ;;ADDRESS OF DISPLAY REGISTER
 (1)  001142  177560                    $TKS:   177560          ;;TTY KBD STATUS
 (1)  001144  177562                    $TKB:   177562          ;;TTY KBD BUFFER
 (1)  001146  177564                    $TPS:   177564          ;;TTY PRINTER STATUS REG. ADDRESS
 (1)  001150  177566                    $TPB:   177566          ;;TTY PRINTER BUFFER REG. ADDRESS
 (1)  001152      000                   $NULL:  .BYTE   0       ;;CONTAINS NULL CHARACTER FOR FILLS
 (1)  001153      002                   $FILLS: .BYTE   2       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
 (1)  001154      012                   $FILLC: .BYTE   12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
 (1)  001155      000                   $TPFLG: .BYTE   0       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
 (3)  001156  000000                    $TMPO:  .WORD   0       ;;USER DEFINED
 (1)  001160  000000                    $TIMES: 0               ;;MAX. NUMBER OF ITERATIONS
 (1)  001162  000000                    $ESCAPE:0               ;;ESCAPE ON ERROR ADDRESS
 (1)  001164  177607  000377            $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
 (1)  001170      077                   $QUES:  .ASCII  /?/     ;;QUESTION MARK
 (1)  001171      015                   $CRLF:  .ASCII  <15>    ;;CARRIAGE RETURN
 (1)  001172  000012                    $LF:    .ASCIZ  <12>    ;;LINE FEED
```

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

                        .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 SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                        ;*NOTE1:        IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
                        ;*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

001174                  $ERRTB:
                        ;ERROR 1
001174  001442                  EM1     ;NO "BREPLY" FROM ROM ADDRESS.
001176  002520                  DH1     ;ERRPC  TSTNO   PASCNT  ERRCNT  ROM ADDRESS
001200  003650                  DT1     ;$ERRPC MSKTST  $PASS   $ERTTL  SAVR0
001202  003716                  DF1     ;OCTAL  OCTAL   DEC     DEC     OCTAL

                        ;ERROR 2
001204  001500                  EM2     ;ROM DATA COMPARISON ERROR.
001206  002574                  DH2     ;ERRPC  TSTNO   PASCNT  ERRCNT  ROMADD  WANTED  FOUND
001210  003664                  DT2     ;$ERRPC MSKTST  $PASS   $ERTTL  SAVR0   SAVR5   SAVR4
001212  003716                  DF1     ;OCTAL  OCTAL   DEC     DEC     OCTAL   OCTAL   OCTAL

                        ;ERROR 3
001214  001533                  EM3     ;NO "BREPLY" FROM 600 HZ CLOCK.
001216  002662                  DH3     ;ERRPC  TSTNO   PASCNT  ERRCNT  CLKCSR
001220  003650                  DT1     ;$ERRPC MSKTST  $PASS   $ERTTL  SAVR0
001222  003716                  DF1     ;OCTAL  OCTAL   DEC     DEC     OCTAL

                        ;ERROR 4
001224  001572                  EM4     ;CLOCK R/W BIT6 FAILED.
001226  002731                  DH4     ;ERRPC  TSTNO   PASCNT  ERRCNT  CLKCSR  WANTED  FOUND
001230  003664                  DT2     ;$ERRPC MSKTST  $PASS   $ERTTL  SAVR0   SAVR5   SAVR4
001232  003716                  DF1     ;OCTAL  OCTAL   DEC     DEC     OCTAL   OCTAL   OCTAL

                        ;ERROR 5
001234  001624                  EM5     ;UNEXPECTED INTERUPT.
001236  003017                  DH5     ;ERRPC  TSTNO   PASCNT  ERRCNT  (SP)-TRAP PC
001240  003650                  DT1     ;$ERRPC MSKTST  $PASS   $ERTTL  SAVR0
001242  003716                  DF1     ;OCTAL  OCTAL   DEC     DEC     OCTAL

                        ;ERROR 6
001244  001651                  EM6     ;NO INTERUPT FROM CLOCK.
001246  002662                  DH3     ;ERRPC  TSTNO   PASCNT  ERRCNT  CLKCSR
001250  003650                  DT1     ;$ERRPC MSKTST  $PASS   $ERTTL  SAVR0
001252  003716                  DF1     ;OCTAL  OCTAL   DEC     DEC     OCTAL

                        ;ERROR 7
001254  001701                  EM7     ;CLOCK INTERUPTS NOT CONSISTANT.
001256  002662                  DH3     ;ERRPC  TSTNO   PASCNT  ERRCNT  CLKCSR
001260  003650                  DT1     ;$ERRPC MSKTST  $PASS   $ERTTL  SAVR0
```

```
 4760  001262  003716                    DF1     ;OCTAL   OCTAL   DEC     DEC     OCTAL
 4761
 4762                                ;ERROR 10
 4763  001264  001741                    EM10    ;DEVICE "A" R/W FAILURE.
 4764  001266  003210                    DH10    ;ERRPC   TSTNO   PASCNT  ERRCNT  DEV.A    WANTED  FOUND
 4765  001270  003664                    DT2     ;$ERRPC  MSKTST  $PASS   $ERTTL  SAVR0    SAVR5   SAVR4
 4766  001272  003716                    DF1     ;OCTAL   OCTAL   DEC     DEC     OCTAL    OCTAL   OCTAL
 4767
 4768                                ;ERROR 11
 4769  001274  001771                    EM11    ;NO "BREPLY" FROM DEVICE 'A'.
 4770  001276  003074                    DH6     ;ERRPC   TSTNO   PASCNT  ERRCNT  DEV.A
 4771  001300  003650                    DT1     ;$ERRPC  MSKTST  $PASS   $ERTTL  SAVR0
 4772  001302  003716                    DF1     ;OCTAL   OCTAL   DEC     DEC     OCTAL
 4773
 4774                                ;ERROR 12
 4775  001304  002026                    EM12    ;NO "BREPLY" FROM DEVICE 'B'.
 4776  001306  003142                    DH7     ;ERRPC   TSTNO   PASCNT  ERRCNT  DEV.B
 4777  001310  003650                    DT1     ;$ERRPC  MSKTST  $PASS   $ERTTL  SAVR0
 4778  001312  003716                    DF1     ;OCTAL   OCTAL   DEC     DEC     OCTAL
 4779
 4780                                ;ERROR 13
 4781  001314  002063                    EM13    ;DEVICE "B" R/W FAILURE.
 4782  001316  003276                    DH11    ;ERRPC   TSTNO   PASCNT  ERRCNT  DEV.B    WANTED  FOUND
 4783  001320  003664                    DT2     ;$ERRPC  MSKTST  $PASS   $ERTTL  SAVR0    SAVR5   SAVR4
 4784  001322  003716                    DF1     ;OCTAL   OCTAL   DEC     DEC     OCTAL    OCTAL   OCTAL
 4785
 4786                                ;ERROR 14
 4787  001324  002113                    EM14    ;UNEXPECTED "BREPLY" FROM UNKNOWN DEVICE.
 4788  001326  003364                    DH12    ;ERRPC   TSTNO   PASCNT  ERRCNT  DEVICE
 4789  001330  003650                    DT1     ;$ERRPC  MSKTST  $PASS   $ERTTL  SAVR0
 4790  001332  003716                    DF1     ;OCTAL   OCTAL   DEC     DEC     OCTAL
 4791
 4792                                ;ERROR 15
 4793  001334  002164                    EM15    ;NO INTERUPT FROM DEVICE 'B'.
 4794  001336  003142                    DH7     ;ERRPC   TSTNO   PASCNT  ERRCNT  DEV.B
 4795  001340  003650                    DT1     ;$ERRPC  MSKTST  $PASS   $ERTTL  SAVR0
 4796  001342  003716                    DF1     ;OCTAL   OCTAL   DEC     DEC     OCTAL
 4797
 4798                                ;ERROR 16
 4799  001344  002220                    EM16    ;GENERAL ERROR!
 4800  001346  003433                    DH13    ;ERRPC   TSTNO   PASCNT  ERRCNT
 4801  001350  003704                    DT3     ;$ERRPC  MSKTST  $PASS   $ERTTL
 4802  001352  003716                    DF1     ;OCTAL   OCTAL   DEC     DEC
 4803
 4804                                ;ERROR 17
 4805  001354  002237                    EM17    ;DLV11 DATA ERROR.
 4806  001356  003474                    DH14    ;ERRPC   TSTNO   PASCNT  ERRCNT  DLV11    WANTED  FOUND
 4807  001360  003664                    DT2     ;$ERRPC  MSKTST  $PASS   $ERTTL  SAVR0    SAVR5   SAVR4
 4808  001362  003716                    DF1     ;OCTAL   OCTAL   DEC     DEC     OCTAL    OCTAL   OCTAL
 4809
 4810                                ;ERROR 20
 4811  001364  002261                    EM20    ;MEMORY ERROR.
 4812  001366  003562                    DH15    ;ERRPC   TSTNO   PASCNT  ERRCNT  MEMORY   WANTED  FOUND
 4813  001370  003664                    DT2     ;$ERRPC  MSKTST  $PASS   $ERTTL  SAVR0    SAVR5   SAVR4
 4814  001372  003716                    DF1     ;OCTAL   OCTAL   DEC     DEC     OCTAL    OCTAL   OCTAL
 4815
```

```
                                ;
                                ; 16K 100000
                                ; 12K 060000
                                ;  8K 040000
                                ;  4K 020000
                                ;
001374  100000        MEMSIZ: 100000
001376  177546        CLKCSR: 177546          ;CLOCK CONTROL REGISTER
001400  000100        CLKVEC: 100
001402  000102        CLKPTY: 102
001404  172524        DEV.A:  172524
001406  165250        DEV.B:  165250
001410  000340        B.AVEC: 340
001412  000342        B.APTY: 342
001414  000344        B.BVEC: 344
001416  000346        B.BPTY: 346
001420  000000        SAVR0:  0
001422  000000        SAVR1:  0
001424  000000        SAVR2:  0
001426  000000        SAVR3:  0
001430  000000        SAVR4:  0
001432  000000        SAVR5:  0
001434  000000        MSKTST: 0
001436  000000        TSTFLG: 0
001440  000000        XXSCOP: 0

001442  047516 021040 051102  EM1:    .ASCIZ  /NO "BREPLY" FROM ROM ADDRESS./
001500  047522 020115 040504  EM2:    .ASCIZ  /ROM DATA COMPARISON ERROR./
001533     116 020117 041042  EM3:    .ASCIZ  /NO "BREPLY" FROM 600 HZ CLOCK./
001572  046103 041517 020113  EM4:    .ASCIZ  'CLOCK R/W OF BIT6 FAILED.'
001624  047125 054105 042520  EM5:    .ASCIZ  /UNEXPECTED INTERUPT./
001651     116 020117 047111  EM6:    .ASCIZ  /NO INTERUPT FROM CLOCK./
001701     103 047514 045503  EM7:    .ASCIZ  /CLOCK INTERUPTS NOT CONSISTANT./
001741     104 053105 041511  EM10:   .ASCIZ  'DEVICE "A" R/W FAILURE.'
001771     116 020117 041042  EM11:   .ASCIZ  /NO "BREPLY" FROM DEVICE 'A'./
002026  047516 021040 051102  EM12:   .ASCIZ  /NO "BREPLY" FROM DEVICE 'B'./
002063     104 053105 041511  EM13:   .ASCIZ  "DEVICE 'B' R/W FAILURE."
002113     125 042516 050130  EM14:   .ASCIZ  /UNEXPECTED "BREPLY" FROM UNKNOWN DEVICE./
002164  047516 044440 052116  EM15:   .ASCIZ  /NO INTERUPT FROM DEVICE 'B'/
002220  042507 042516 040522  EM16:   .ASCIZ  /GENERAL ERROR!/
002237     104 053114 030461  EM17:   .ASCIZ  /DLV11 DATA ERROR./
002261     115 046505 051117  EM20:   .ASCIZ  /MEMORY ERROR./
002277     015 044412 041516  EM21:   .ASCII  <15><12>/INCORRECT MEMORY SIZE FOUND./
002336     015 041412 040510          .ASCII  <15><12>/CHANGE LOCATION "MEMSIZ" TO SPECIFIC/
002405     015 051412 055111          .ASCII  <15><12>/SIZE IN YOUR CONFIGURATION./
002440  005015 046442 046505          .ASCII  <15><12>/"MEMSIZ" SHOULD BE 100000 FOR A 16K SYSTEM./<15><12><0>

002520  051105 050122 020103  DH1:    .ASCIZ  /ERRPC   TSTNO  PASCNT  ERRCNT  ROM ADDRESS/
002574  051105 050122 020103  DH2:    .ASCIZ  /ERRPC   TSTNO  PASCNT  ERRCNT  ROMADD  WANTED  FOUND/
002662  051105 050122 020103  DH3:    .ASCIZ  /ERRPC   TSTNO  PASCNT  ERRCNT  CLKCSR/
002731     105 051122 041520  DH4:    .ASCIZ  /ERRPC   TSTNO  PASCNT  ERRCNT  CLKCSR  WANTED  FOUND/
003017     105 051122 041520  DH5:    .ASCIZ  /ERRPC   TSTNO  PASCNT  ERRCNT  (SP)-TRAP PC/
003074  051105 050122 020103  DH6:    .ASCIZ  /ERRPC   TSTNO  PASCNT  ERRCNT  DEV.A/
003142  051105 050122 020103  DH7:    .ASCIZ  /ERRPC   TSTNO  PASCNT  ERRCNT  DEV.B/
003210  051105 050122 020103  DH10:   .ASCIZ  /ERRPC   TSTNO  PASCNT  ERRCNT  DEV.A   WANTED  FOUND/
003276  051105 050122 020103  DH11:   .ASCIZ  /ERRPC   TSTNO  PASCNT  ERRCNT  DEV.B   WANTED  FOUND/
003344  051105 050122 020103  DH12:   .ASCIZ  /ERRPC   TSTNO  PASCNT  ERRCNT  DEVICE/
003433     105 051122 041520  DH13:   .ASCIZ  /ERRPC   TSTNO  PASCNT  ERRCNT  /
003474  051105 050122 020103  DH14:   .ASCIZ  /ERRPC   TSTNO  PASCNT  ERRCNT  DLV11   WANTED  FOUND/
```

```
003562 051105 050122 020103 DH15:   .ASCIZ  /ERRPC    TSTNO  PASCNT  ERRCNT  MEMORY  WANTED  FOUND/
                            .EVEN

003650 001116 001434 001100 DT1:    .WORD   $ERRPC,MSKTST,$PASS,$ERTTL,SAVR0,0
003664 001116 001434 001100 DT2:    .WORD   $ERRPC,MSKTST,$PASS,$ERTTL,SAVR0,SAVR5,SAVR4,0
003704 001116 001434 001100 DT3:    .WORD   $ERRPC,MSKTST,$PASS,$ERTTL,0
003716                      DT4:
003716                      DT5:
003716                      DT6:
003716                      DT7:
003716                      DT10:
003716    000    000    001 DF1:    .BYTE   0,0,1,1,0,0,0,0
                            .EVEN
```

```
 4894  003726  005237  001436          SETTST: INC     TSTFLG          ;
 4895  003732                          START:
   (1) 003732  012706  001100                  MOV     #SCNTAG,R6      ;;FIRST LOCATION TO BE CLEARED
   (1) 003736  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
   (1) 003740  022706  001126                  CMP     #SCOPAT,R6      ;;DONE?
   (1) 003744  001374                          BNE     .-6             ;;LOOP BACK IF NO
   (1) 003746  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
   (1) 003752  012737  011270  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
   (1) 003760  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
   (1) 003766  012737  011602  000030          MOV     #ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
   (1) 003774  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
   (1) 004002  012737  013454  000034          MOV     #TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
   (1) 004010  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
   (1) 004016  013737  011202  011174          MOV     $ENDCT,SEOPCT   ;;SETUP END-OF-PROGRAM COUNTER
   (1) 004024  005037  001160                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
   (1) 004030  005037  001162                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
   (1) 004034  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
   (1) 004042  012737  004042  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
   (1) 004050  012737  004050  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
   (2) 004056  013746  000004                  MOV     @#4,-(SP)       ;;SAVE ERROR VECTOR
   (2) 004062  013746  000006                  MOV     @#6,-(SP)
   (2) 004066  012737  004102  000004          MOV     #64$,4          ;;SET UP TIME OUT VECTOR
   (2) 004074  005777  175036                  TST     @SWR            ;;TRY TO REFERENCE HARDWARE SWR
   (2) 004100  000407                          BR      65$             ;;BRANCH IF NO TIMEOUT TRAP OCCURS
   (2) 004102  012737  000176  001136  64$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
   (2) 004110  012737  000174  001140          MOV     #DISREG,DISPLAY ;;POINT TO SOFTWARE DISPLAY REG
   (2) 004116  022626                          CMP     (SP)+,(SP)+     ;;RESTORE STACK
   (2) 004120  012637  000006          65$:    MOV     (SP)+,@#6       ;;RESTORE ERROR VECTOR
   (2) 004124  012637  000004                  MOV     (SP)+,@#4
 4896  004130  005227  177777                  INC     #-1             ;;FIRST TIME?
   (1) 004134  001055                          BNE     66$             ;;BRANCH IF NO
   (1) 004136  022737  011234  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
   (1) 004144  001451                          BEQ     66$             ;;BRANCH IF NO
   (1) 004146  104400  004154                  TYPE    ,67$            ;;TYPE ASCIZ STRING
   (1) 004152  000446                          BR      66$             ;;GET OVER THE ASCIZ
   (1)                             ;;67$:      .ASCIZ  <200>/MD-11-11WD3A-A/<15><12>/LSI-11 SUIT-CASE  ROM, 600 HZ CLK, AND BUS
   (1) 004270                      66$:
 4897  004270  012737  000006  000004          MOV     #6,4            ;
 4898  004276  005037  000006                  CLR     6               ;
 4899  004302  013777  001402  175070          MOV     CLKPTY,@CLKVEC  ;
 4900  004310  012777  000002  175064          MOV     #2,@CLKPTY      ;
 4901  004316  013777  001412  175064          MOV     @.APTY,@#.AVEC  ;
 4902  004324  005077  175062                  CLR     @#.APTY         ;
 4903  004330  013777  001416  175056          MOV     @.BPTY,@#.BVEC  ;
 4904  004336  005077  175054                  CLR     @#.BPTY         ;
 4905  004342  005037  000022                  CLR     @#IOTVEC+2      ;ZERO SCOPE PRIO.
 4906  004346  005037  001440                  CLR     XXSCOP          ;
 4907  004352  012737  014516  000060          MOV     #KBISR,@#60     ;SET VECTOR
 4908  004360  012737  000200  000062          MOV     #200,@#62       ;SET PRIO
 4909  004366  012777  000100  174546          MOV     #100,@$TKS      ;SET INTR ENABLE
 4910  004374  106427  000000                  MTPS    #0              ;SET PSW TO 0
 4911  004400  005737  001436                  TST     TSTFLG          ;FLAG SET?
 4912  004404  001431                          BEQ     TST1            ;
 4913  004406  005037  001436                  CLR     TSTFLG          ;ZERO FLAG
 4914  004412  104400  014631          X1$:    TYPE    ,XTSTN          ;TYPE "TEST NO: "
 4915  004416  104407                          RDOCT                   ;GET NUMBER
```

```
 4916  004420  012600                        MOV     (SP)+,R0        ;
 4917  004422  001773                        BEQ     X1$             ;BR IF NO NUMBER INPUT
 4918  004424  020037  015140                CMP     R0,LASTN        ;VALID TEST?
 4919  004430  101370                        BHI     X1$             ;BR IF NO
 4920  004432  110037  001102                MOVB    R0,STSTNM       ;LOAD INITAL TEST NO.
 4921  004436  000241                        CLC                     ;
 4922  004440  006100                        ROL     R0              ;MAKE POWER OF 2.
 4923  004442  016037  015012  001106        MOV     TEST.TABLE(R0),SLPADR
 4924                                                                ;GET TEST PC.
 4925  004450  062737  000002  001106        ADD     #2,SLPADR       ;GET PAST SCOPE
 4926  004456  013737  001106  001110        MOV     SLPADR,SLPERR   ;
 4927  004464  000177  174416                JMP     @SLPADR         ;GOTO TEST.
 4928
 4935
   (3)                                  ;##################################################################
   (4)                                  ;#TEST 1          ROM "BREPLY" TEST
   (4)                                  ;#TEST TO REFERENCE ALL 256. ADDRESS
   (4)                                  ;#OF THE ROM ONLY MAKING SURE THAT
   (4)                                  ;#THERE IS A "BREPLY" RESPONSE - NO DATA IS
   (4)                                  ;#CHECKED IN THIS TEST.
   (3)                                  ;##################################################################
   (2)  004470  000004                  TST1:   SCOPE
 4936  004472  012700  173000                    MOV     #173000,R0      ;SET INITAL START ADDRESS OF ROM.
 4937  004476  012701  000400                    MOV     #256.,R1        ;SET NUMBER OF WORDS TO BE TESTED.
 4938  004502  013746  000004                    MOV     4,-(SP)         ;SAVE LOC 4.
 4939  004506  013746  000006                    MOV     6,-(SP)         ;SAVE LOC 6.
 4940  004512  012737  004544  000004            MOV     #3$,4           ;SET TIME-OUT TRAP VECTOR.
 4941  004520  012737  000200  000006            MOV     #200,6          ;LOAD PRIORITY = 4
 4942  004526  005710                1$:         TST     (R0)            ;REFERENCE ROM ADDRESS.
 4943  004530  005710                            TST     (R0)            ;DO IT AGAIN.
 4944  004532  062700  000002        2$:         ADD     #2,R0           ;UPDATE ADDRESS.
 4945  004536  005301                            DEC     R1              ;ALL DONE?
 4946  004540  001372                            BNE     1$              ;BR IF NO
 4947  004542  000404                            BR      4$              ;CONTINUE TEST
 4948  004544  104001                3$:         ERROR   1               ;ROM DID NOT ISSUE "BREPLY".
 4949  004546  012716  004532                    MOV     #2$,(SP)        ;SET RETURN ADDRESS
 4950  004552  000002                            RTI                     ;RETURN
 4951  004554  012637  000006        4$:         MOV     (SP)+,6         ;RESTORE ADD 6.
 4952  004560  012637  000004                    MOV     (SP)+,4         ;RESTORE ADD 4.
 4953
 4954
 4960
 4961                                   ;##################################################################
   (3)                                  ;#TEST 2          ROM WRITE TEST
   (4)                                  ;#TEST THAT WRITING THE ROM
   (4)                                  ;#PRODUCES A TIME-OUT TRAP.
   (4)                                  ;#
   (3)                                  ;##################################################################
   (2)  004564  000004                  TST2:   SCOPE
 4962  004566  012700  173000                    MOV     #173000,R0      ;SET ROM ADDRESS
 4963  004572  012701  000400                    MOV     #256.,R1        ;DO ALL ADDRESS
 4964  004576  013746  000004                    MOV     4,-(SP)         ;STORE 4
 4965  004602  013746  000006                    MOV     6,-(SP)         ;STORE 6
 4966  004606  012737  004632  000004            MOV     #2$,4           ;SET TRAP VECTOR
 4967  004614  012737  000200  000006            MOV     #200,6          ;PRIO.
 4968  004622  005020                1$:         CLR     (R0)+           ;WRITE ROM WITH ZERO.
 4969  004624  000240                            NOP                     ;
```

```
 4970  004626  104016                        ERROR   16              ;WRITE ROM AND NO TRAP.
 4971  004630  024646                        CMP     -(SP),-(SP)     ;FAKE AN INTERUPT.
 4972  004632  105301                2$:     DECB    R1              ;256 DONE?
 4973  004634  001403                        BEQ     3$              ;BR IF YES
 4974  004636  012716  004622                MOV     #1$,(SP)        ;SET RETURN
 4975  004642  000002                        RTI                     ;RETURN
 4976  004644  106427  000000        3$:     MTPS    #0              ;SET PTY TO 0
 4977  004650  022626                        CMP     (SP)+,(SP)+     ;FAKE RTI
 4978  004652  012637  000006                MOV     (SP)+,6         ;RESTORE 6
 4979  004656  012637  000004                MOV     (SP)+,4         ;RESTORE 4
 4980
 4987                          ;#############################################################
  (3)                          ;#TEST 3          ROM DATA TEST
  (4)                          ;#TEST TO READ AND COMPARE ALL
  (4)                          ;#256 ROM ADDRESS TO THE ROM DATA
  (4)                          ;#MAP IN MEMORY.  ALL ADDRESSES ARE
  (4)                          ;#VERIFIED FOR GOOD DATA.
  (3)                          ;#############################################################
  (2)  004662  000004          TST3:   SCOPE
 4988  004664  012700  173000                MOV     #173000,R0      ;SET START OF ROM ADDRESS.
 4989  004670  012701  000400                MOV     #256.,R1        ;SET NUMBER OF WORDS TO CHECK.
 4990  004674  012702  013516                MOV     #ROMMAP,R2      ;GET SOFTWARE ADDRESS
 4991  004700  011004                1$:     MOV     (R0),R4         ;READ ROM.
 4992  004702  011205                        MOV     (R2),R5         ;READ SOFTWARE IMAGE
 4993  004704  020504                        CMP     R5,R4           ;ARE THEY GOOD?
 4994  004706  001401                        BEQ     .+4             ;ROM DATA ERROR?
 4995  004710  104002                        ERROR   2               ;BAD DATA IN ROMS!!
 4996  004712  022022                        CMP     (R0)+,(R2)+     ;POP POINTERS
 4997  004714  005301                        DEC     R1              ;ALL DONE?
 4998  004716  001370                        BNE     1$              ;BR IF NOT DONE.
 4999
 5006
 5007                          ;#############################################################
  (3)                          ;#TEST 4          600 HZ ADDRESS TEST
  (4)                          ;#TEST TO VERIFY A "BREPLY" RESPONSE
  (4)                          ;#FROM THE 600 HZ CLOCK.
  (4)                          ;#IT IS ASSUMED THAT THE CLOCK
  (4)                          ;#IS AT ADDRESS "177546".
  (3)                          ;#############################################################
  (2)  004720  000004          TST4:   SCOPE
 5008  004722  013746  000004                MOV     4,-(SP)         ;SAVE ADDRESS 4.
 5009  004726  013746  000006                MOV     6,-(SP)         ;SAVE ADDRESS 6.
 5010  004732  012701  000017                MOV     #15.,R1         ;SET INTERNAL ICOUNT TO 15.
 5011  004736  013700  001376                MOV     CLKCSR,R0       ;GET CLOCK CSR.
 5012  004742  012737  004766  000004        MOV     #3$,4           ;SET TIME-OUT VECTOR.
 5013  004750  012737  000200  000006        MOV     #200,6          ;SET PRIO.
 5014  004756  005710                1$:     TST     (R0)            ;REFERENCE CLOCK CSR.
 5015  004760  005301                2$:     DEC     R1              ;ICOUNT = 0?
 5016  004762  001375                        BNE     1$              ;BR IF NO.
 5017  004764  000404                        BR      4$              ;CONT TEST
 5018  004766  104003                3$:     ERROR   3               ;TIME-OUT ERROR.
 5019  004770  012716  004760                MOV     #2$,(SP)        ;SET RETURN ADD.
 5020  004774  000002                        RTI                     ;RETURN.
 5021  004776  012637  000006        4$:     MOV     (SP)+,6         ;RESTORE 6
 5022  005002  012637  000004                MOV     (SP)+,4         ;RESTORE 4
 5023
```

```
                                ;#################################################################
                                ;#TEST 5          600 HZ "EVENT ENABLE" R/W TEST
                                ;#TEST THAT BIT6 OF 600HZ
                                ;#CLOCK IS R/W.
                                ;#SET PSW TO "4" AND SET BIT6
                                ;#OF 600HZ CLOCK VERIFYING THAT NO
                                ;#INTERRUPT OCCURED AND THAT BIT6
                                ;#DID INDEED SET,  THEN CLEAR BIT6
                                ;#AND VERIFY BIT6 IS CLEARED.
                                ;#VERIFY NO INTERUPT OCCURED.
                                ;#THEN CLEAR PS VERIFYING THAT
                                ;#THE INTERUPT PENDING OCCURES.
                                ;#
                                ;#################################################################
005006  000004                  TST5:   SCOPE
005010  012737  000005  001160          MOV     #5,STIMES       ;;DO 5 ITERATIONS
005016  106427  000200                  MTPS    #200            ;SET PRIO TO "4"
005022  013700  001376                  MOV     CLKCSR,R0       ;SET CLOCK CSR
005026  017746  174346                  MOV     @CLKVEC,-(SP)   ;SAVE CLKVEC
005032  017746  174344                  MOV     @CLKPTY,-(SP)   ;SAVE CLKPTY
005036  012777  005132  174334          MOV     #3$,@CLKVEC     ;SET CLOCK VECTOR
005044  012777  000200  174330          MOV     #200,@CLKPTY    ;SET CLOCK PRIORITY
005052  012701  000017                  MOV     #15.,R1         ;SET ICOUNT
005056  012705  000100          1$:     MOV     #BIT6,R5        ;SET EXPECTED
005062  010510                          MOV     R5,(R0)         ;WRITE BIT6
005064  011004                          MOV     (R0),R4         ;READ CSR
005066  020504                          CMP     R5,R4           ;COMPARE
005070  001401                          BEQ     .+4             ;DID BIT6 SET?
005072  104004                          ERROR   4               ;BIT6 NOT SET!
005074  012727  007640                  MOV     #4000.,(PC)+    ;DELAY TIME
005100  000000          65$:    0                               ;STORAGE
005102  005337  005100                  DEC     65$             ;DELAY
005106  001375                          BNE     .-4
005110  040510                          BIC     R5,(R0)         ;CLEAR BIT6
005112  011004                          MOV     (R0),R4         ;READ CSR
005114  005005                          CLR     R5              ;CLEAR EXPECTED
005116  005704                          TST     R4              ;IS R4=0?
005120  001401                          BEQ     .+4             ;BR IF OK!
005122  104004                          ERROR   4               ;REGISTER NOT=0!
005124  005301          2$:     DEC     R1                      ;I COUNT=0?
005126  001353                          BNE     1$              ;BR IF NO
005130  000410                          BR      4$              ;CONT TEST.
005132  010037  001156          3$:     MOV     R0,STMP0        ;SAVE R0
005136  104005                          ERROR   5               ;UNEXPECTED INTERRUPT
005140  013700  001156                  MOV     STMP0,R0        ;RESTORE R0
005144  012716  005124                  MOV     #2$,(SP)        ;SET RETURN ADDRESS
005150  000002                          RTI                     ;EXIT
005152  005005          4$:     CLR     R5
005154  012777  005204  174216          MOV     #5$,@CLKVEC
005162  012777  000200  174212          MOV     #200,@CLKPTY
005170  106427  000000                  MTPS    #0
005174  005205                          INC     R5
005176  001376                          BNE     .-2
005200  104006                          ERROR   6
005202  000401                          BR      6$
```

```
005204  022626                  5$:     CMP     (SP)+,(SP)+
005206  012677  174170          6$:     MOV     (SP)+,@CLKPTY   ;RESTORE CLKPTY
005212  012677  174162                  MOV     (SP)+,@CLKVEC   ;RESTORE CLKVEC
005216  106427  000000                  MTPS    #0              ;SET PRIO TO LVL 0

                        ;#################################################################
                        ;#TEST 6         600HZ INTERRUPT TEST
                        ;#TEST THAT THE 600HZ CLOCK
                        ;#CAN INTERRUPT AND THAT THE
                        ;#INTERRUPTS ARE WITHIN THE SAME
                        ;#TIME SPAN.  200. INTERRUPTS WILL BE
                        ;#EXECUTED
                        ;#################################################################
005222  000004         TST6:   SCOPE
005224  017746  174150          MOV     @CLKVEC,-(SP)   ;SAVE CLKVEC
005230  017746  174146          MOV     @CLKPTY,-(SP)   ;SAVE CLKPTY
005234  012777  005310  174136  MOV     #2$,@CLKVEC     ;SET INTERRUPT VECTOR
005242  012777  000200  174132  MOV     #200,@CLKPTY    ;SET PRIO.
005250  012704  000310          MOV     #200.,R4        ;SET ICOUNT
005254  012705  000002          MOV     #2,R5           ;SET COUNT POSITION
005260  005001                  CLR     R1              ;ZERO COUNT UP.
005262  106427  000000          MTPS    #0              ;CLEAR PSW
005266  013700  001376          MOV     CLKCSR,R0       ;SAVE FOR PRINTOUT
005272  052777  000100  174076  BIS     #BIT6,@CLKCSR   ;SET EVENT ENABLE
005300  005201                  1$:     INC     R1              ;COUNT TIME
005302  001376                  BNE     1$              ;=0?
005304  104006                  ERROR   6               ;NO INTERRUPT
005306  000423                  BR      5$              ;EXIT TEST
005310  005305                  2$:     DEC     R5              ;HOW MANY INTERRUPTS?
005312  001414                  BEQ     3$              ;BR IF 2ND!
005314  100014                  BPL     4$              ;BR IF 1ST!
005316  010102                  MOV     R1,R2           ;MUST BE 3RD OR MORE!
005320  062702  000010          ADD     #10,R2          ;GIVE TOLARENCE OF +10
005324  160302                  SUB     R3,R2           ;GET DIFFERENCE
005326  100001                  BPL     .+4             ;SKIP IF POSITIVE
005330  005402                  NEG     R2              ;MAKE POSITIVE
005332  020227  000270          CMP     R2,#270         ;GIVE +270 TOLARENCE
005336  101401                  BLOS    .+4
005340  104007                  ERROR   7               ;INTERRUPTS OUT OF TOLERANCE
005342  000401                  BR      4$              ;CONT TEST
005344  010103                  3$:     MOV     R1,R3           ;SAVE COUNT
005346  005001                  4$:     CLR     R1              ;ZERO COUNTER
005350  005304                  DEC     R4              ;ICOUNT =0?
005352  001401                  BEQ     5$              ;BR IF YES
005354  000002                  RTI                     ;EXIT
005356  042777  000100  174012  5$:     BIC     #BIT6,@CLKCSR   ;CLEAR EVENT ENABLE
005364  022626                  CMP     (SP)+,(SP)+     ;POP INTR OFF STACK
005366  012677  174010          MOV     (SP)+,@CLKPTY   ;RESTORE CLKPTY
005372  012677  174002          MOV     (SP)+,@CLKVEC   ;RESTORE CLKVEC.
005376  106427  000000          MTPS    #0              ;SET PRIO TO LEVEL 0

                        ;#################################################################
                        ;#TEST 7         UNEXPECTED "BREPLY" TEST.
                        ;#TEST TO SCAN THROUGH ALL ADDRESS
```

```
                                        ;#VERIFYING THAT ONLY EXPECTED DEVICES
                                        ;#RETURN "BREPLY".  THIS TEST CHECKS
                                        ;#ONLY THAT "UNEXPECTED  BREPLYS" AREN'T RECEIVED
                                        ;#NOT THAT ALL DEVICE ISSUE "BREPLY".
                                        ;#
                                        ;#################################################################
005402  000004                  TST7:   SCOPE
005404  012737  000003  001160          MOV     #3,$TIMES       ;;DO 3 ITERATIONS
005412  013746  000004                  MOV     4,-(SP)         ;SAVE 4
005416  013746  000006                  MOV     6,-(SP)         ;SAVE 6
005422  012737  005444  000004          MOV     #2$,4           ;SET TIME-OUT TRAP
005430  005037  000006                  CLR     6               ;
005434  005000                          CLR     R0              ;SET FIRST ADDRESS TO 0
005436  005710                  1$:     TST     (R0)            ;READ ADDRESS
005440  005720                          TST     (R0)+           ;POP ADDRESS
005442  000775                          BR      1$              ;CONTINUE TEST
005444  022626                  2$:     CMP     (SP)+,(SP)+     ;FAKE AN RTI
005446  023700  001374                  CMP     MEMSIZ,R0       ;16K ?
005452  001403                          BEQ     .+10            ;BR IF MEMORY SIZE IS CORRECT
005454  104016                          ERROR   16              ;INCORRECT MEMORY SIZE FOUND
005456  104400  002277                  TYPE    ,EM21           ;REPORT CHANGE MSG.
                                                                ;LOC SAVR0 HAS MEMORY FOUND.
005462  012737  005520  000004          MOV     #6$,4           ;RESET TIME-OUT TRAP
005470  005710                  3$:     TST     (R0)
005472  012701  014650                  MOV     #BREPLY,TABLE,R1 ;SET EXPECTED DEVICES
005476  022100                  4$:     CMP     (R1)+,R0        ;EXPECTED DEVICE?
005500  001002                          BNE     5$              ;BR IF NO
005502  062100                          ADD     (R1)+,R0        ;POP MODULO OFFSET
005504  000771                          BR      3$              ;CONT TEST
005506  005721                  5$:     TST     (R1)+           ;POP PAST OFFSET
005510  005711                          TST     (R1)            ;END OF TABLE?
005512  001371                          BNE     4$              ;BR IF NO
005514  104014                          ERROR   14              ;UNEXPECTED "BREPLY" FROM DEVICE.
005516  000401                          BR      7$              ;SKIP FAKE RTI
005520  022626                  6$:     CMP     (SP)+,(SP)+     ;FAKE RTI
005522  062700  000002          7$:     ADD     #2,R0           ;POP TO NEXT ADDRESS
005526  022700  177700                  CMP     #177700,R0      ;ALL DONE?
005532  001356                          BNE     3$
005534  012637  000006                  MOV     (SP)+,6         ;RESTOE 6
005540  012637  000004                  MOV     (SP)+,4 ;RESTORE 4
                                        ;#################################################################
                                        ;#TEST 10       "DEVICE 'A'" ADDRESS TEST
                                        ;#TEST TO VERIFY A "BREPLY" RESPONSE
                                        ;#FROM "DEVICE'A'".
                                        ;#IT IS ASSUMED THAT THE DEVICE
                                        ;#IS AT ADDRESS "172524".
                                        ;#################################################################
005544  000004                  TST10:  SCOPE
005546  013746  000004                  MOV     4,-(SP)         ;SAVE ADDRESS 4.
005552  013746  000006                  MOV     6,-(SP)         ;SAVE ADDRESS 6.
005556  012701  000017                  MOV     #15.,R1         ;SET INTERNAL ICOUNT TO 15.
005562  013700  001404                  MOV     DEV.A,R0        ;GET DEVICE CSR.
005566  012737  005612  000004          MOV     #3$,4           ;SET TIME-OUT VECTOR.
005574  012737  000200  000006          MOV     #200,6          ;SET PRIO.
005602  005710                  1$:     TST     (R0)            ;REFERENCE DEVICE CSR.
```

```
 5186  005604  005301                    2$:     DEC     R1              ;ICOUNT = 0?
 5187  005606  001375                            BNE     1$              ;BR IF NO
 5188  005610  000404                            BR      4$              ;CONT TEST
 5189  005612  104011                    3$:     ERROR   11              ;TIME-OUT ERROR.
 5190  005614  012716  005604                    MOV     #2$,(SP)        ;SET RETURN ADD.
 5191  005620  000002                            RTI                     ;RETURN.
 5192  005622  012637  000006            4$:     MOV     (SP)+,6         ;RESTORE 6
 5193  005626  012637  000004                    MOV     (SP)+,4         ;RESTORE 4
 5194
 5217
   (5)                                  ;*****************************************************************
   (4)                                  ;*TEST 11       R/W TEST OF BIT 0 IN DEVICE 'A'
   (5)                                  ;*R/W TEST OF BIT 0 IN DEVICE 'A'.
   (5)                                  ;*WRITE BIT 0 VERIFY ONLY BIT 0 IS SET.
   (5)                                  ;*CLEAR BIT 0 VERIFY BIT 0 IS CLEARED.
   (5)                                  ;*
   (4)                                  ;*****************************************************************
   (3)  005632  000004                  TST11:  SCOPE
   (1)  005634  013700  001404                  MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
   (1)  005640  012705  000001                  MOV     #BIT0,R5        ;SET EXPECTED.
   (1)  005644  010510                          MOV     R5,(R0)         ;WRITE BIT0
   (1)  005646  011004                          MOV     (R0),R4         ;READ CSR
   (1)  005650  020504                          CMP     R5,R4           ;BIT OK?
   (2)  005652  001401                          BEQ     .+4             ;BR IF OK
   (1)  005654  104010                          ERROR   10              ;REGISTER HAS WRONG DATA.
   (1)  005656  040510                          BIC     R5,(R0)         ;CLEAR BIT0
   (1)  005660  011004                          MOV     (R0),R4         ;READ REGISTER.
   (1)  005662  042705  000001                  BIC     #BIT0,R5        ;ZERO EXPECTED
   (1)  005666  005704                          TST     R4              ;REGISTER OK?
   (2)  005670  001401                          BEQ     .+4             ;BR IF YES
   (1)  005672  104010                          ERROR   10              ;REGISTER NOT =0.
   (1)
   (5)                                  ;*****************************************************************
   (4)                                  ;*TEST 12       R/W TEST OF BIT 1 IN DEVICE 'A'
   (5)                                  ;*R/W TEST OF BIT 1 IN DEVICE 'A'
   (5)                                  ;*WRITE BIT 1 VERIFY ONLY BIT 1 IS SET.
   (5)                                  ;*CLEAR BIT 1 VERIFY BIT 1 IS CLEARED.
   (5)                                  ;*
   (4)                                  ;*****************************************************************
   (3)  005674  000004                  TST12:  SCOPE
   (1)  005676  013700  001404                  MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
   (1)  005702  012705  000002                  MOV     #BIT1,R5        ;SET EXPECTED.
   (1)  005706  010510                          MOV     R5,(R0)         ;WRITE BIT1
   (1)  005710  011004                          MOV     (R0),R4         ;READ CSR
   (1)  005712  020504                          CMP     R5,R4           ;BIT OK?
   (2)  005714  001401                          BEQ     .+4             ;BR IF OK
   (1)  005716  104010                          ERROR   10              ;REGISTER HAS WRONG DATA.
   (1)  005720  040510                          BIC     R5,(R0)         ;CLEAR BIT1
   (1)  005722  011004                          MOV     (R0),R4         ;READ REGISTER.
   (1)  005724  042705  000002                  BIC     #BIT1,R5        ;ZERO EXPECTED
   (1)  005730  005704                          TST     R4              ;REGISTER OK?
   (2)  005732  001401                          BEQ     .+4             ;BR IF YES
   (1)  005734  104010                          ERROR   10              ;REGISTER NOT =0.
   (1)
   (5)                                  ;*****************************************************************
   (4)                                  ;*TEST 13       R/W TEST OF BIT 2 IN DEVICE 'A'
```

```
(5)                                     ;*R/W TEST OF BIT 2 IN DEVICE 'A'.
(5)                                     ;*WRITE BIT 2 VERIFY ONLY BIT 2 IS SET.
(5)                                     ;*CLEAR BIT 2 VERIFY BIT 2 IS CLEARED.
(5)                                     ;*
(4)                                     ;*********************************************************************
(3)  005736  000004                     TST13:  SCOPE
(1)  005740  013700  001404                     MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  005744  012705  000004                     MOV     #BIT2,R5        ;SET EXPECTED.
(1)  005750  010510                             MOV     R5,(R0)         ;WRITE BIT2
(1)  005752  011004                             MOV     (R0),R4         ;READ CSR
(1)  005754  020504                             CMP     R5,R4           ;BIT OK?
(2)  005756  001401                             BEQ     .+4             ;BR IF OK
(1)  005760  104010                             ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  005762  040510                             BIC     R5,(R0)         ;CLEAR BIT2
(1)  005764  011004                             MOV     (R0),R4         ;READ REGISTER.
(1)  005766  042705  000004                     BIC     #BIT2,R5        ;ZERO EXPECTED
(1)  005772  005704                             TST     R4              ;REGISTER OK?
(2)  005774  001401                             BEQ     .+4             ;BR IF YES
(1)  005776  104010                             ERROR   10              ;REGISTER NOT =0.
(1)
(5)                                     ;*********************************************************************
(4)                                     ;*TEST 14        R/W TEST OF BIT 3 IN DEVICE 'A'
(5)                                     ;*R/W TEST OF BIT 3 IN DEVICE 'A'.
(5)                                     ;*WRITE BIT 3 VERIFY ONLY BIT 3 IS SET.
(5)                                     ;*CLEAR BIT 3 VERIFY BIT 3 IS CLEARED.
(5)                                     ;*
(4)                                     ;*********************************************************************
(3)  006000  000004                     TST14:  SCOPE
(1)  006002  013700  001404                     MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  006006  012705  000010                     MOV     #BIT3,R5        ;SET EXPECTED.
(1)  006012  010510                             MOV     R5,(R0)         ;WRITE BIT3
(1)  006014  011004                             MOV     (R0),R4         ;READ CSR
(1)  006016  020504                             CMP     R5,R4           ;BIT OK?
(2)  006020  001401                             BEQ     .+4             ;BR IF OK
(1)  006022  104010                             ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  006024  040510                             BIC     R5,(R0)         ;CLEAR BIT3
(1)  006026  011004                             MOV     (R0),R4         ;READ REGISTER.
(1)  006030  042705  000010                     BIC     #BIT3,R5        ;ZERO EXPECTED
(1)  006034  005704                             TST     R4              ;REGISTER OK?
(2)  006036  001401                             BEQ     .+4             ;BR IF YES
(1)  006040  104010                             ERROR   10              ;REGISTER NOT =0.
(1)
(5)                                     ;*********************************************************************
(4)                                     ;*TEST 15        R/W TEST OF BIT 4 IN DEVICE 'A'
(5)                                     ;*R/W TEST OF BIT 4 IN DEVICE 'A'.
(5)                                     ;*WRITE BIT 4 VERIFY ONLY BIT 4 IS SET.
(5)                                     ;*CLEAR BIT 4 VERIFY BIT 4 IS CLEARED.
(5)                                     ;*
(4)                                     ;*********************************************************************
(3)  006042  000004                     TST15:  SCOPE
(1)  006044  013700  001404                     MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  006050  012705  000020                     MOV     #BIT4,R5        ;SET EXPECTED.
(1)  006054  010510                             MOV     R5,(R0)         ;WRITE BIT4
(1)  006056  011004                             MOV     (R0),R4         ;READ CSR
(1)  006060  020504                             CMP     R5,R4           ;BIT OK?
(2)  006062  001401                             BEQ     .+4             ;BR IF OK
```

```
(1)  006064  104010                    ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  006066  040510                    BIC     R5,(R0)         ;CLEAR BIT4
(1)  006070  011004                    MOV     (R0),R4         ;READ REGISTER.
(1)  006072  042705  000020            BIC     #BIT4,R5        ;ZERO EXPECTED
(1)  006076  005704                    TST     R4              ;REGISTER OK?
(2)  006100  001401                    BEQ     .+4             ;BR IF YES
(1)  006102  104010                    ERROR   10              ;REGISTER NOT =0.
(1)
(5)                         ;##################################################################
(4)                         ;#TEST 16       R/W TEST OF BIT 5 IN DEVICE 'A'
(5)                         ;#R/W TEST OF BIT 5 IN DEVICE 'A'.
(5)                         ;#WRITE BIT 5 VERIFY ONLY BIT 5 IS SET.
(5)                         ;#CLEAR BIT 5 VERIFY BIT 5 IS CLEARED.
(5)                         ;#
(4)                         ;##################################################################
(3)  006104  000004         TST16:  SCOPE
(1)  006106  013700  001404            MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  006112  012705  000040            MOV     #BIT5,R5        ;SET EXPECTED.
(1)  006116  010510                    MOV     R5,(R0)         ;WRITE BIT5
(1)  006120  011004                    MOV     (R0),R4         ;READ CSR
(1)  006122  020504                    CMP     R5,R4           ;BIT OK?
(2)  006124  001401                    BEQ     .+4             ;BR IF OK
(1)  006126  104010                    ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  006130  040510                    BIC     R5,(R0)         ;CLEAR BIT5
(1)  006132  011004                    MOV     (R0),R4         ;READ REGISTER.
(1)  006134  042705  000040            BIC     #BIT5,R5        ;ZERO EXPECTED
(1)  006140  005704                    TST     R4              ;REGISTER OK?
(2)  006142  001401                    BEQ     .+4             ;BR IF YES
(1)  006144  104010                    ERROR   10              ;REGISTER NOT =0.
(1)
(5)                         ;##################################################################
(4)                         ;#TEST 17       R/W TEST OF BIT 6 IN DEVICE 'A'
(5)                         ;#R/W TEST OF BIT 6 IN DEVICE 'A'.
(5)                         ;#WRITE BIT 6 VERIFY ONLY BIT 6 IS SET.
(5)                         ;#CLEAR BIT 6 VERIFY BIT 6 IS CLEARED.
(5)                         ;#
(4)                         ;##################################################################
(3)  006146  000004         TST17:  SCOPE
(1)  006150  013700  001404            MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  006154  012705  000100            MOV     #BIT6,R5        ;SET EXPECTED.
(1)  006160  010510                    MOV     R5,(R0)         ;WRITE BIT6
(1)  006162  011004                    MOV     (R0),R4         ;READ CSR
(1)  006164  020504                    CMP     R5,R4           ;BIT OK?
(2)  006166  001401                    BEQ     .+4             ;BR IF OK
(1)  006170  104010                    ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  006172  040510                    BIC     R5,(R0)         ;CLEAR BIT6
(1)  006174  011004                    MOV     (R0),R4         ;READ REGISTER.
(1)  006176  042705  000100            BIC     #BIT6,R5        ;ZERO EXPECTED
(1)  006202  005704                    TST     R4              ;REGISTER OK?
(2)  006204  001401                    BEQ     .+4             ;BR IF YES
(1)  006206  104010                    ERROR   10              ;REGISTER NOT =0.
(1)
(5)                         ;##################################################################
(4)                         ;#TEST 20       R/W TEST OF BIT 7 IN DEVICE 'A'
(5)                         ;#R/W TEST OF BIT 7 IN DEVICE 'A'.
(5)                         ;#WRITE BIT 7 VERIFY ONLY BIT 7 IS SET.
```

```
(5)                                     ;#CLEAR BIT 7 VERIFY BIT 7 IS CLEARED.
(5)                                     ;#
(4)                                     ;##################################################################
(3)  006210  000004                     TST20:  SCOPE
(1)  006212  013700  001404                     MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  006216  012705  000200                     MOV     #BIT7,R5        ;SET EXPECTED.
(1)  006222  010510                             MOV     R5,(R0)         ;WRITE BIT7
(1)  006224  011004                             MOV     (R0),R4         ;READ CSR
(1)  006226  020504                             CMP     R5,R4           ;BIT OK?
(2)  006230  001401                             BEQ     .+4             ;BR IF OK
(1)  006232  104010                             ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  006234  040510                             BIC     R5,(R0)         ;CLEAR BIT7
(1)  006236  011004                             MOV     (R0),R4         ;READ REGISTER.
(1)  006240  042705  000200                     BIC     #BIT7,R5        ;ZERO EXPECTED
(1)  006244  005704                             TST     R4              ;REGISTER OK?
(2)  006246  001401                             BEQ     .+4             ;BR IF YES
(1)  006250  104010                             ERROR   10              ;REGISTER NOT =0.
(1)
(5)                                     ;##################################################################
(4)                                     ;#TEST 21       R/W TEST OF BIT 8 IN DEVICE 'A'
(5)                                     ;#R/W TEST OF BIT 8 IN DEVICE 'A'.
(5)                                     ;#WRITE BIT 8 VERIFY ONLY BIT 8 IS SET.
(5)                                     ;#CLEAR BIT 8 VERIFY BIT 8 IS CLEARED.
(5)                                     ;#
(4)                                     ;##################################################################
(3)  006252  000004                     TST21:  SCOPE
(1)  006254  013700  001404                     MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  006260  012705  000400                     MOV     #BIT8,R5        ;SET EXPECTED.
(1)  006264  010510                             MOV     R5,(R0)         ;WRITE BIT8
(1)  006266  011004                             MOV     (R0),R4         ;READ CSR
(1)  006270  020504                             CMP     R5,R4           ;BIT OK?
(2)  006272  001401                             BEQ     .+4             ;BR IF OK
(1)  006274  104010                             ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  006276  040510                             BIC     R5,(R0)         ;CLEAR BIT8
(1)  006300  011004                             MOV     (R0),R4         ;READ REGISTER.
(1)  006302  042705  000400                     BIC     #BIT8,R5        ;ZERO EXPECTED
(1)  006306  005704                             TST     R4              ;REGISTER OK?
(2)  006310  001401                             BEQ     .+4             ;BR IF YES
(1)  006312  104010                             ERROR   10              ;REGISTER NOT =0.
(1)
(5)                                     ;##################################################################
(4)                                     ;#TEST 22       R/W TEST OF BIT 9 IN DEVICE 'A'
(5)                                     ;#R/W TEST OF BIT 9 IN DEVICE 'A'.
(5)                                     ;#WRITE BIT 9 VERIFY ONLY BIT 9 IS SET.
(5)                                     ;#CLEAR BIT 9 VERIFY BIT 9 IS CLEARED.
(5)                                     ;#
(4)                                     ;##################################################################
(3)  006314  000004                     TST22:  SCOPE
(1)  006316  013700  001404                     MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  006322  012705  001000                     MOV     #BIT9,R5        ;SET EXPECTED.
(1)  006326  010510                             MOV     R5,(R0)         ;WRITE BIT9
(1)  006330  011004                             MOV     (R0),R4         ;READ CSR
(1)  006332  020504                             CMP     R5,R4           ;BIT OK?
(2)  006334  001401                             BEQ     .+4             ;BR IF OK
(1)  006336  104010                             ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  006340  040510                             BIC     R5,(R0)         ;CLEAR BIT9
```

```
(1)  006342  011004                    MOV     (R0),R4         ;READ REGISTER.
(1)  006344  042705  001000            BIC     #BIT9,R5        ;ZERO EXPECTED
(1)  006350  005704                    TST     R4              ;REGISTER OK?
(2)  006352  001401                    BEQ     .+4             ;BR IF YES
(1)  006354  104010                    ERROR   10              ;REGISTER NOT =0.
(1)
(5)                             ;##################################################################
(4)                             ;*TEST 23       R/W TEST OF BIT 10 IN DEVICE 'A'
(5)                             ;*R/W TEST OF BIT 10 IN DEVICE 'A'.
(5)                             ;*WRITE BIT 10 VERIFY ONLY BIT 10 IS SET.
(5)                             ;*CLEAR BIT 10 VERIFY BIT 10 IS CLEARED.
(5)                             ;*
(4)                             ;##################################################################
(3)  006356  000004             TST23:  SCOPE
(1)  006360  013700  001404             MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  006364  012705  002000             MOV     #BIT10,R5       ;SET EXPECTED.
(1)  006370  010510                     MOV     R5,(R0)         ;WRITE BIT10
(1)  006372  011004                     MOV     (R0),R4         ;READ CSR
(1)  006374  020504                     CMP     R5,R4           ;BIT OK?
(2)  006376  001401                     BEQ     .+4             ;BR IF OK
(1)  006400  104010                     ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  006402  040510                     BIC     R5,(R0)         ;CLEAR BIT10
(1)  006404  011004                     MOV     (R0),R4         ;READ REGISTER.
(1)  006406  042705  002000             BIC     #BIT10,R5       ;ZERO EXPECTED
(1)  006412  005704                     TST     R4              ;REGISTER OK?
(2)  006414  001401                     BEQ     .+4             ;BR IF YES
(1)  006416  104010                     ERROR   10              ;REGISTER NOT =0.
(1)
(5)                             ;##################################################################
(4)                             ;*TEST 24       R/W TEST OF BIT 11 IN DEVICE 'A'
(5)                             ;*R/W TEST OF BIT 11 IN DEVICE 'A'.
(5)                             ;*WRITE BIT 11 VERIFY ONLY BIT 11 IS SET.
(5)                             ;*CLEAR BIT 11 VERIFY BIT 11 IS CLEARED.
(5)                             ;*
(4)                             ;##################################################################
(3)  006420  000004             TST24:  SCOPE
(1)  006422  013700  001404             MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  006426  012705  004000             MOV     #BIT11,R5       ;SET EXPECTED.
(1)  006432  010510                     MOV     R5,(R0)         ;WRITE BIT11
(1)  006434  011004                     MOV     (R0),R4         ;READ CSR
(1)  006436  020504                     CMP     R5,R4           ;BIT OK?
(2)  006440  001401                     BEQ     .+4             ;BR IF OK
(1)  006442  104010                     ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  006444  040510                     BIC     R5,(R0)         ;CLEAR BIT11
(1)  006446  011004                     MOV     (R0),R4         ;READ REGISTER.
(1)  006450  042705  004000             BIC     #BIT11,R5       ;ZERO EXPECTED
(1)  006454  005704                     TST     R4              ;REGISTER OK?
(2)  006456  001401                     BEQ     .+4             ;BR IF YES
(1)  006460  104010                     ERROR   10              ;REGISTER NOT =0.
(1)
(5)                             ;##################################################################
(4)                             ;*TEST 25       R/W TEST OF BIT 12 IN DEVICE 'A'
(5)                             ;*R/W TEST OF BIT 12 IN DEVICE 'A'.
(5)                             ;*WRITE BIT 12 VERIFY ONLY BIT 12 IS SET.
(5)                             ;*CLEAR BIT 12 VERIFY BIT 12 IS CLEARED.
(5)                             ;*
```

```
(4)
(3)  006462  000004                ;#########################################################################
(1)  006464  013700  001404  TST25: SCOPE
(1)  006470  012705  010000         MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  006474  010510                 MOV     #BIT12,R5       ;SET EXPECTED.
(1)  006476  011004                 MOV     R5,(R0)         ;WRITE BIT12
(1)  006500  020504                 MOV     (R0),R4         ;READ CSR
(2)  006502  001401                 CMP     R5,R4           ;BIT OK?
(1)  006504  104010                 BEQ     .+4             ;BR IF OK
(1)  006506  040510                 ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  006510  011004                 BIC     R5,(R0)         ;CLEAR BIT12
(1)  006512  042705  010000         MOV     (R0),R4         ;READ REGISTER.
(1)  006516  005704                 BIC     #BIT12,R5       ;ZERO EXPECTED
(2)  006520  001401                 TST     R4              ;REGISTER OK?
(1)  006522  104010                 BEQ     .+4             ;BR IF YES
(1)                                 ERROR   10              ;REGISTER NOT =0.
(5)
(4)                         ;#########################################################################
                            ;#TEST 26        R/W TEST OF BIT 13 IN DEVICE 'A'
                            ;#R/W TEST OF BIT 13 IN DEVICE 'A'.
                            ;#WRITE BIT 13 VERIFY ONLY BIT 13 IS SET.
                            ;#CLEAR BIT 13 VERIFY BIT 13 IS CLEARED.
                            ;#
                            ;#########################################################################
(3)  006524  000004  TST26: SCOPE
(1)  006526  013700  001404         MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  006532  012705  020000         MOV     #BIT13,R5       ;SET EXPECTED.
(1)  006536  010510                 MOV     R5,(R0)         ;WRITE BIT13
(1)  006540  011004                 MOV     (R0),R4         ;READ CSR
(1)  006542  020504                 CMP     R5,R4           ;BIT OK?
(2)  006544  001401                 BEQ     .+4             ;BR IF OK
(1)  006546  104010                 ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  006550  040510                 BIC     R5,(R0)         ;CLEAR BIT13
(1)  006552  011004                 MOV     (R0),R4         ;READ REGISTER.
(1)  006554  042705  020000         BIC     #BIT13,R5       ;ZERO EXPECTED
(1)  006560  005704                 TST     R4              ;REGISTER OK?
(2)  006562  001401                 BEQ     .+4             ;BR IF YES
(1)  006564  104010                 ERROR   10              ;REGISTER NOT =0.
(1)
(5)                         ;#########################################################################
(4)                         ;#TEST 27        R/W TEST OF BIT 14 IN DEVICE 'A'
(5)                         ;#R/W TEST OF BIT 14 IN DEVICE 'A'.
(5)                         ;#WRITE BIT 14 VERIFY ONLY BIT 14 IS SET.
(5)                         ;#CLEAR BIT 14 VERIFY BIT 14 IS CLEARED.
(5)                         ;#
(4)                         ;#########################################################################
(3)  006566  000004  TST27: SCOPE
(1)  006570  013700  001404         MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
(1)  006574  012705  040000         MOV     #BIT14,R5       ;SET EXPECTED.
(1)  006600  010510                 MOV     R5,(R0)         ;WRITE BIT14
(1)  006602  011004                 MOV     (R0),R4         ;READ CSR
(1)  006604  020504                 CMP     R5,R4           ;BIT OK?
(2)  006606  001401                 BEQ     .+4             ;BR IF OK
(1)  006610  104010                 ERROR   10              ;REGISTER HAS WRONG DATA.
(1)  006612  040510                 BIC     R5,(R0)         ;CLEAR BIT14
(1)  006614  011004                 MOV     (R0),R4         ;READ REGISTER.
(1)  006616  042705  040000         BIC     #BIT14,R5       ;ZERO EXPECTED
```

```
    (1)  006622  005704                        TST     R4              ;REGISTER OK?
    (2)  006624  001401                        BEQ     .+4             ;BR IF YES
    (1)  006626  104010                        ERROR   10              ;REGISTER NOT =0.
    (1)
    (5)                             ;#################################################################
    (4)                             ;#TEST 30       R/W TEST OF BIT 15 IN DEVICE 'A'
    (5)                             ;#R/W TEST OF BIT 15 IN DEVICE 'A'
    (5)                             ;#WRITE BIT 15 VERIFY ONLY BIT 15 IS SET.
    (5)                             ;#CLEAR BIT 15 VERIFY BIT 15 IS CLEARED.
    (5)                             ;#
    (4)                             ;#################################################################
    (3)  006630  000004             TST30:  SCOPE
    (1)  006632  013700  001404             MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
    (1)  006636  012705  100000             MOV     #BIT15,R5       ;SET EXPECTED.
    (1)  006642  010510                      MOV     R5,(R0)         ;WRITE BIT15
    (1)  006644  011004                      MOV     (R0),R4         ;READ CSR
    (1)  006646  020504                      CMP     R5,R4           ;BIT OK?
    (2)  006650  001401                      BEQ     .+4             ;BR IF OK
    (1)  006652  104010                      ERROR   10              ;REGISTER HAS WRONG DATA.
    (1)  006654  040510                      BIC     R5,(R0)         ;CLEAR BIT15
    (1)  006656  011004                      MOV     (R0),R4         ;READ REGISTER.
    (1)  006660  042705  100000             BIC     #BIT15,R5       ;ZERO EXPECTED
    (1)  006664  005704                      TST     R4              ;REGISTER OK?
    (2)  006666  001401                      BEQ     .+4             ;BR IF YES
    (1)  006670  104010                      ERROR   10              ;REGISTER NOT =0.
5218
5224
5225                                ;#################################################################
    (3)                             ;#TEST 31       BINARY COUNT TEST
    (4)                             ;#TEST TO WRITE A BINARY COUNT PATTERN
    (4)                             ;#(000000-177777) THRU DEVICE 'A' REGISTER
    (4)                             ;#
    (3)                             ;#################################################################
    (2)  006672  000004             TST31:  SCOPE
    (1)  006674  012737  000003  001160             MOV     #3,STIMES       ;;DO 3 ITERATIONS
5226 006702  013700  001404                      MOV     DEV.A,R0        ;GET DEVICE 'A' CSR.
5227 006706  005005                      CLR     R5              ;ZERO EXPECTED
5228 006710  010510                 1$:  MOV     R5,(R0)         ;LOAD DATA
5229 006712  011004                      MOV     (R0),R4         ;READ DATA.
5230 006714  020504                      CMP     R5,R4           ;WAS DATA GOOD?
5231 006716  001401                      BEQ     .+4             ;BR IF YES
5232 006720  104010                      ERROR   10              ;REGISTER DATA ERROR.
5233 006722  005205                      INC     R5              ;UPADE DATA
5234 006724  001371                      BNE     1$              ;BR IF NOT DONE
5235
5241
5242                                ;#################################################################
    (3)                             ;#TEST 32       1'S AND 0'S TEST
    (4)                             ;#TEST TO WRITE ALTERNATE 1'S AND 0'S
    (4)                             ;#(125252 AND 052525) INTO DEVICE 'A'.
    (4)                             ;#
    (3)                             ;#################################################################
    (2)  006726  000004             TST32:  SCOPE
    (1)  006730  012737  000012  001160             MOV     #10.,STIMES     ;;DO 10. ITERATIONS
5243 006736  013700  001404                      MOV     DEV.A,R0        ;GET DEVICE 'A' CSR
5244 006742  005003                      CLR     R3              ;SET TO DO 128. TIMES
```

E03


```
006744  012705  125252             MOV     #125252,R5      ;LOAD DATA
006750  010510             1$:     MOV     R5,(R0)         ;WRITE DATA
006752  011004                     MOV     (R0),R4         ;READ REGISTER
006754  020504                     CMP     R5,R4           ;DATA OK?
006756  001401                     BEQ     .+4             ;BR IF OK?
006760  104010                     ERROR   10              ;REGISTER R/W ERROR
006762  005105                     COM     R5              ;CHANGE DATA
006764  105203                     INCB    R3              ;DONE 128. TIMES?
006766  100370                     BPL     1$              ;BR IF NO.



                           ;*****************************************************
                           ;*TEST 33        DEVICE 'A' BYTE OPERATIONS
                           ;*TEST OF DEVICE 'A' BYTE OPERATIONS
                           ;*SET LOW BYTE OF DEVICE 'A' TO 0 AND
                           ;*SET HIGH BYTE OF DEVICE 'A' TO 3 THEN
                           ;*DO A 'INCB' ON LOW BYTE CHECK RESULTS
                           ;*DO A 'INCB' ON HIBH BYTE CHECK RESULTS
                           ;*CONTINUE TILL LOW BYTE GOES TO ZERO
                           ;*****************************************************
006770  000004             TST33:  SCOPE
006772  013700  001404             MOV     DEV.A,R0        ;LOAD DEVICE 'A' CSR.
006776  012710  001400             MOV     #3*400,(R0)     ;PLACE 3 IN HIGH BYTE
007002  005002                     CLR     R2              ;ZERO LOW BYTE IMAGE
007004  012703  000003             MOV     #3,R3           ;SET HIGH BYTE IMAGE TO 3.
007010  010001                     MOV     R0,R1           ;GET CSR
007012  005201                     INC     R1              ;MAKE EQUAL TO HIGH BYTE.
007014  105210             1$:     INCB    (R0)            ;ALTER LOW BYTE
007016  111004                     MOVB    (R0),R4         ;READ IT
007020  105202                     INCB    R2              ;ALTER LOW BYTE IMAGE
007022  110205                     MOVB    R2,R5           ;READ GOOD DATA
007024  020504                     CMP     R5,R4           ;IS DATA OK?
007026  001401                     BEQ     .+4             ;BR IF YES
007030  104010                     ERROR   10              ;REGISTER ERROR
007032  011004                     MOV     (R0),R4         ;READ WORD
007034  110237  001156             MOVB    R2,$TMP0        ;GET LOW BYTE
007040  110337  001157             MOVB    R3,$TMP0+1      ;GET HIGH BYTE
007044  013705  001156             MOV     $TMP0,R5        ;GOOD DATA
007050  020504                     CMP     R5,R4           ;STILL OK?
007052  001401                     BEQ     .+4
007054  104010                     ERROR   10              ;COMPARISON ERROR
007056  105705                     TSTB    R5              ;IF LOW BYTE =0 THEN DONE
007060  001410                     BEQ     2$              ;BR IF END OF TEST
007062  105211                     INCB    (R1)            ;ALTER HIGH BYTE
007064  111104                     MOVB    (R1),R4         ;READ IT
007066  105203                     INCB    R3              ;ALTER HIGH BYTE IMAGE
007070  110305                     MOVB    R3,R5           ;LOAD GOOD DATA
007072  020504                     CMP     R5,R4           ;DATA OK?
007074  001401                     BEQ     .+4             ;BR IF YES.
007076  104010                     ERROR   10              ;REGISTER ERROR
007100  000745                     BR      1$              ;CONT TEST
007102  000240             2$:     NOP



                           ;*****************************************************
                           ;*TEST 34        BUS INITIALIZE TEST FOR DEVICE 'A'
```

```
                                        ;#TEST THAT ALL BITS IN DEVICE 'A'
                                        ;#REGISTER CAN BE CLEARED BY "INIT" (RESET INSTR).
                                        ;#
                                        ;##################################################################
                                        TST34:  SCOPE
007106  012737  000005  001160                 MOV     #5,STIMES       ;;DO 5 ITERATIONS
007114  013700  001404                       MOV     DEV.A,R0        ;GET CSR
007120  012710  177777                       MOV     #-1,(R0)        ;SET ALL BITS
007124  005005                               CLR     R5              ;SET EXPECTED TO ALL ZEROS
007126  000005                               RESET                   ;ISSUE INIT
007130  052777  000100  172004               BIS     #100,@STKS      ;SET INTR ENABLE
007136  011004                               MOV     (R0),R4         ;READ REGISTER
007140  001401                               BEQ     .+4             ;BR IF ALL ZEROS
007142  104010                               ERROR   10              ;REGISTER NOT ALL ZEROS
```

```
                        ;#################################################################
                        ;#TEST 35        "DEVICE 'B'" ADDRESS TEST (SEL 0)
                        ;#TEST TO VERIFY A "BREPLY" RESPONSE
                        ;#FROM "DEVICE'B'" (SEL 0).
                        ;#IT IS ASSUMED THAT THE DEVICE
                        ;#IS AT ADDRESS "165250" (SEL 0).
                        ;#################################################################
007144  000004          TST35:  SCOPE
007146  013746  000004          MOV     4,-(SP)         ;SAVE ADDRESS 4.
007152  013746  000006          MOV     6,-(SP)         ;SAVE ADDRESS 6.
007156  012701  000017          MOV     #15.,R1         ;SET INTERNAL ICOUNT TO 15.
007162  013700  001406          MOV     DEV.B,R0        ;GET DEVICE CSR.
007166  012737  007212  000004  MOV     #3$,4           ;SET TIME-OUT VECTOR.
007174  012737  000200  000006  MOV     #200,6          ;SET PRIO.
007202  005710          1$:     TST     (R0)            ;REFERENCE DEVICE CSR.
007204  005301          2$:     DEC     R1              ;ICOUNT = 0?
007206  001375                  BNE     1$              ;BR IF NO.
007210  000404                  BR      4$              ;CONT TEST
007212  104012          3$:     ERROR   12              ;TIME-OUT ERROR.
007214  012716  007204          MOV     #2$,(SP)        ;SET RETURN ADD.
007220  000002                  RTI                     ;RETURN.
007222  012637  000006  4$:     MOV     (SP)+,6         ;RESTORE 6
007226  012637  000004          MOV     (SP)+,4         ;RESTORE 4

                        ;#################################################################
                        ;#TEST 36        R/W TEST OF BIT 6 IN DEVICE 'B' (SEL 0)
                        ;#TEST THAT BIT 6 IN DEVICE 'B' (SEL 0)
                        ;#IS R/W.
                        ;#SET BIT 6 VERIFY IT IS SET.
                        ;#CLEAR BIT 6 VERIFY IT IS CLEAR.
                        ;#
                        ;#################################################################
007232  000004          TST36:  SCOPE
007234  106427  000200          MTPS    #200
007240  013700  001406          MOV     DEV.B,R0        ;LOAD CSR INTO R0
007244  012705  000100          MOV     #BIT6,R5        ;SET BIT 6 INTO R5
007250  010510                  MOV     R5,(R0)         ;WRITE BIT
007252  011004                  MOV     (R0),R4         ;READ REGISTER
007254  020504                  CMP     R5,R4           ;OK?
007256  001401                  BEQ     .+4             ;BR IF OK.
007260  104013                  ERROR   13              ;REGISTER R/W ERROR
007262  040510                  BIC     R5,(R0)         ;CLEAR BIT 6
007264  011004                  MOV     (R0),R4         ;READ DEVICE
007266  042705  000100          BIC     #BIT6,R5        ;CLEAR EXPECTED
007272  005704                  TST     R4              ;REGISTER =0?
007274  001401                  BEQ     .+4             ;BR IF =0!
007276  104013                  ERROR   13              ;REGISTER ERROR

                        ;#################################################################
                        ;#TEST 37        R/W TEST OF BIT 7 IN DEVICE 'B' (SEL 0)
                        ;#TEST THAT BIT 7 IN DEVICE 'B' (SEL 0)
                        ;#IS R/W.
                        ;#SET BIT 7 VERIFY IT IS SET.
                        ;#CLEAR BIT 7 VERIFY IT IS CLEAR.
```

```
                                ;#
                                ;#######################################################################
                                TST37:  SCOPE
007300  000004                         
007302  106427  000200                  MTPS    #200
007306  013700  001406                  MOV     DEV.B,R0        ;LOAD CSR INTO R0
007312  012705  000200                  MOV     #BIT7,R5        ;SET BIT 7 INTO R5
007316  010510                          MOV     R5,(R0)         ;WRITE BIT
007320  011004                          MOV     (R0),R4         ;READ REGISTER
007322  020504                          CMP     R5,R4           ;OK?
007324  001401                          BEQ     .+4             ;BR IF OK.
007326  104013                          ERROR   13              ;REGISTER R/W ERROR
007330  040510                          BIC     R5,(R0)         ;CLEAR BIT 7
007332  011004                          MOV     (R0),R4         ;READ DEVICE
007334  042705  000200                  BIC     #BIT7,R5        ;CLEAR EXPECTED
007340  005704                          TST     R4              ;REGISTER =0?
007342  001401                          BEQ     .+4             ;BR IF =0!
007344  104013                          ERROR   13              ;REGISTER ERROR
5362                            
5371                            
5372                            ;#######################################################################
                                ;#TEST 40       DEVICE 'B' INTERRUPT TEST (SEL 0)
                                ;#TEST OF DEVICE 'B' INTERRUPTS (SEL 0)
                                ;#SET BIT6 VERIFY NO INTERRUPT
                                ;#CLEAR BIT6; SET BIT7 -VERIFY NO INTERRUPT
                                ;#NOW SET BIT6 AGAIN -EXPECT INTERUPT AT
                                ;#VECTOR "340".
                                ;#
                                ;#######################################################################
007346  000004                  TST40:  SCOPE
5373  007350  106427  000000            MTPS    #0
5374  007354  017746  172030            MOV     @#B.AVEC,-(SP)  ;SAVE DVB VECTOR
5375  007360  017746  172026            MOV     @#B.APTY,-(SP)  ;SAVE DVB PRIO
5376  007364  012777  007500  172016     MOV     #5$,@#B.AVEC    ;LOAD VECTOR
5377  007372  012777  000200  172012     MOV     #200,@#B.APTY   ;LOAD PRIO
5378  007400  005001                     CLR     R1
5379  007402  013700  001406            MOV     DEV.B,R0        ;
5380  007406  052710  000100      1$:    BIS     #BIT6,(R0)      ;SET EVENT ENABLE
5381  007412  000240                     NOP                     ;WASTE TIME
5382  007414  042710  000100            BIC     #BIT6,(R0)      ;CLEAR IT
5383  007420  052710  000200            BIS     #BIT7,(R0)      ;SET OTHER INTR
5384  007424  000240                     NOP                     ;WASTE TIME
5385  007426  012777  007446  171754     MOV     #2$,@#B.AVEC    ;SET GOOD VECTOR
5386  007434  052710  000100            BIS     #BIT6,(R0)      ;SET IE
5387  007440  000240                     NOP                     ;WASTE TIME
5388  007442  104015                     ERROR   15              ;NO INTERRUPT
5389  007444  000407                     BR      4$              ;CONT TEST
5390  007446  005010            2$:    CLR     (R0)            ;ZERO REGISTER
5391  007450  105203                     INCB    R3              ;UPDATE ICOUNT
5392  007452  100403                     BMI     3$              ;DONE?
5393  007454  012716  007406            MOV     #1$,(SP)        ;SET RETURN
5394  007460  000002                     RTI                     ;EXIT
5395  007462  022626            3$:    CMP     (SP)+,(SP)+     ;POP PC+PSW
5396  007464  005010            4$:    CLR     (R0)            ;DSABLE DEVICE
5397  007466  012677  171720            MOV     (SP)+,@#B.APTY  ;RESTORE PTY
5398  007472  012677  171712            MOV     (SP)+,@#B.AVEC  ;RESTORE VEC
5399  007476  000404                     BR      6$              ;CONT TEST
```

```
                                                ;#################################################################
                                                ;#TEST 41        "DEVICE 'B'" ADDRESS TEST (SEL 2)
                                                ;#TEST TO VERIFY A "BREPLY" RESPONSE
                                                ;#FROM "DEVICE'B'" (SEL 2).
                                                ;#IT IS ASSUMED THAT THE DEVICE
                                                ;#IS AT ADDRESS "165252" (SEL 2).
                                                ;#################################################################
007514  000004                                TST41:  SCOPE
007516  013746  000004                              MOV     4,-(SP)         ;SAVE ADDRESS 4.
007522  013746  000006                              MOV     6,-(SP)         ;SAVE ADDRESS 6.
007526  012701  000017                              MOV     #15.,R1         ;SET INTERNAL ICOUNT TO 15.
007532  013700  001406                              MOV     DEV.B,R0        ;GET DEVICE CSR.
007536  062700  000002                              ADD     #2,R0           ;MAKE IT SEL 2
007542  012737  007566  000004                      MOV     #3$,4           ;SET TIME-OUT VECTOR.
007550  012737  000200  000006                      MOV     #200,6          ;SET PRIO.
007556  005710                              1$:     TST     (R0)            ;REFERENCE DEVICE CSR.
007560  005301                              2$:     DEC     R1              ;ICOUNT = 0?
007562  001375                                      BNE     1$              ;BR IF NO.
007564  000404                                      BR      4$              ;CONT TEST
007566  104012                              3$:     ERROR   12              ;TIME-OUT ERROR.
007570  012716  007560                              MOV     #2$,(SP)        ;SET RETURN ADD.
007574  000002                                      RTI                     ;RETURN.
007576  012637  000006                      4$:     MOV     (SP)+,6         ;RESTORE 6
007602  012637  000004                              MOV     (SP)+,4         ;RESTORE 4

                                                ;#################################################################
                                                ;#TEST 42        R/W TEST OF BIT 6 IN DEVICE 'B' (SEL 2)
                                                ;#TEST THAT BIT 6 IN DEVICE 'B' (SEL 2)
                                                ;#IS R/W.
                                                ;#SET BIT 6 VERIFY IT IS SET.
                                                ;#CLEAR BIT 6 VERIFY IT IS CLEAR.
                                                ;#
                                                ;#################################################################
007606  000004                                TST42:  SCOPE
007610  106427  000200                              MTPS    #200
007614  013700  001406                              MOV     DEV.B,R0        ;LOAD CSR INTO R0
007620  062700  000002                              ADD     #2,R0           ;MAKE IT SEL 2
007624  012705  000100                              MOV     #BIT6,R5        ;SET BIT 6 INTO R5
007630  010510                                      MOV     R5,(R0)         ;WRITE BIT
007632  011004                                      MOV     (R0),R4         ;READ REGISTER
007634  020504                                      CMP     R5,R4           ;OK?
007636  001401                                      BEQ     .+4             ;BR IF OK.
007640  104013                                      ERROR   13              ;REGISTER R/W ERROR
007642  040510                                      BIC     R5,(R0)         ;CLEAR BIT 6
007644  011004                                      MOV     (R0),R4         ;READ DEVICE
007646  042705  000100                              BIC     #BIT6,R5        ;CLEAR EXPECTED
007652  005704                                      TST     R4              ;REGISTER =0?
007654  001401                                      BEQ     .+4             ;BR IF =0!
```

```
(1)  007656  104013                    ERROR   13              ;REGISTER ERROR
(1)  007660  106427  000000            MTPS    #0              ;SET PTY TO 0
(1)
(5)                          ;*****************************************************************
(5)                          ;*TEST 43       R/W TEST OF BIT 7 IN DEVICE 'B' (SEL 2)
(5)                          ;*TEST THAT BIT 7 IN DEVICE 'B' (SEL 2)
(5)                          ;*IS R/W.
(5)                          ;*SET BIT 7 VERIFY IT IS SET.
(5)                          ;*CLEAR BIT 7 VERIFY IT IS CLEAR.
(5)                          ;*
(4)                          ;*****************************************************************
(3)  007664  000004          TST43:  SCOPE
(1)  007666  106427  000200            MTPS    #200
(1)  007672  013700  001406            MOV     DEV.B,R0        ;LOAD CSR INTO R0
(1)  007676  062700  000002            ADD     #2,R0           ;MAKE IT SEL 2
(1)  007702  012705  000200            MOV     #BIT7,R5        ;SET BIT 7 INTO R5
(1)  007706  010510                    MOV     R5,(R0)         ;WRITE BIT
(1)  007710  011004                    MOV     (R0),R4         ;READ REGISTER
(1)  007712  020504                    CMP     R5,R4           ;OK?
(2)  007714  001401                    BEQ     .+4             ;BR IF OK.
(1)  007716  104013                    ERROR   13              ;REGISTER R/W ERROR
(1)  007720  040510                    BIC     R5,(R0)         ;CLEAR BIT 7
(1)  007722  011004                    MOV     (R0),R4         ;READ DEVICE
(1)  007724  042705  000200            BIC     #BIT7,R5        ;CLEAR EXPECTED
(1)  007730  005704                    TST     R4              ;REGISTER =0?
(2)  007732  001401                    BEQ     .+4             ;BR IF =0!
(1)  007734  104013                    ERROR   13              ;REGISTER ERROR
(1)  007736  106427  000000            MTPS    #0              ;SET PTY TO 0
5458
5467
5468                         ;*****************************************************************
(3)                          ;#TEST 44       DEVICE 'B' INTERRUPT TEST (SEL 2)
(4)                          ;#TEST OF DEVICE 'B' INTERRUPTS (SEL 2)
(4)                          ;#SET BIT6 VERIFY NO INTERRUPT
(4)                          ;#CLEAR BIT6; SET BIT7 -VERIFY NO INTERRUPT
(4)                          ;#NOW SET BIT6 AGAIN -EXPECT INTERUPT AT
(4)                          ;#VECTOR "344".
(4)                          ;#
(3)                          ;*****************************************************************
(2)  007742  000004          TST44:  SCOPE
5469 007744  106427  000000            MTPS    #0
5470 007750  017746  171440            MOV     @#B.BVEC,-(SP)  ;SAVE DVB VECTOR
5471 007754  017746  171436            MOV     @#B.BPTY,-(SP)  ;SAVE DVB PRIO
5472 007760  012777  010100  171426    MOV     #5$,@#B.BVEC    ;LOAD VECTOR
5473 007766  012777  000200  171422    MOV     #200,@#B.BPTY   ;LOAD PRIO
5474 007774  005001                    CLR     R1
5475 007776  013700  001406            MOV     DEV.B,R0        ;
5476 010002  062700  000002            ADD     #2,R0           ;MAKE IT SEL 2
5477 010006  052710  000100    1$:     BIS     #BIT6,(R0)      ;SET EVENT ENABLE
5478 010012  000240                    NOP                     ;WASTE TIME
5479 010014  042710  000100            BIC     #BIT6,(R0)      ;CLEAR IT
5480 010020  052710  000200            BIS     #BIT7,(R0)      ;SET OTHER INTR
5481 010024  000240                    NOP                     ;WASTE TIME
5482 010026  012777  010046  171360    MOV     #2$,@#B.BVEC    ;SET GOOD VECTOR
5483 010034  052710  000100            BIS     #BIT6,(R0)      ;SET IE
5484 010040  000240                    NOP                     ;WASTE TIME
```

```
5485  010042  104015                        ERROR   15              ;NO INTERRUPT
5486  010044  000407                        BR      4$              ;CONT TEST
5487  010046  005010              2$:       CLR     (R0)            ;ZERO REGISTER
5488  010050  105203                        INCB    R3              ;UPDATE ICOUNT
5489  010052  100403                        BMI     3$              ;DONE?
5490  010054  012716  010006                MOV     #1$,(SP)        ;SET RETURN
5491  010060  000002                        RTI                     ;EXIT
5492  010062  022626              3$:       CMP     (SP)+,(SP)+     ;POP PC+PSW
5493  010064  005010              4$:       CLR     (R0)            ;DSABLE DEVICE
5494  010066  012677  171324                MOV     (SP)+,@B.BPTY   ;RESTORE PTY
5495  010072  012677  171316                MOV     (SP)+,@B.BVEC   ;RESTORE VEC
5496  010076  000404                        BR      6$              ;CONT TEST
5497  010100  010002              5$:       MOV     R0,R2           ;SAVE R0
5498  010102  011600                        MOV     (SP),R0         ;SAVE PC.
5499  010104  104005                        ERROR   5               ;UNEXPECTED INTERUPT
5500  010106  010200                        MOV     R2,R0           ;RESTORE R0
5501  010110  106427  000000      6$:       MTPS    #0              ;ZERO PTY
5502
5509
5510
  (3)                         ;#################################################################
  (4)                         ;#TEST 45       TEST OF 'EVENT' AT DEVICE 'B'
  (4)                         ;#TEST THAT SETTING BIT1 OF DEVICE 'B' (SEL 2)
  (4)                         ;#CAUSES AN 'EVENT' VECTORING TO ADDRESS 100.
  (4)                         ;#
  (4)                         ;#
  (3)                         ;#################################################################
  (2)  010114  000004         TST45:  SCOPE
5511  010116  106427  000000            MTPS    #0              ;ZERO PSW
5512  010122  017746  171262            MOV     @CLKVEC,-(SP)   ;SAVE VEC
5513  010126  017746  171250            MOV     @CLKPTY,-(SP)   ;SAVE PTY
5514  010132  013700  001406            MOV     DEV.B,R0        ;GET CSR
5515  010136  062700  000002            ADD     #2,R0           ;MAKE IT SEL 2
5516  010142  005003                    CLR     R3              ;ICOUNTER=0
5517  010144  012777  010206  171226    MOV     #2$,@CLKVEC     ;SET VECTOR
5518  010152  012777  000200  171222    MOV     #200,@CLKPTY    ;SET PRIO
5519  010160  005010              1$:   CLR     (R0)            ;ZERO REGISTER
5520  010162  052710  000002            BIS     #BIT1,(R0)      ;SET EVENT
5521  010166  005027                    CLR     (PC)+           ;STALL FOR TIME.
5522  010170  000000              64$:  0
5523  010172  062737  000001  010170    ADD     #1,64$          ;INC TIMER
5524  010200  001374                    BNE     .-6             ;TIMER DONE?
5525  010202  104015                    ERROR   15              ;NO INTERRUPT
5526  010204  000406                    BR      4$              ;CONT
5527  010206  105203              2$:   INCB    R3              ;ICOUNT=ICOUT+1
5528  010210  100403                    BMI     3$              ;BR IF DONE
5529  010212  012716  010160            MOV     #1$,(SP)        ;SET RTN
5530  010216  000002                    RTI                     ;EXIT
5531  010220  022626              3$:   CMP     (SP)+,(SP)+     ;POP PC+PSW
5532  010222  005010              4$:   CLR     (R0)            ;DSABLE DEVICE
5533  010224  012677  171152            MOV     (SP)+,@CLKPTY   ;RESTORE PTY
5534  010230  012677  171144            MOV     (SP)+,@CLKVEC   ;RESTORE VEC
5535  010234  106427  000000            MTPS    #0              ;PTY 0
5536
5544
5545
  (3)                         ;#################################################################
                              ;*TEST 46       DLV11 CHARACTER TEST
```

```
                                        ;#TEST TO OUTPUT CHARACTERS ON THE DLV11
                                        ;#NOT BEING USED AS THE CONSOLE INTERFACE.
                                        ;#PROGRAM WILL DO 2 FULL BINARY COUNT PATTERNS
                                        ;#AND THEN A SLIDE ASCII PATTERN FOR 5 ROWS.
                                        ;#
                                        ;*****************************************************************
010240  000004                  TST46:  SCOPE
010242  012737  000002  001160          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
010250  012700  175610                  MOV     #175610,R0      ;SET DEVICE ADDRESS
010254  012701  000002                  MOV     #2,R1           ;SET BINARY COUNTER
010260  005003                          CLR     R3              ;
010262  005005                          CLR     R5              ;CLEAR COUNTER
010264  005760  000002                  TST     2(R0)           ;CLR RX DONE
010270  005760  000002                  TST     2(R0)           ;
010274  105760  000004          1$:     TSTB    4(R0)           ;PRINTER READY?
010300  100407                          BMI     64$             ;BR IF YES
010302  023737  000000  000000          CMP     0,0             ;WASTE TIME
010310  062703  000001                  ADD     #1,R3           ;DELAY COUNTER+1
010314  001367                          BNE     1$              ;DELAY DONE?  NO!
010316  104016                          ERROR   16              ;TPS BIT7 <>1 (NOT READY>
010320  110560  000006          64$:    MOVB    R5,6(R0)        ;LOAD DATA CHAR.
010324  005003                          CLR     R3              ;CLEAR POINTER
010326  105710                  65$:    TSTB    (R0)            ;RECEIVER READY(DONE)?
010330  100407                          BMI     66$             ;BR IF YES
010332  023737  000000  000000          CMP     0,0             ;WASTE TIME.
010340  062703  000001                  ADD     #1,R3           ;
010344  001370                          BNE     65$             ;DELAY DONE?  NO!
010346  104016                          ERROR   16              ;RECEIVER NOT DONE.
010350  016004  000002          66$:    MOV     2(R0),R4        ;READ DATA
010354  020504                          CMP     R5,R4           ;DATA GOOD?
010356  001401                          BEQ     67$             ;BR IF YES
010360  104017                          ERROR   17              ;DATA COMPARE ERROR.
010362  105205                  67$:    INCB    R5              ;NEXT CHAR
010364  001343                          BNE     1$              ;
010366  005301                          DEC     R1              ;
010370  001341                          BNE     1$              ;
010372  105760  000004                  TSTB    4(R0)           ;
010376  100375                          BPL     .-4             ;
010400  112760  000015  000006          MOVB    #15,6(R0)
010406  105760  000004                  TSTB    4(R0)           ;
010412  100375                          BPL     .-4             ;
010414  112760  000012  000006          MOVB    #12,6(R0)       ;
010422  012705  000005                  MOV     #5.,R5          ;
010426  012703  000040          2$:     MOV     #40,R3          ;
010432  005203                  3$:     INC     R3              ;
010434  022703  000176                  CMP     #176,R3         ;
010440  001772                          BEQ     2$              ;
010442  012701  000110                  MOV     #72.,R1         ;
010446  010304                          MOV     R3,R4           ;
010450  105760  000004                  TSTB    4(R0)           ;
010454  100375                          BPL     .-4             ;
010456  112760  000015  000006          MOVB    #15,6(R0)       ;;
010464  105760  000004                  TSTB    4(R0)           ;
010470  100375                          BPL     .-4             ;
010472  112760  000012  000006          MOVB    #12,6(R0)       ;;
010500  022704  000176          4$:     CMP     #176,R4         ;;
```

```
5594  010504  001002                    BNE     .+6
5595  010506  012704  000040            MOV     #40,R4          ;
5596  010512  105760  000004            TSTB    4(R0)           ;
5597  010516  100375                    BPL     .-4             ;
5598  010520  110460  000006            MOVB    R4,6(R0)        ;
5599  010524  005204                    INC     R4              ;
5600  010526  005301                    DEC     R1              ;
5601  010530  001363                    BNE     4$              ;
5602  010532  005305                    DEC     R5              ;
5603  010534  001336                    BNE     3$              ;
5604  010536  105760  000004            TSTB    4(R0)           ;TP READY?
5605  010542  100375                    BPL     .-4             ;BR IF NO
5606  010544  005060  000006            CLR     6(R0)           ;XMIT A ZERO.
5607  010550  105760  000004            TSTB    4(R0)           ;READY?
5608  010554  100375                    BPL     .-4             ;BR IF NO.
5609  010556  005060  000006            CLR     6(R0)           ;LOAD A ZERO.
5610  010562  105760  000004            TSTB    4(R0)           ;READY?
5611  010566  100375                    BPL     .-4             ;BR IF NO
5612  010570  005005                    CLR     R5              ;
5613  010572  005205                    INC     R5              ;
5614  010574  001376                    BNE     .-2             ;
5615  010576  005760  000002            TST     2(R0)           ;CLEAR RX FLAG.
5616
5623
5624
 (3)                                    ;##################################################################
 (4)                                    ;#TEST 47       DLV11 INTERUPT TEST
 (4)                                    ;#TEST OF DLV11 INTERUPTS.
 (4)                                    ;#THIS TEST WILL EXECUTE 32. INTERUPTS ON
 (4)                                    ;#BOTH THE RECEIVER AND TRANSMITTER AND
 (4)                                    ;#WILL ALSO CHECK THE DATA.
 (3)                                    ;##################################################################
 (2)  010602  000004                    TST47:  SCOPE
 (1)  010604  012737  000002  001160            MOV     #2,$TIMES       ;;DO 2 ITERATIONS
5625  010612  012700  175610                    MOV     #175610,R0      ;GET DLV11 CSR
5626  010616  013746  000300                    MOV     300,-(SP)       ;SAVE
5627  010622  013746  000302                    MOV     302,-(SP)       ;VECTORS
5628  010626  013746  000304                    MOV     304,-(SP)       ;ON THE
5629  010632  013746  000306                    MOV     306,-(SP)       ;STACK.
5630  010636  012737  010732  000304            MOV     #4$,304         ;SET TX VECTOR.
5631  010644  012737  000200  000306            MOV     #200,306        ;SET PRIO.
5632  010652  012737  010760  000300            MOV     #5$,300         ;SET RX VECTOR
5633  010660  012737  000200  000302            MOV     #200,302        ;SET PRIO.
5634  010666  005005                            CLR     R5              ;SET DATA POINTER TO ZERO
5635  010670  052760  000100  000004            BIS     #100,4(R0)      ;SET TX IE.
5636  010676  005003                    1$:     CLR     R3              ;SET TIMER TO ZERO.
5637  010700  062703  000001            2$:     ADD     #1,R3           ;TIME WAITING FOR INTERUPTS.
5638  010704  001375                            BNE     2$              ;KEEP COUNTING.
5639  010706  104016                            ERROR   16              ;DLV11 INTERUPT PROBLEM.
5640  010710  012637  000306            3$:     MOV     (SP)+,306       ;RESTORE
5641  010714  012637  000304                    MOV     (SP)+,304       ;ALL
5642  010720  012637  000302                    MOV     (SP)+,302       ;DLV11
5643  010724  012637  000300                    MOV     (SP)+,300       ;VECTORS.
5644  010730  000444                            BR      TST50           ;;;EXIT TEST
5645
5646  010732                            4$:     ;TRANSMITTER INTERUPTS TO HERE.
5647  010732  110560  000006                    MOVB    R5,6(R0)        ;LOAD DATA CHAR.
```

```
5648  010736  042760  000100  000004          BIC     #100,4(R0)      ;DSABLE TX INTERUPTS.
5649  010744  052760  000100  000000          BIS     #100,0(R0)      ;ENABLE RX INTERUPTS.
5650  010752  012716  010676                  MOV     #1$,(SP)        ;SET RETURN
5651  010756  000002                          RTI                     ;EXIT ISR.
5652
5653  010760                          5$:     ;RECEIVER INTERUPTS TO HERE.
5654  010760  000240                          NOP                     ;
5655  010762  116004  000002                  MOVB    2(R0),R4        ;GET DATA REVEIVERD
5656  010766  020504                          CMP     R5,R4           ;DATA GOOD?
5657  010770  001401                          BEQ     6$              ;BE IF GOOD DATA
5658  010772  104017                          ERROR   17              ;DLV11 DATA ERROR (INTERUPTS)
5659  010774  005205                  6$:     INC     R5              ;UPDATE DATA CHAR.
5660  010776  022705  000040                  CMP     #32.,R5         ;ALL CHARS DONE?
5661  011002  001006                          BNE     7$              ;BR IF NO
5662  011004  042760  000100  000000          BIC     #100,0(R0)      ;DSABLE RX INTERUPTS.
5663  011012  012716  010710                  MOV     #3$,(SP)        ;SET RETURN
5664  011016  000002                          RTI                     ;RETURN
5665  011020  042760  000100  000000  7$:     BIC     #100,0(R0)      ;DSABLE RX INTERUPTS
5666  011026  052760  000100  000004          BIS     #100,4(R0)      ;ENABLE TX INTERUPTS
5667  011034  012716  010676                  MOV     #1$,(SP)        ;SET RETURN
5668  011040  000002                          RTI
5669
5677
5678                                  ;#################################################################
 (3)                                  ;#TEST 50       MINI MEMORY TEST
 (4)                                  ;#MINI MEMORY TESTS
 (4)                                  ;#ALL UNSED MEMORY WILL BE FILLED WITH ALL 1'S
 (4)                                  ;#AND THEN EACH ADDRESS WILL BE TESTED
 (4)                                  ;#WITH A FLOATING ZERO.
 (4)                                  ;#
 (3)                                  ;#################################################################
 (2)  011042  000004                  TST50:  SCOPE
 (1)  011044  012737  000002  001160          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
5679  011052  013701  001374                  MOV     MEMSIZ,R1       ;GET MAX MEMORY SIZE.
5680  011056  162701  000310                  SUB     #310,R1         ;PROTECT ABL
5681  011062  012700  015142                  MOV     #CORMAX,R0      ;GET LAST ADDRESS USED BY PROGRAM
5682  011066  012720  177777          1$:     MOV     #177777,(R0)+   ;
5683  011072  020100                          CMP     R1,R0           ;ALL MEMORY FILLED?
5684  011074  001374                          BNE     1$              ;
5685  011076  012700  015142                  MOV     #CORMAX,R0      ;
5686  011102  042710  000001          2$:     BIC     #BIT0,(R0)      ;
5687  011106  012705  177776                  MOV     #^C<1>,R5       ;
5688  011112  011004                  3$:     MOV     (R0),R4         ;
5689  011114  020504                          CMP     R5,R4           ;
5690  011116  001401                          BEQ     4$              ;
5691  011120  104020                          ERROR   20              ;
5692  011122  000261                  4$:     SEC                     ;
5693  011124  006105                          ROL     R5              ;
5694  011126  000261                          SEC                     ;
5695  011130  006110                          ROL     (R0)            ;
5696  011132  103767                          BCS     3$              ;
5697  011134  005720                          TST     (R0)+           ;
5698  011136  100401                          BMI     .+4             ;
5699  011140  104016                          ERROR   16              ;GENERAL ERROR. BIT15<>1!
5700  011142  020100                          CMP     R1,R0           ;ALL MEMORY DONE?
5701  011144  001356                          BNE     2$              ;
```

5702

```
5758                                    ;*****************************************************************
 (1)                                    
 (1)                                    .SBTTL  END OF PASS ROUTINE
 (1)                                    
 (1)                                    ;#INCREMENT THE PASS NUMBER (SPASS)
 (1)                                    ;#TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
 (1)                                    ;#IF THERES A MONITOR GO TO IT
 (1)                                    ;#IF THERE ISN'T JUMP TO TST1
 (1)                                    
 (1)  011146                            SEOP:
 (1)  011146  000004                            SCOPE
 (1)  011150  005037  001102                    CLR     @TSTNM          ;;ZERO THE TEST NUMBER
 (1)  011154  005037  001160                    CLR     @TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 (1)  011160  005237  001100                    INC     @PASS           ;;INCREMENT THE PASS NUMBER
 (1)  011164  042737  100000  001100            BIC     #100000,@PASS   ;;DON'T ALLOW A NEG. NUMBER
 (1)  011172  005327                            DEC     (PC)+           ;;LOOP?
 (1)  011174  000001            SEOPCT: .WORD   1
 (1)  011176  003022                            BGT     $DOAGN          ;;YES
 (1)  011200  012737                            MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 (1)  011202  000001            SENDCT: .WORD   1
 (1)  011204  011174                            SEOPCT
 (1)  011206  104400  011250                    TYPE    ,SENDMG         ;;TYPE "END PASS #"
 (2)  011212  013746  001100                    MOV     @PASS,-(SP)     ;;SAVE @PASS FOR TYPEOUT
 (2)  011216  104404                            TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 (1)  011220  104400  011265                    TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
 (1)  011224                            $GET42:
 (1)
 (1)  011224  013700  000042                    MOV     @#42,R0         ;;GET MONITOR ADDRESS
 (1)  011230  001405                            BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 (1)  011232  000005                            RESET                   ;;CLEAR THE WORLD
 (1)  011234  004710            $ENDAD: JSR     PC,(R0)                 ;;GO TO MONITOR
 (1)  011236  000240                            NOP                     ;;SAVE ROOM
 (1)  011240  000240                            NOP                     ;;FOR
 (1)  011242  000240                            NOP                     ;;ACT11
 (1)  011244                            $DOAGN:
 (1)  011244  000137  004470                    JMP     @#TST1          ;;RETURN
 (1)  011250  005015  047105  020104    $ENDMG: .ASCIZ  <15><12>/END PASS #/
 (1)  011256  040520  051523  021440
 (1)  011264     000
 (1)  011265     377     377     000    $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
5759                                    ;*****************************************************************
 (1)                                    
 (1)                                    .SBTTL  SCOPE HANDLER ROUTINE
 (1)                                    
 (1)                                    ;#THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS, IT WILL INCREMENT
 (1)                                    ;#AND LOAD THE TEST NUMBER(@TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
 (1)                                    ;#AND LOAD THE ERROR FLAG (@ERFLG) INTO DISPLAY<15:08>
 (1)                                    ;#THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 (1)                                    ;#SW14=1        LOOP ON TEST
 (1)                                    ;#SW11=1        INHIBIT ITERATIONS
 (1)                                    ;#SW09=1        LOOP ON ERROR
 (1)                                    ;#CALL
 (1)                                    ;#      SCOPE           ;;SCOPE=IOT
 (1)                                    
 (1)  011270                            $SCOPE:
 (3)  011270  032777  000400  167640            BIT     #BIT8,@SWR      ;CONFIDENCE?
```

```
(3)  011276  001435                        BEQ     99$             ;BR IF NO
(3)  011300  123737  001102  001440        CMPB    $ISTNM,XXSCOP   ;WAS THIS TEST ALREADY TYPED OUT?
(3)  011306  001421                        BEQ     99$             ;BR IF YES
(3)  011310  123737  001102  015140        CMPB    $ISTNM,LASTN    ;LEGAL TEST #
(3)  011316  003015                        BGT     99$             ;BR IF NO
(3)  011320  105737  001102                TSTB    $ISTNM          ;TEST 0?
(3)  011324  001412                        BEQ     99$             ;BR IF YES (THERE IS NO TEST 0)
(3)  011326  113737  001102  001440        MOVB    $ISTNM,XXSCOP   ;LOAD FOR NEXT TIME
(3)  011334  104400  014644                TYPE    ,XTST           ;TYPE "T "
(3)  011340  013746  001440                MOV     XXSCOP,-(SP)    ;
(3)  011344  104401                        TYPOC                   ;
(3)  011346  104400  014610                TYPE    ,XSPA           ;
(3)  011352                         99$:
(1)  011352  032777  040000  167556  1$:    BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
(1)  011360  001101                        BNE     $OVER           ;;YES IF SW14=1
(1)                                 ;#####START OF CODE FOR THE XOR TESTER#####
(1)  011362  000416                 $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
(1)                                                                 ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
(1)  011364  013746  000004                MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
(1)  011370  012737  011410  000004        MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
(1)  011376  005737  177060                TST     @#177060        ;;TIME OUT ON XOR?
(1)  011402  012637  000004                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1)  011406  000453                        BR      $SVLAD          ;;GO TO THE NEXT TEST
(1)  011410  022626                 5$:    CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
(1)  011412  012637  000004                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1)  011416  000413                        BR      7$              ;;LOOP ON THE PRESENT TEST
(1)  011420                         6$:;#####END OF CODE FOR THE XOR TESTER#####
(1)  011420  105737  001103                TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
(1)  011424  001421                        BEQ     3$              ;;BR IF NO
(1)  011426  123737  001115  001103        CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
(1)  011434  101015                        BHI     3$              ;;BR IF NO
(1)  011436  032777  001000  167472        BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
(1)  011444  001404                        BEQ     4$              ;;BR IF NO
(1)  011446  013737  001110  001106  7$:    MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
(1)  011454  000443                        BR      $OVER
(1)  011456  105037  001103          4$:    CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
(1)  011462  005037  001160                CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
(1)  011466  000415                        BR      1$              ;;ESCAPE TO THE NEXT TEST
(1)  011470  032777  004000  167440  3$:    BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
(1)  011476  001011                        BNE     1$              ;;BR IF YES
(1)  011500  005737  001100                TST     $PASS           ;;IF FIRST PASS OF PROGRAM
(1)  011504  001406                        BEQ     1$              ;;      INHIBIT ITERATIONS
(1)  011506  005237  001104                INC     $ICNT           ;;INCREMENT ITERATION COUNT
(1)  011512  023737  001160  001104        CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
(1)  011520  002021                        BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
(1)  011522  012737  000001  001104  1$:    MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
(1)  011530  013737  011600  001160        MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
(1)  011536  105237  001102          $SVLAD: INCB    $ISTNM          ;;COUNT TEST NUMBERS
(1)  011542  011637  001106                MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
(1)  011546  011637  001110                MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
(1)  011552  005037  001162                CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
(1)  011556  112737  000001  001115        MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1)  011564  013777  001102  167346  $OVER: MOV     $ISTNM,@DISPLAY ;;DISPLAY TEST NUMBER
(1)  011572  013716  001106                MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
(1)  011576  000002                        RTI                     ;;FIXES PS
(1)  011600  000100                 $MXCNT: 100                     ;;MAX. NUMBER OF ITERATIONS
```

```
 5760                                   ;*****************************************************************
  (1)                                   
  (1)                                   .SBTTL  ERROR HANDLER ROUTINE
  (1)                                   
  (1)                                   ;#THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
  (1)                                   ;#SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
  (1)                                   ;#AND GO TO XERR1 ON ERROR
  (1)                                   ;#THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
  (1)                                   ;#SW15=1        HALT ON ERROR
  (1)                                   ;#SW13=1        INHIBIT ERROR TYPEOUTS
  (1)                                   ;#SW10=1        BELL ON ERROR
  (1)                                   ;#SW09=1        LOOP ON ERROR
  (1)                                   ;#CALL
  (1)                                   ;#      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
  (1)                                   
  (1)  011602                           $ERROR:
  (1)  011602  105237  001103           7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
  (1)  011606  001775                           BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
  (1)  011610  013777  001102  167322           MOV     $ISTNM,@#DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
  (1)  011616  032777  002000  167312           BIT     #BIT10,@#SWR    ;;BELL ON ERROR?
  (1)  011624  001402                           BEQ     1$              ;;NO - SKIP
  (1)  011626  104400  001164                   TYPE    ,$BELL          ;;RING BELL
  (1)  011632  005237  001112           1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
  (1)  011636  011637  001116                   MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
  (1)  011642  162737  000002  001116           SUB     #2,$ERRPC
  (1)  011650  117737  167242  001114           MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
  (1)  011656  032777  020000  167252           BIT     #BIT13,@#SWR    ;;SKIP TYPEOUT IF SET
  (1)  011664  001004                           BNE     20$             ;;SKIP TYPEOUTS
  (1)  011666  004737  012100                   JSR     PC,XERR1        ;;GO TO USER ERROR ROUTINE
  (1)  011672  104400  001171                   TYPE    ,$CRLF
  (1)  011676                           20$:
  (1)  011676  005777  167234           2$:     TST     @#SWR           ;;HALT ON ERROR
  (1)  011702  100006                           BPL     3$              ;;SKIP IF CONTINUE
  (1)  011704  000000                           HALT                    ;;HALT ON ERROR!
  (1)  011706  022737  011234  000042           CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
  (1)  011714  001001                           BNE     3$              ;;BRANCH IF NO
  (1)  011716  000000                           HALT                    ;;YES
  (1)  011720  032777  001000  167210   3$:     BIT     #BIT09,@#SWR    ;;LOOP ON ERROR SWITCH SET?
  (1)  011726  001402                           BEQ     4$              ;;BR IF NO
  (1)  011730  013716  001110                   MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
  (1)  011734  005737  001162           4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
  (1)  011740  001402                           BEQ     5$              ;;BR IF NONE
  (1)  011742  013716  001162                   MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
  (1)  011746                           5$:
  (3)  011746  010046                           MOV     R0,-(SP)        ;SAVE R0
  (3)  011750  010146                           MOV     R1,-(SP)        ;SAVE R1
  (3)  011752  032777  010000  167156           BIT     #BIT12,@#SWR    ;ESCAPE TO NEXT TEST ON ERROR?
  (3)  011760  001443                           BEQ     90$             ;BR IF NO
  (4)  011762  000005                           RESET                   ;ISSUE INIT
  (4)  011764  052777  000100  167150           BIS     #100,@#TKS      ;SET INTR ENABLE
  (3)  011772  106427  000000                   MTPS    #0              ;SET PTY TO 0
  (3)  011776  012700  015014                   MOV     #TEST.TABLE+2,R0 ;GET FIRST TEST PC
  (3)  012002  013701  001106                   MOV     $LPADR,R1       ;GET CURRENT TEST PC
  (3)  012006  162701  000002                   SUB     #2,R1           ;MAKE IT EQUALT TO TEST TABLE ENTRY
  (3)  012012  005710                   91$:    TST     (R0)            ;AT END GOTO 90$
  (3)  012014  001425                           BEQ     90$             ;
```

```
(3)  012016  020120                    CMP     R1,(R0)+        ;LOOK FOR CURRENT TEST.
(3)  012020  001424                    BMI     90$             ;BR IF NOT FOUND
(3)  012022  011037  012076            MOV     (R0),92$        ;SAVE TEST PC
(3)  012026  062737  000002  012076    ADD     #2,92$          ;POP PAST SCOPE
(3)  012034  012737  000001  001104    MOV     #1,$ICNT        ;
(3)  012042  013737  011600  001160    MOV     $MXCNT,$TIMES   ;
(3)  012050  012706  001100            MOV     #STACK,SP       ;SET SP
(3)  012054  012746  000000            MOV     #0,-(SP)        ;SET PTYO
(3)  012060  013746  012076            MOV     92$,-(SP)       ;
(3)  012064  000137  011536            JMP     $SVLAD          ;GO INTO SCOPE ROUTINE
(3)  012070  012601            90$:    MOV     (SP)+,R1        ;RESTORE R1
(3)  012072  012600                    MOV     (SP)+,R0        ;RESTORE R0
(3)  012074  000002                    RTI                     ;EXIT
(3)  012076  000000            92$:    0                       ;SAVE TEST PC
(3)

5761 012100  010037  001420    XERR1:  MOV     R0,SAVR0
(1)  012104  010137  001422            MOV     R1,SAVR1
(1)  012110  010237  001424            MOV     R2,SAVR2
(1)  012114  010337  001426            MOV     R3,SAVR3
(1)  012120  010437  001430            MOV     R4,SAVR4
(1)  012124  010537  001432            MOV     R5,SAVR5
(1)  012130  113737  001102  001434    MOVB    $TSTNM,MSKTST
(3)          ;******************************************************************
(2)
(2)          .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
(2)
(2)          ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
(2)          ;*ERROR IS TO BE REPORTED, IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
(2)          ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
(2)
(2)  012136  SERRTYP:
(2)  012136  104400  001171            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
(2)  012142  010046                    MOV     R0,-(SP)        ;;SAVE R0
(2)  012144  005000                    CLR     R0              ;;PICKUP THE ITEM INDEX
(2)  012146  153700  001114            BISB    @#$ITEMB,R0
(2)  012152  001004                    BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
(2)                                                            ;;TYPE THE PC OF THE ERROR
(3)  012154  013746  001116            MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
(3)                                                            ;;ERROR ADDRESS
(2)  012160  104401                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
(2)  012162  000445                    BR      10$             ;;GET OUT
(2)  012164  005300            1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
(2)  012166  006300                    ASL     R0              ;;      WORK FOR THE ERROR TABLE
(2)  012170  006300                    ASL     R0
(2)  012172  006300                    ASL     R0
(2)  012174  062700  001174            ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
(2)  012200  012037  012210            MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
(2)  012204  001404                    BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
(2)  012206  104400                    TYPE                    ;;TYPE THE "ERROR MESSAGE"
(2)  012210  000000            2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
(2)  012212  104400  001171            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
(2)  012216  012037  012226    3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
(2)  012222  001404                    BEQ     5$              ;;SKIP TYPEOUT IF 0
(2)  012224  104400                    TYPE                    ;;TYPE THE "DATA HEADER"
(2)  012226  000000            4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
(2)  012230  104400  001171            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
```

```
(2)  012234  010146                 5$:    MOV     R1,-(SP)        ;;SAVE R1
(2)  012236  012001                        MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
(2)  012240  001415                        BEQ     9$              ;;BR IF NO DATA TO BE TYPED
(2)  012242  012000                        MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
(2)  012244  105720                 6$:    TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
(2)  012246  001003                        BNE     7$              ;;BR IF DECIMAL
(3)  012250  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
(3)  012252  104401                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
(3)  012254  000402                        BR      8$
(3)  012256                         7$:
(3)  012256  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
(3)  012260  104404                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
(2)  012262  005711                 8$:    TST     (R1)            ;;IS THERE ANOTHER NUMBER?
(2)  012264  001403                        BEQ     9$              ;;BR IF NO
(2)  012266  104400  012306                TYPE    ,11$            ;;TYPE TWO(2) SPACES
(2)  012272  000764                        BR      6$              ;;LOOP
(2)
(2)  012274  012601                 9$:    MOV     (SP)+,R1        ;;RESTORE R1
(2)  012276  012600                 10$:   MOV     (SP)+,R0        ;;RESTORE R0
(2)  012300  104400  001171                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
(2)  012304  000207                        RTS     PC              ;;RETURN
(2)  012306  020040     000        11$:    .ASCIZ  /  /            ;;TWO(2) SPACES
(2)          012312                        .EVEN
5762                                ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
(1)
(1)                                 .SBTTL  TYPE ROUTINE
(1)
(1)                                 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1)                                 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1)                                 ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1)                                 ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1)                                 ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1)                                 ;*
(1)                                 ;*CALL:
(1)                                 ;*1) USING A TRAP INSTRUCTION
(1)                                 ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1)                                 ;*OR
(1)                                 ;*      TYPE
(1)                                 ;*      MESADR
(1)                                 ;*
(1)
(1)  012312  105737  001155         $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
(1)  012316  100002                         BPL     1$              ;;BR IF YES
(1)  012320  000000                         HALT                    ;;HALT HERE IF NO TERMINAL
(1)  012322  000407                         BR      3$              ;;LEAVE
(1)  012324  010046                 1$:     MOV     R0,-(SP)        ;;SAVE R0
(1)  012326  017600  000002                 MOV     2(SP),R0        ;;GET ADDRESS OF ASCIZ STRING
(1)  012332  112046                 2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
(1)  012334  001005                         BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
(1)  012336  005726                         TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
(1)  012340  012600                 60$:    MOV     (SP)+,R0        ;;RESTORE R0
(1)  012342  062716  000002         3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
(1)  012346  000002                         RTI                     ;;RETURN
(1)  012350  122716  000011         4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
(1)  012354  001426                         BEQ     8$
(1)  012356  122716  000200                 CMPB    #TCRLF,(SP)     ;;BRANCH IF NOT <CRLF>
```

```
(1)  012362  001004                        BNE     5$
(1)  012364  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
(1)  012366  104400                        TYPE                    ;;TYPE A CR AND LF
(1)  012370  001171                        $CRLF
(1)  012372  000757                        BR      2$              ;;GET NEXT CHARACTER
(1)  012374  004737  012456        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
(1)  012400  123726  001154        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
(1)  012404  001352                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
(1)  012406  013746  001152                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
(1)                                                                 ;;AND THE NULL CHAR.
(1)  012412  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
(1)  012416  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
(1)  012420  004737  012456                JSR     PC,$TYPEC       ;;GO TYPE A NULL
(1)  012424  105337  012522                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
(1)  012430  000770                        BR      7$              ;;LOOP
(1)
(1)                                ;HORIZONTAL TAB PROCESSOR
(1)
(1)  012432  112716  000040        8$:     MOVB    #40,(SP)        ;;REPLACE TAB WITH SPACE
(1)  012436  004737  012456        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
(1)  012442  132737  000007  012522        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
(1)  012450  001372                        BNE     9$              ;;TAB STOP
(1)  012452  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
(1)  012454  000726                        BR      2$              ;;GET NEXT CHARACTER
(1)  012456  105777  166464        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
(1)  012462  100375                        BPL     $TYPEC
(1)  012464  116677  000002  166456        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1)  012472  122766  000015  000002        CMPB    #15,2(SP)       ;;BRANCH IF
(1)  012500  001003                        BNE     1$              ;;NOT <CR>
(1)  012502  105037  012522                CLRB    $CHARCNT        ;;
(1)  012506  000406                        BR      $TYPEX          ;;EXIT
(1)  012510  122766  000012  000002  1$:     CMPB    #12,2(SP)       ;;BRANCH IF
(1)  012516  002002                        BGE     $TYPEX          ;;<LF>
(1)  012520  105227                        INCB    (PC)+           ;;INC SPACE
(1)  012522  000000                $CHARCNT:.WORD  0               ;;COUNT
(1)  012524  000207                $TYPEX: RTS     PC
(1)                                ;       EQUATES
(1)           000011                       HT=11
(1)           000200                       CRLF=200
(1)
5763                               ;*****************************************************************

(1)                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
(1)
(1)                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
(1)                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
(1)                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
(1)                                ;*CALL:
(1)                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
(1)                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
(1)                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
(1)                                ;*      .BYTE   M               ;;M=1 OR 0
(1)                                ;*                                      ;;1=TYPE LEADING ZEROS
(1)                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
(1)                                ;*
(1)                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
```

```
(1)                                     ;*STYPOS OR STYPOC
(1)                                     ;*CALL:
(1)                                     ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
(1)                                     ;*      TYPON                   ;;CALL FOR TYPEOUT
(1)                                     ;*
(1)                                     ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
(1)                                     ;*CALL:
(1)                                     ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
(1)                                     ;*      TYPOC                   ;;CALL FOR TYPEOUT
(1)
(1)  012526  017646  000000             STYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
(1)  012532  116637  000001  012751             MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
(1)  012540  112637  012753                      MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
(1)  012544  062716  000002                      ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
(1)  012550  000406                              BR      STYPON
(1)  012552  112737  000001  012751     STYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
(1)  012560  112737  000006  012753             MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
(1)  012566  112737  000005  012750     STYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
(1)  012574  010346                              MOV     R3,-(SP)        ;;SAVE R3
(1)  012576  010446                              MOV     R4,-(SP)        ;;SAVE R4
(1)  012600  010546                              MOV     R5,-(SP)        ;;SAVE R5
(1)  012602  113704  012753                      MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
(1)  012606  005404                              NEG     R4
(1)  012610  062704  000006                      ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
(1)  012614  110437  012752                      MOVB    R4,$OMODE       ;;SAVE IT FOR USE
(1)  012620  113704  012751                      MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
(1)  012624  016605  000012                      MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
(1)  012630  005003                              CLR     R3              ;;CLEAR THE OUTPUT WORD
(1)  012632  006105                      1$:     ROL     R5              ;;ROTATE MSB INTO "C"
(1)  012634  000404                              BR      3$              ;;GO DO MSB
(1)  012636  006105                      2$:     ROL     R5              ;;FORM THIS DIGIT
(1)  012640  006105                              ROL     R5
(1)  012642  006105                              ROL     R5
(1)  012644  010503                              MOV     R5,R3
(1)  012646  006103                      3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
(1)  012650  105337  012752                      DECB    $OMODE          ;;TYPE THIS DIGIT?
(1)  012654  100016                              BPL     7$              ;;BR IF NO
(1)  012656  042703  177770                      BIC     #177770,R3      ;;GET RID OF JUNK
(1)  012662  001002                              BNE     4$              ;;TEST FOR 0
(1)  012664  005704                              TST     R4              ;;SUPPRESS THIS 0?
(1)  012666  001403                              BEQ     5$              ;;BR IF YES
(1)  012670  005204                      4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
(1)  012672  052703  000060                      BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
(1)  012676  052703  000040              5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
(1)  012702  110337  012746                      MOVB    R3,6$           ;;SAVE FOR TYPING
(1)  012706  104400  012746                      TYPE    ,6$             ;;GO TYPE THIS DIGIT
(1)  012712  105337  012750              7$:     DECB    $OCNT           ;;COUNT BY 1
(1)  012716  003347                              BGT     2$              ;;BR IF MORE TO DO
(1)  012720  002402                              BLT     6$              ;;BR IF DONE
(1)  012722  005204                              INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
(1)  012724  000744                              BR      2$              ;;GO DO THE LAST DIGIT
(1)  012726  012605                      6$:     MOV     (SP)+,R5        ;;RESTORE R5
(1)  012730  012604                              MOV     (SP)+,R4        ;;RESTORE R4
(1)  012732  012603                              MOV     (SP)+,R3        ;;RESTORE R3
(1)  012734  016666  000002  000004              MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
(1)  012742  012616                              MOV     (SP)+,(SP)
```

```
                                        RTI                     ;;RETURN
(1)  012744  000002                     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
(1)  012746  000000                     .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
(1)  012747  000000                     .BYTE   0               ;;OCTAL DIGIT COUNTER
(1)  012750  000000                     .BYTE   0               ;;ZERO FILL SWITCH
(1)  012751  000000                     .WORD   0               ;;NUMBER OF DIGITS TO TYPE
(1)  012752  000000             $OMODE: ;*************************************************************
5764                                    .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
                                        ;*      TYPOS                   ;;GO TO THE ROUTINE

(2)  012754                     $TYPOS:
(3)  012754  010046                     MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  012756  010146                     MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  012760  010246                     MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(3)  012762  010346                     MOV     R3,-(SP)        ;;PUSH R3 ON STACK
(3)  012764  010546                     MOV     R5,-(SP)        ;;PUSH R5 ON STACK
(1)  012766  012746  020200             MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
(1)  012772  016605  000020             MOV     20(SP),R5       ;;GET THE INPUT NUMBER
(1)  012776  100004                     BPL     1$              ;;BR IF INPUT IS POS.
(1)  013000  005405                     NEG     R5              ;;MAKE THE BINARY NUMBER POS.
(1)  013002  112766  000055  000001     MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
(1)  013010  005000              1$:    CLR     R0              ;;ZERO THE CONSTANTS INDEX
(1)  013012  012703  013170             MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
(1)  013016  112723  000040             MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
(1)  013022  005002              2$:    CLR     R2              ;;CLEAR THE BCD NUMBER
(1)  013024  016001  013160             MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
(1)  013030  160105              3$:    SUB     R1,R5           ;;FORM THIS BCD DIGIT
(1)  013032  002402                     BLT     4$              ;;BR IF DONE
(1)  013034  005202                     INC     R2              ;;INCREASE THE BCD DIGIT BY 1
(1)  013036  000774                     BR      3$
(1)  013040  060105              4$:    ADD     R1,R5           ;;ADD BACK THE CONSTANT
(1)  013042  005702                     TST     R2              ;;CHECK IF BCD DIGIT=0
(1)  013044  001002                     BNE     5$              ;;FALL THROUGH IF 0
(1)  013046  105716                     TSTB    (SP)            ;;STILL DOING LEADING 0'S?
(1)  013050  100407                     BMI     7$              ;;BR IF YES
(1)  013052  106316              5$:    ASLB    (SP)            ;;MSD?
(1)  013054  103003                     BCC     6$              ;;BR IF NO
(1)  013056  116663  000001  177777     MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
(1)  013064  052702  000060      6$:    BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
(1)  013070  052702  000040      7$:    BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1)  013074  110223                     MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1)  013076  005720                     TST     (R0)+           ;;JUST INCREMENTING
(1)  013100  020027  000010             CMP     R0,#10          ;;CHECK THE TABLE INDEX
(1)  013104  002746                     BLT     2$              ;;GO DO THE NEXT DIGIT
(1)  013106  003002                     BGT     8$              ;;GO TO EXIT
(1)  013110  010502                     MOV     R5,R2           ;;GET THE LSD
(1)  013112  000764                     BR      6$              ;;GO CHANGE TO ASCII
```

```
(1)  013114  105726                    8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
(1)  013116  100003                            BPL     9$              ;;BR IF NO
(1)  013120  116663  177777  177776            MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
(1)  013126  105013                    9$:     CLRB    (R3)            ;;SET THE TERMINATOR
(3)  013130  012605                            MOV     (SP)+,R5        ;;POP STACK INTO R5
(3)  013132  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  013134  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  013136  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  013140  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  013142  104400  013170                    TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
(1)  013146  016666  000002  000004            MOV     2(SP),4(SP)     ;;ADJUST THE STACK
(1)  013154  012616                            MOV     (SP)+,(SP)
(1)  013156  000002                            RTI                     ;;RETURN TO USER
(1)  013160  023420                    $DTBL:  10000.
(1)  013162  001750                            1000.
(1)  013164  000144                            100.
(1)  013166  000012                            10.
(1)  013170  000004                    $DBLK:  .BLKW   4
5765                                   ;*****************************************************************
(1)                                    
(1)                                    .SBTTL  TTY INPUT ROUTINE
(1)                                    
(1)                                    ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                    ;*CALL:
(1)                                    ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                    ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
(1)                                    ;
(1)                                    
(1)  013200  011646                    $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
(1)  013202  016666  000004  000002            MOV     4(SP),2(SP)     ;;SAVE THE PS
(1)  013210  105777  165726            1$:     TSTB    @$TKS           ;;WAIT FOR
(1)  013214  100375                            BPL     1$              ;;A CHARACTER
(1)  013216  117766  165722  000004            MOVB    @$TKB,4(SP)     ;;READ THE TTY
(1)  013224  042766  177600  000004            BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
(1)  013232  000002                            RTI                     ;;GO BACK TO USER
(2)                                    ;*****************************************************************
(1)                                    ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1)                                    ;*CALL:
(1)                                    ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
(1)                                    ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1)                                    ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
(1)                                    
(1)  013234  010346                    $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
(1)  013236  012703  013342            1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
(1)  013242  022703  013352            2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
(1)  013246  101405                            BLOS    4$              ;;BR IF YES
(1)  013250  104405                            RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
(1)  013252  112613                            MOVB    (SP)+,(R3)      ;;GET CHARACTER
(1)  013254  122713  000177                    CMPB    #177,(R3)       ;;IS IT A RUBOUT
(1)  013260  001003                            BNE     3$              ;;SKIP IF NOT
(1)  013262  104400  001170            4$:     TYPE    ,$QUES          ;;TYPE A '?'
(1)  013266  000763                            BR      1$              ;;CLEAR THE BUFFER AND LOOP
(1)  013270  111337  013340            3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
(1)  013274  104400  013340                    TYPE    ,9$
(1)  013300  122723  000015                    CMPB    #15,(R3)+       ;;CHECK FOR RETURN
(1)  013304  001356                            BNE     2$              ;;LOOP IF NOT RETURN
```

```
(1)  013306  105063  177777                CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
(1)  013312  104400  001172                TYPE    $LF             ;;TYPE A LINE FEED
(1)  013316  012603                        MOV     (SP)+,R3        ;;RESTORE R3
(1)  013320  011646                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
(1)  013322  016666  000004  000002        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
(1)  013330  012766  013342  000004        MOV     #STTYIN,4(SP)
(1)  013336  000002                        RTI                     ;;RETURN
(1)  013340  000                   9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
(1)  013341  000                           .BYTE   0               ;;TERMINATOR
(1)  013342  000010                STTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
5766                                       ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
(1)
(1)                                        .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
(1)
(1)                                        ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
(1)                                        ;*CHANGE IT TO BINARY.
(1)                                        ;*CALL:
(1)                                        ;*      RDOCT                   ;;READ AN OCTAL NUMBER
(1)                                        ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
(1)                                        ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
(1)
(1)  013352  011646                SRDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
(1)  013354  016666  000004  000002        MOV     4(SP),2(SP)     ;;INPUT NUMBER
(3)  013362  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  013364  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  013366  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(1)  013370  104406                1$:     RDLIN                   ;;READ AN ASCIZ LINE
(1)  013372  012600                        MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
(1)  013374  005001                        CLR     R1              ;;CLEAR DATA WORD
(1)  013376  005002                        CLR     R2
(1)  013400  112046                2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
(1)  013402  001412                        BEQ     3$              ;;IF ZERO GET OUT
(1)  013404  006301                        ASL     R1              ;;*2
(1)  013406  006102                        ROL     R2
(1)  013410  006301                        ASL     R1              ;;*4
(1)  013412  006102                        ROL     R2
(1)  013414  006301                        ASL     R1              ;;*8
(1)  013416  006102                        ROL     R2
(1)  013420  042716  177770                BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
(1)  013424  062601                        ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
(1)  013426  000764                        BR      2$              ;;LOOP
(1)  013430  005726                3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
(1)  013432  010166  000012                MOV     R1,12(SP)       ;;SAVE THE RESULT
(1)  013436  010237  013452                MOV     R2,$HIOCT
(3)  013442  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  013444  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  013446  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  013450  000002                        RTI                     ;;RETURN
(1)  013452  000000                $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
5768                                    ;*****************************************************************
  (1)                                   
  (1)                                   .SBTTL  TRAP DECODER
  (1)                                   
  (1)                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
  (1)                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
  (1)                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
  (1)                                   ;*GO TO THAT ROUTINE.
  (1)                                   
  (1)  013454  010046                   $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
  (1)  013456  016600  000002                   MOV     2(SP),R0        ;;GET TRAP ADDRESS
  (1)  013462  005740                           TST     -(R0)           ;;BACKUP BY 2
  (1)  013464  111000                           MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
  (1)  013466  006300                           ASL     R0              ;;POSITION FOR INDEXING
  (1)  013470  016000  013476                   MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
  (1)  013474  000200                           RTS     R0              ;;GO TO ROUTINE
  (1)                                   
  (3)                                   
  (3)                                   .SBTTL  TRAP TABLE
  (3)                                   
  (3)                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
  (3)                                   ;*BY THE "TRAP" INSTRUCTION.
  (3)                                   
  (3)                                   ;       ROUTINE
  (3)                                   ;       -------
  (3)  013476                           $TRPAD:
  (3)  013476  012312                           $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
  (3)  013500  012552                           $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
  (3)  013502  012526                           $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
  (3)  013504  012566                           $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
  (3)  013506  012754                           $TYPDS  ;;CALL=TYPDS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
  (3)  013510  013200                           $RDCHR  ;;CALL=RDCHR    TRAP+5(104405)  TTY TYPEIN CHARACTER ROUTINE
  (3)  013512  013234                           $RDLIN  ;;CALL=RDLIN    TRAP+6(104406)  TTY TYPEIN STRING ROUTINE
  (3)  013514  013352                           $RDOCT  ;;CALL=RDOCT    TRAP+7(104407)  READ AN OCTAL NUMBER FROM TTY
```

```
5770  013516                          ROMMAP:
5778  013516  000400  000440  000167          .WORD   000400,000440,000167,000432,132710,100200,001775
5779  013534  102507  016005  000002          .WORD   102507,016005,000002,032705,177560,001406,005703
5780  013552  001100  120527  000101          .WORD   001100,120527,000101,001075,000207,042705,177760
5781  013570  126560  173064  000002          .WORD   126560,173064,000002,001066,000207
5782  013602     000     201     202          .BYTE   000,201,202,003,204,005,006,207
5783  013612     210     011     012          .BYTE   210,011,012,213,014,215,216,017
5784  013622  012700  175610  112710          .WORD   012700,175610,112710,000006,12706,077776,005003
5785  013640  004767  000142  020527          .WORD   004767,000142,020527,040001,001372,004767,00130
5786  013656  005705  001366  004767          .WORD   005705,001366,004767,000206,010502,162702,000004
5787  013674  004767  000174  010501          .WORD   004767,000174,010501,005702,001431,160705,022705
5788  013712  000240  003407  060205          .WORD   000240,003407,060205,022705,177704,003003,004767
5789  013730  000164  000733  004767          .WORD   000164,000733,004767,000046,005702,100402,110521
5790  013746  000772  105703  001403          .WORD   000772,105703,001403,004767,000144,000720,004767
5791  013764  000144  000723  004767          .WORD   000144,000723,004767,000012,105703,001366,004767
5792  014002  000126  010107  004767          .WORD   000126,010107,004767,177514,010546,111666,000001
5793  014020  042716  007760  004767          .WORD   042716,007760,004767,177476,006305,006305,006305
5794  014036  006305  042705  170017          .WORD   006305,042705,170017,052605,060503,005302,000207
5795  014054  032710  100200  001622          .WORD   032710,100200,001622,100734,016005,000002,100731
5796  014072  000207  004767  177706          .WORD   000207,004767,177706,010504,004767,177700,000305
5797  014110  105005  150405  000207          .WORD   105005,150405,000207,112705,000117,000405,112705
5798  014126  000102  000402  112705          .WORD   000102,000402,112705,000015,105760,000004,100375
5799  014144  110560  000006  000207          .WORD   110560,000006,000207
5800
5801  014152     240     037     015          .BYTE   240,037,015,005,024,224
5802  014160  005004  012700  177500          .WORD   005004,012700,177500,000005,010410,012701,173434
5803  014176  012702  000375  112103          .WORD   012702,000375,112103,112110,100407,130310,001776
5804  014214  105202  100772  116012          .WORD   105202,100772,116012,000002,000771,005710,100756
5805  014232  005002  120312  001377          .WORD   005002,120312,001377,00112
5806
```

```
5812  014516  017746  164422          KBISR:  MOV     @#TKB,-(SP)     ;SAVE CHAR
5813  014522  042716  000200                  BIC     #BIT7,(SP)      ;CLEAR BIT SEVEN
5814  014526  022716  000024                  CMP     #24,(SP)        ;WAS IT <^T> (TEST NO.)
5815  014532  001004                          BNE     3$              ;BR IF NO
5816  014534  012706  001100                  MOV     #STACK,SP       ;SET STACK
5817  014540  000137  004412                  JMP     X1$             ;GO TO ROUTINE
5818  014544  022726  000007          3$:     CMP     #7,(SP)+        ;WAS IT <^G>?
5819  014550  001016                          BNE     1$              ;BR IF NO
5820  014552  104400  014614                  TYPE    ,XSWR           ;SWR=
5821  014556  017746  164354                  MOV     @#SWR,-(SP)     ;
5822  014562  104401                          TYPOC                   ;
5823  014564  104400  014625                  TYPE    ,EQUALS         ;
5824  014570  104407                          ROOCT                   ;
5825  014572  105737  013342                  TSTB    @#TTYIN         ;WAS JUST <CR> HIT?
5826  014576  001402                          BEQ     2$              ;BR IF YES
5827  014600  011677  164332                  MOV     (SP),@#SWR      ;SET UP SWR
5828  014604  005726                  2$:     TST     (SP)+           ;SET STACK
5829  014606  000002                  1$:     RTI                     ;EXIT
5830  014610  020040  000040          XSPA:   .ASCIZ  / /
5831  014614  024200  053523  024522  XSWR:   .ASCIZ  <200>"(SWR)=/"
5832  014622     057  027475     000  EQUALS: .ASCIZ  "/=/"
5833  014631     200  042524  052123  XTSTN:  .ASCIZ  <200>/TEST NO: /
5834  014644  052200  000040          XTST:   .ASCIZ  <200>/T /
5835                                  .EVEN
5836
5837  014650                          BREPLY.TABLE:
5838  014650  165250  000004                  165250,004
5839  014654  172524  000004                  172524,004
5840  014660  173000  001000                  173000,512.
5841  014664  177546  000002                  177546,002
5842  014670  177550  000010                  177550,010
5843  014674  177560  000010                  177560,010
5844  014700  175610  000010                  175610,010
5845  014704  000000  000000                  000000,000
5846  014710  000000  000000                  000000,000
5847  014714  000000  000000                  000000,000
5848  014720  000000  000000                  000000,000
5849  014724  000000  000000                  000000,000
5850  014730  000000  000000                  000000,000
5851
5852                                  ;.PRINT . ;DUBUG PC
5853  014734  011637  015006          DEBUG:  MOV     (SP),1$
5854  014740  162737  000002  015006          SUB     #2,1$
5855  014746  017737  000034  015010          MOV     @1$,2$
5856  014754  022737  106400  015010          CMP     #MTPS,2$
5857  014762  001401                          BEQ     .+4
5858  014764  000000                          HALT                    ;REAL TRAP TO ADDRESS '4'.
5859  014766  011637  015006                  MOV     (SP),1$
5860  014772  062716  000002                  ADD     #2,(SP)
5861  014776  017766  000004  000002          MOV     @1$,2(SP)
5862  015004  000002                          RTI
5863  015006  000000          1$:     0
5864  015010  000000          2$:     0
5865  015012                  TEST.TABLE:
5866  015012  000000                  0
5867  015014  004470                  TST1                    ;PC FOR TEST NUMBER 1
```

```
                (2)  015016  004564          TST2
                (2)  015020  004662          TST3
                (2)  015022  004720          TST4
                (2)  015024  005006          TST5
                (2)  015026  005222          TST6
                (2)  015030  005402          TST7
                (2)  015032  005544          TST10
                (2)  015034  005632          TST11
                (2)  015036  005674          TST12
                (2)  015040  005736          TST13
                (2)  015042  006000          TST14
                (2)  015044  006040          TST15
                (2)  015046  006104          TST16
                (2)  015050  006146          TST17
                (2)  015052  006210          TST20
                (2)  015054  006314          TST21
                (2)  015056  006356          TST22
                (2)  015060  006420          TST23
                (2)  015062  006462          TST24
                (2)  015064  006524          TST25
                (2)  015066  006566          TST26
                (2)  015070  006630          TST27
                (2)  015072  006672          TST30
                (2)  015074  006726          TST31
                (2)  015076  006770          TST32
                (2)  015100  007104          TST33
                (2)  015102  007144          TST34
                (2)  015104  007232          TST35
                (2)  015106  007300          TST36
                (2)  015110  007346          TST37
                (2)  015112  007514          TST40
                (2)  015114  007606          TST41
                (2)  015116  007664          TST42
                (2)  015120  007742          TST43
                (2)  015122  010114          TST44
                (2)  015124  010240          TST45
                (2)  015126  010502          TST46
                (2)  015130  011042          TST47
                (2)  015132  011146          TST50
5884            015134  011146          $EOP
5885            015136  000000          0000
5888            015140  000050  LASTN:  50
5889            015142  000000  CORMAX:  0
5890                    000001  .END

                                ;PC FOR TEST NUMBER 2
                                ;PC FOR TEST NUMBER 3
                                ;PC FOR TEST NUMBER 4
                                ;PC FOR TEST NUMBER 5
                                ;PC FOR TEST NUMBER 6
                                ;PC FOR TEST NUMBER 7
                                ;PC FOR TEST NUMBER 10
                                ;PC FOR TEST NUMBER 11
                                ;PC FOR TEST NUMBER 12
                                ;PC FOR TEST NUMBER 13
                                ;PC FOR TEST NUMBER 14
                                ;PC FOR TEST NUMBER 15
                                ;PC FOR TEST NUMBER 16
                                ;PC FOR TEST NUMBER 17
                                ;PC FOR TEST NUMBER 20
                                ;PC FOR TEST NUMBER 21
                                ;PC FOR TEST NUMBER 22
                                ;PC FOR TEST NUMBER 23
                                ;PC FOR TEST NUMBER 24
                                ;PC FOR TEST NUMBER 25
                                ;PC FOR TEST NUMBER 26
                                ;PC FOR TEST NUMBER 27
                                ;PC FOR TEST NUMBER 30
                                ;PC FOR TEST NUMBER 31
                                ;PC FOR TEST NUMBER 32
                                ;PC FOR TEST NUMBER 33
                                ;PC FOR TEST NUMBER 34
                                ;PC FOR TEST NUMBER 35
                                ;PC FOR TEST NUMBER 36
                                ;PC FOR TEST NUMBER 37
                                ;PC FOR TEST NUMBER 40
                                ;PC FOR TEST NUMBER 41
                                ;PC FOR TEST NUMBER 42
                                ;PC FOR TEST NUMBER 43
                                ;PC FOR TEST NUMBER 44
                                ;PC FOR TEST NUMBER 45
                                ;PC FOR TEST NUMBER 46
                                ;PC FOR TEST NUMBER 47
                                ;PC FOR TEST NUMBER 50
                        ;END PASS PC
                        ;TABLE TERMINATOR -
```

```
BIT0    = 000001          4716#   5217    5686
BIT00   = 000001          4716#
BIT01   = 000002          4716#
BIT02   = 000004          4716#
BIT03   = 000010          4716#
BIT04   = 000020          4716#
BIT05   = 000040          4716#
BIT06   = 000100          4716#
BIT07   = 000200          4716#
BIT08   = 000400          4716#
BIT09   = 001000          4716#   5759    5760
BIT1    = 000002          4716#   5217    5520
BIT10   = 002000          4716#   5217    5760
BIT11   = 004000          4716#   5217    5759
BIT12   = 010000          4716#   5217    5760
BIT13   = 020000          4716#   5217    5760
BIT14   = 040000          4716#   5217    5759
BIT15   = 100000          4716#   5217
BIT2    = 000004          4716#   5217
BIT3    = 000010          4716#   5217
BIT4    = 000020          4716#   5217
BIT5    = 000040          4716#   5217
BIT6    = 000100          4716#   5046    5101    5123    5217    5361    5380    5382    5386    5457    5477    5479    5483
BIT7    = 000200          4716#   5217    5361    5383    5457    5480    5813
BIT8    = 000400          4716#   5217    5759
BIT9    = 001000          4716#   5217
BPTVEC= 000014            4716#
BREPLY  014650            5155    5843#
B.APTY  001412            4827#   4901    4902#   5375    5377#   5397#
B.AVEC  001410            4826#   4901#   5374    5376#   5385#   5398#
B.BPTY  001416            4829#   4903    4904#   5471    5473#   5494#
B.BVEC  001414            4828#   4903#   5470    5472#   5482#   5495#
CLKCSR  001376            4821#   5011    5040    5100    5101#   5123#
CLKPTY  001402            4823#   4899    4900#   5042    5044#   5072#   5079#   5093    5095#   5125#   5513    5518#   5533#
CLKVEC  001400            4822#   4899#   5041    5043#   5071#   5080#   5092    5094#   5126#   5512    5517#   5534#
CORMAX  015142            5681    5685    5889#
DDISP = 177570            4716#   4720
DEBUG   014734            5859#
DEV.A   001404            4824#   5182    5217    5226    5243    5265    5304
DEV.B   001406            4825#   5324    5361    5379    5417    5457    5475    5514
DF1     003716            4725    4731    4737    4743    4748    4754    4760    4766    4772    4778    4784    4790    4796
                          4802    4808    4814    4887#
DH1     002520            4723    4864#
DH10    003210            4764    4871#
DH11    003276            4782    4872#
DH12    003364            4788    4873#
DH13    003433            4800    4874#
DH14    003474            4806    4875#
DH15    003562            4812    4876#
DH2     002574            4729    4865#
DH3     002662            4735    4752    4758    4866#
DH4     002731            4741    4867#
DH5     003017            4746    4868#
DH6     003074            4770    4869#
DH7     003142            4776    4794    4870#
DISPLA  001140            4720#   4895#   5759#   5760#
```

```
DISPRE  000174          4717#  4895
DSWR  = 177570          4716#  4720
DT1     003650          4724   4736   4747   4753   4759   4771   4777   4789   4795   4879#
DT10    003716          4886#
DT2     003664          4730   4742   4765   4783   4807   4813   4880#
DT3     003704          4801   4881#
DT4     003716          4882#
DT5     003716          4883#
DT6     003716          4884#
DT7     003716          4885#
EMTVEC= 000030          4716#  4895#
EM1     001442          4725   4843#
EM10    001741          4763   4850#
EM11    001771          4769   4851#
EM12    002026          4775   4852#
EM13    002063          4781   4853#
EM14    002113          4787   4854#
EM15    002164          4793   4855#
EM16    002220          4799   4856#
EM17    002237          4805   4857#
EM2     001500          4728   4844#
EM20    002261          4811   4858#
EM21    002277          4859#  5151
EM3     001533          4734   4845#
EM4     001572          4740   4846#
EM5     001624          4745   4847#
EM6     001651          4751   4848#
EM7     001701          4757   4849#
EQUALS  014625          5823   5835#
ERRVEC= 000004          4716#  5759#
GWS   = ******  U       4717   4896   5768
IOTVEC= 000020          4716#  4895#  4905#
KBISR   014516          4907   5812#
LASTN   015140          4918   5759   5888#
MEMSIZ  001374          4817#  5148   5679
MSKTST  001434          4836#  4879   4880   4881   5761#
PC    =%000007          4716#  5052#  5521#  5758#  5760#  5761#  5762#
PIRQ  = 177772          4716#
PIRQVE= 000240          4716#
PR0   = 000000          4716#
PR1   = 000040          4716#
PR2   = 000100          4716#
PR3   = 000140          4716#
PR4   = 000200          4716#
PR5   = 000240          4716#
PR6   = 000300          4716#
PR7   = 000340          4716#
PS    = 177776          4716#
PSW   = 177776          4716#
PWRVEC= 000024          4716#
RDCHR = 104405          5765   5768#
RDLIN = 104406          5766   5768#
RDOCT = 104407          4915   5768#  5824
RESVEC= 000010          4716#
ROMMAP  013516          4990   5770#  5776   5808
R0    =%000000          4716#  4916#  4918   4920   4922#  4923   4936#  4942   4943   4944#  4962#  4968#  4988#
```

```
                    4991    4996    5011*   5014    5040*   5047*   5048    5056*   5057    5065    5067*   5100*   5143*
                    5144    5145    5148*   5154    5156    5158*   5166*   5167    5182*   5185    5217*   5226*   5228*
                    5229    5243*   5246*   5247    5265*   5266*   5269    5271*   5272    5278    5304*   5305*   5308
                    5324*   5327    5361*   5379*   5380*   5382*   5383*   5386*   5390*   5396*   5400    5401*   5403*
                    5417*   5418*   5421    5457*   5475*   5476*   5477*   5478*   5480*   5483*   5487*   5493*   5497
                    5498*   5500*   5514*   5515*   5519*   5520*   5532*   5546*   5550    5551    5552    5558*   5560
                    5566    5574    5576*   5577    5579*   5587    5589*   5590    5592*   5596    5598*   5604    5606*
                    5607    5609*   5610    5615    5625*   5635*   5647*   5648*   5649*   5665    5663*   5665*   5666*
                    5681*   5682*   5683    5685*   5686*   5688    5695*   5697    5700    5758*   5760*   5761*   5762*
                    5764*   5766*   5768*

R1      =x000001    4716#   4937*   4945*   4963*   4972*   4989*   4997*   5010*   5015*   5045*   5062*   5098*   5102*
                    5109    5118    5119*   5155*   5156    5158    5160    5161    5181*   5186*   5269*   5270*   5287*
                    5288    5323*   5328*   5378*   5416*   5422*   5474*   5547*   5572*   5585*   5600*   5679*   5680*
                    5683    5700    5760*   5761*   5764*   5766*

R2      =x000002    4716#   4990*   4992    4996    5109*   5110*   5111*   5113*   5114    5267*   5273*   5274    5279
                    5400*   5403    5497*   5500    5761    5764*   5766*

R3      =x000003    4716#   5111    5118*   5244*   5262*   5264*   5280    5289*   5290    5291*   5488*   5516*   5527*
                    5548*   5555*   5559*   5563*   5581*   5582*   5583    5586    5636*   5637*   5761    5763*   5764*
                    5765*

R4      =x000004    4716#   4991*   4993    5048*   5049    5057*   5059    5096*   5120*   5217*   5229*   5230    5247*
                    5248    5272*   5275    5278*   5282    5288*   5291    5309*   5361*   5457*   5566*   5567    5586*
                    5593    5595*   5598    5599*   5655*   5656    5688*   5689    5761    5763*
                    5763*

R5      =x000005    4716#   4992*   4993    5046*   5047    5049    5056    5058*   5070*   5074*   5097*   5106*   5217*
                    5227*   5228    5230    5233*   5245*   5246    5248    5251*   5274*   5275    5281*   5282    5285
                    5290*   5291    5306*   5361*   5457*   5549*   5558    5567    5570*   5580*   5602*   5612*   5613*
                    5634*   5647    5656    5659*   5660    5687*   5689    5693*   5761    5763*   5764*

R6      =x000006    4716#   4995*
R7      =x000007    4716#
SAVR0     001420    4830#   4879    4880    5761*
SAVR1     001422    4831#   5761*
SAVR2     001424    4832#   5761*
SAVR3     001426    4833#   5761*
SAVR4     001430    4834#   4880    5761*
SAVR5     001432    4835#   4880    5761*
SETTST    003726    4719    4879#
SP      =x000006    4716#   4895*   4916    4938*   4939*   4949*   4951    4952    4964*   4965*   4971    4974*   4977
                    4978    4979    5008*   5009*   5019*   5021    5022    5041*   5042*   5068*   5078    5079    5080
                    5092*   5093*   5124    5125    5126    5139*   5140*   5147    5165    5169    5170    5179*   5180*
                    5190*   5192    5193    5321*   5322*   5332*   5334    5335    5374*   5375*   5393*   5395    5397
                    5398    5401    5414*   5415*   5426*   5428    5429    5470*   5471*   5490*   5492    5494    5495
                    5498    5512*   5513*   5529*   5531    5533    5534    5626*   5627*   5628*   5629*   5640    5641
                    5642    5643    5650*   5663*   5667*   5758*   5759*   5760*   5761*   5762*   5763*   5764*   5765*
                    5766*   5768*   5812*   5813*   5814    5816*   5818    5821*   5827    5828    5859    5865    5866*
                    5867#

STACK =   001100    4716#   4895    5760    5816
START     003732    4717    4895#
STKLMT=   177774    4716#
SWR       001136    4720#   4895*   5759    5760    5821    5827*
SWREG     000176    4717#   4895
SW0     =   000001  4716#
SW00    =   000001  4716#
SW01    =   000002  4716#
SW02    =   000004  4716#
SW03    =   000010  4716#
SW04    =   000020  4716#
SW05    =   000040  4716#
```

```
SW06  = 000100          4716#
SW07  = 000200          4716#
SW08  = 000400          4716#
SW09  = 001000          4716#
SW1   = 000002          4716#
SW10  = 002000          4716#
SW11  = 004000          4716#
SW12  = 010000          4716#
SW13  = 020000          4716#
SW14  = 040000          4716#
SW15  = 100000          4716#
SW2   = 000004          4716#
SW3   = 000010          4716#
SW4   = 000020          4716#
SW5   = 000040          4716#
SW6   = 000100          4716#
SW7   = 000200          4716#
SW8   = 000400          4716#
SW9   = 001000          4716#
TBITVE= 000014          4716#
TCRLF = 000200          5762#
TEST.T  015012          4923    5760    5871#
THT   = 000011          5762#
TKVEC = 000060          4716#
TPVEC = 000064          4716#
TRAPVE= 000034          4716#   4895#
TRTVEC= 000014          4716#
TSTFLG  001436          4837#   4894#   4911    4913#
TST1    004470          4912    4935#   5758    5883
TST10   005544          5178#   5883
TST11   005632          5217#   5883
TST12   005674          5217#   5883
TST13   005736          5217#   5883
TST14   006000          5217#   5883
TST15   006042          5217#   5883
TST16   006104          5217#   5883
TST17   006146          5217#   5883
TST2    004564          4961#   5883
TST20   006210          5217#   5883
TST21   006252          5217#   5883
TST22   006314          5217#   5883
TST23   006356          5217#   5883
TST24   006420          5217#   5883
TST25   006462          5217#   5883
TST26   006524          5217#   5883
TST27   006566          5217#   5883
TST3    004663          4987#   5883
TST30   006630          5217#   5883
TST31   006672          5225#   5883
TST32   006726          5242#   5883
TST33   006770          5264#   5883
TST34   007104          5303#   5883
TST35   007144          5320#   5883
TST36   007232          5361#   5883
TST37   007300          5361#   5883
TST4    004720          5007#   5883
```

```
TST40   007346          5372#   5883
TST41   007514          5413#   5883
TST42   007606          5457#   5883
TST43   007664          5457#   5883
TST44   007742          5468#   5883
TST45   010114          5510#   5883
TST46   010240          5545#   5883
TST47   010602          5624#   5883
TST5    005006          5038#   5883
TST50   011042          5644    5678#   5883
TST6    005222          5091#   5883
TST7    005402          5138#   5883
TYPDS = 104404          5758    5761    5768#
TYPE  = 104400          4896    4914    5151    5758    5759    5760    5761    5762    5763    5764    5765    5768#   5820
                        5823
TYPOC = 104401          5759    5761    5768#   5822
TYPON = 104403          5768#
TYPOS = 104402          5768#
XERR1   012100          5700    5761#
XSPA    014610          5759    5833#
XSWR    014614          5820    5834#
XTST    014644          5759    5837#
XTSTN   014631          4914    5836#
XXSCOP  001440          4838#   4906*   5759*
X1$     004412          4914#   4917    4919    5817
$BDADR  001122          4720#
$BDDAT  001126          4720#   4895
$BELL   001164          4720#   5760
$CHARC  012522          5762#*
$CNTAG  001100          4720#   4895
$CR3  = 000000          4720#
$CRN  = 000001          4720#
$CRLF   001171          4720#   5760    5761    5762    5765
$DBLK   013170          5764#
$DDRGN  011244          5758#
$DTBL   013160          5764#
$ENDAD  011234          4896    5758#   5760
$ENDCT  011202          4895    5758#
$ENDMG  011250          5758#
$ENULL  011265          5758#
$EOP    011146          5758#   5884
$EOPCT  011174          4895*   5758#
$ERFLG  001103          4720#   5759*   5760*
$ERMAX  001115          4720#   4895*   5759*
$ERROR  011602          4895    5760#
$ERRPC  001116          4720#   4879    4880    4881    5760*   5761
$ERRTB  001174          4720#   5761
$ERRTY  012136          5761#
$ERTTL  001112          4720#   4879    4880    4881    5760*
$ESCAP  001162          4720#   4895*   5759*   5760
$FILLC  001154          4720#   5762
$FILLS  001153          4720#   5762
$GDADR  001120          4720#
$GDDAT  001124          4720#
$GET42  011224          5758#
$HD   = 000000          4713
```

```
SHIOCT  013452          5766#
SICNT   001104          4720#   5759#   5760#
SITEMB  001114          4720#   5760#   5761
SLF     001172          4720#   5760    5762    5765
SLPADR  001106          4720#   4895*   4923*   4925*   4926    4927    5759*   5760
SLPENR  001110          4720#   4895*   4926*   5759*   5760
SMAIL = ###### U        4895    4935    4961    4987    5007    5038    5091    5138    5178    5217    5225    5242    5264
                        5303    5320    5361    5372    5413    5457    5468    5510    5545    5624    5678    5759    5760
                        5762
SMXCNT  011600          5759#   5760
SNULL   001152          4720#   5762
SNWTST= 000001          4935#   4961#   4987#   5007#   5038#   5091#   5138#   5178#   5217#   5225#   5242#   5264#   5303#
                        5320#   5361#   5372#   5413#   5457#   5468#   5510#   5545#   5624#   5678#
SOCNT   012750          5763#*
SOMODE  012752          5763#*
SOVER   011564          5759#
SPASS   001100          4720#   4879    4880    4881    5758*   5759
SQ      = 000051        5874#   5883#
SQUES   001170          4720#   5760    5762    5765
SRDCHR  013200          5765#   5768
SRDDEC= ###### U        5768
SRDLIN  013234          5765#   5768
SRDOCT  013362          5766#   5768
SRDSZ   = 000010        5765#
SR2A    = ###### U      5768
SSAVRE= ###### U        5768
SSCOPE  011270          4895    5759#
SSETUP= 000127          4893#   4895    4896    5758    5760
SSTUP   = 177777        4893#
SSVLAD  011536          5759#   5760
SSWR    = 167000        4698#   4713    4715    4720    4895    4935    4961    4987    5007    5038    5091    5138    5178
                        5217    5225    5242    5264    5303    5320    5361    5372    5413    5457    5468    5510    5545
                        5624    5678    5758    5759    5760
SSWRMK= 000000          5759
STIMES  001160          4720#   4895*   5038*   5138*   5225*   5242*   5303*   5545*   5624*   5678*   5758*   5759*   5760*
STKB    001144          4720#   5765    5812
STKS    001142          4720#   4909*   5307*   5760*   5765
STMPO   001156          4720#   5065*   5067    5279*   5280*   5281
STN     = 000051        4693#   4713    4912    4935#   4961#   4987#   5007#   5038#   5091#   5138#   5178#   5217#   5225#
                        5242#   5264#   5303#   5320#   5361#   5372#   5413#   5457#   5468#   5510#   5545#   5624#   5644
                        5678#   5876    5886
STPB    001150          4720#   5762*
STPFLG  001155          4720#   5762
STPS    001146          4720#   5762
STRAP   013454          4895    5768#
STRP    = 000010        5768#
STRPAD  013476          5768#
STSTNM  001102          4720#   4920*   5758*   5759*   5760    5761
STTYIN  013342          5765#   5825
STYPBM= ###### U        5768
STYPDS  012754          5764#   5768
STYPE   012312          5762#   5768
STYPEC  012456          5762#
STYPEX  012524          5762#
STYPOC  012552          5763#   5768
STYPON  012566          5763#   5768
```

```
$TYPOS  012526          5763#   5768
$XTSTR  011362          5759#
$DFILL  012751          5763#*
$40CAT= ****** U        5759    5760
.       = 015144        4717#   4718#   4720#   4895    4896#   4994    5050    5055    5060    5075    5112    5115    5149
                        5217    5231    5249    5276    5283    5292    5309    5361    5457    5524    5575    5578    5588
                        5591    5594    5597    5605    5608    5611    5614    5698    5758    5759    5760    5761#   5762
                        5764#   5765#   5776#   5808#   5863
```

```
COMMEN  1023#  4716#
ENDCOM  1035#  4716#
ERROR   4716#  4948   4970   4995   5018   5051   5061   5066   5076   5104   5116   5150   5163   5189   5217
        5232   5250   5277   5284   5293   5310   5331   5361   5388   5402   5425   5457   5485   5499   5525
        5557   5565   5569   5639   5658   5691   5699
ESCAPE  1146#  4716#
GETPRI   797#
MSG     4929#  4935   4955#  4961   4981#  4987   5000#  5007   5024#  5038   5083#  5091   5129#  5138   5171#
        5178   5217#  5219#  5225   5236#  5242   5255#  5264   5297#  5303   5313#  5320   5361#  5363#  5372
        5406#  5413   5457#  5459#  5468   5503#  5510   5537#  5545   5617#  5624   5670#  5678
MULT    3529#  4716#
NEWTST  1079#  4716#  4935   4961   4987   5007   5038   5091   5138   5178   5217   5225   5242   5264   5303
        5320   5361   5372   5413   5457   5468   5510   5545   5624   5678
POP     1532#  4716#  5764   5766
PUSH    1524#  4716#  5764   5766
REPORT  4507#
SCOPE   4716#  4935   4961   4987   5007   5038   5091   5138   5178   5217   5225   5242   5264   5303   5320
        5361   5372   5413   5457   5468   5510   5545   5624   5678   5758
SETPRI   783#
SETTRA  5768#
SETUP    825#  4716#  4895
SKIP    1180#  4716#  4912   5644
SLASH    975#  4716#
SPACE   4716#
STARS    944#  4716#  4720   4935   4961   4987   5007   5038   5091   5138   5178   5217   5225   5242   5264
        5303   5320   5361   5372   5413   5457   5468   5510   5545   5624   5678   5758   5759   5760   5761
        5762   5763   5764   5765   5766   5768
SWRSU    918#  4895#
TRMTRP  5768#
TYPBIN  1468#  4716#
TYPDEC  1438#  4716#  5758   5761
TYPNAM  1226#  4702#  4896
TYPNUM  1405#  4716#
TYPOCS  1358#  4716#
TYPOCT  1321#  4716#  5761
TYPTXT  1275#  4716#
XERREN  5714#  5760
XSCOP   5742#  5759
$EGOLF  5704#  5761
$RESET  4709#  5307   5760
$SKIP   4705#  4994   5050   5060   5112   5115   5217   5231   5249   5276   5283   5292   5309   5361   5457
$$CMRE  4720#
$$CMTH  4720#
$$ESCA  1159#  4716#
$$NEWT  1113#  4716#  4935   4961   4987   5007   5038   5091   5138   5178   5217   5225   5242   5264   5303
        5320   5361   5372   5413   5457   5468   5510   5545   5624   5678
$$SET   5768#
$$SKIP  1193#  4716#  4912   5644
.EQUAT   166#  4702#  4716
.HEADE    39#  4703#  4713
.KT11    301#
.SETUP   737#  4703#  4893
.SWRHI    81#  4702#  4715
.SWRLO  4715#
.$ACT1  4102#
.$APTB  4150#
```

.SAPTH	4420#		
.SAPTY	4583#		
.SASTA	4467#		
.SCATC	478#	4702#	4717
.SCNTA	580#	4702#	4720
.SDB2D	3736#		
.SDB2O	3860#		
.SDIV	3631#		
.SEOP	1584#	4702#	5758
.SERRO	2002#	4702#	5760
.SERRT	2195#	4702#	5761
.SMULT	3567#		
.SPOWE	3280#		
.SRAND	3343#		
.SRDDE	2973#		
.SRDDC	2881#	4704#	5766
.SREAD	2665#	4704#	5765
.SR2A2	4004#		
.SSAVE	3049#		
.SSB2D	3821#		
.SSB2O	3923#		
.SSCOP	1794#	4703#	5759
.SSIZE	3405#		
.SSUPR	3961#		
.STRAP	3149#	4703#	5768
.STYPB	2580#		
.STYPD	2502#	4703#	5764
.STYPE	2283#	4703#	5762
.STYPO	2405#	4703#	5763
.S40CA	509#		

```
ADD     4925  4944  5110  5158  5166  5418  5457  5476  5515  5523  5555  5563  5637  5760  5761
        5762  5763  5764  5766  5866
ASL     5761  5766  5768
ASLB    5764
BCC     5764
BCS     5696
BEQ     4896  4912  4917  4973  4994  5050  5060  5107  5121  5149  5217  5231  5249  5276  5283
        5286  5292  5309  5361  5457  5568  5584  5657  5690  5758  5759  5760  5761  5762  5763
        5766  5826  5863
BGE     5759  5762
BGT     5758  5759  5763  5764
BHI     4919  5759
BIC     5056  5123  5217  5361  5382  5457  5479  5648  5662  5665  5686  5758  5763  5765  5766
        5813
BIS     5101  5307  5380  5383  5386  5477  5480  5483  5520  5635  5649  5666  5760  5763  5764
BISB    5761
BIT     5759  5760
BITB    5762
BLOS    5115  5765
BLT     5762  5763  5764
BMI     5392  5489  5528  5553  5561  5698  5764
BNE     4895  4896  4946  4998  5016  5055  5063  5075  5103  5157  5162  5168  5187  5234  5329
        5423  5524  5556  5564  5571  5573  5594  5601  5603  5614  5638  5661  5684  5701  5759
        5760  5761  5762  5763  5764  5765  5815  5819
BPL     5108  5112  5253  5575  5578  5588  5591  5597  5605  5608  5611  5760  5762  5763  5764
        5765
BR      4895  4896  4947  5017  5064  5077  5105  5117  5146  5159  5164  5188  5294  5330  5389
        5399  5424  5486  5496  5526  5644  5759  5761  5762  5763  5764  5765  5766
CLC     4921
CLR     4895  4898  4902  4904  4905  4906  4913  4968  5058  5070  5098  5119  5142  5143  5227
        5244  5267  5306  5378  5390  5396  5474  5487  5493  5516  5519  5521  5532  5548  5549
        5559  5606  5609  5612  5634  5636  5758  5759  5761  5763  5764  5766
CLRB    5759  5762  5764  5765
CMP     4895  4896  4918  4971  4977  4993  4996  5049  5078  5114  5124  5147  5148  5156  5165
        5167  5217  5230  5248  5275  5282  5291  5361  5395  5457  5492  5531  5554  5562  5567
        5583  5593  5656  5660  5683  5689  5700  5759  5760  5764  5765  5814  5818  5862
CMPB    5759  5762  5765
COM     5251
DEC     4945  4997  5015  5054  5062  5106  5120  5186  5328  5422  5572  5600  5602  5758  5761
DECB    4972  5762  5763
EMT     4716
HALT    4717  5760  5762  5864
INC     4894  4896  5074  5102  5233  5270  5582  5599  5613  5659  5758  5759  5760  5763  5764
INCB    5252  5271  5273  5287  5289  5391  5488  5527  5570  5759  5760  5762
IOT     4716
JMP     4717  4719  4927  5758  5760  5817
JSR     5758  5760  5762
MOV     4895  4897  4899  4900  4901  4903  4907  4908  4909  4916  4923  4926  4936  4937  4938
        4939  4940  4941  4949  4951  4952  4962  4963  4964  4965  4966  4967  4974  4978  4979
        4988  4989  4990  4991  4992  5008  5009  5010  5011  5012  5013  5019  5021  5022  5038
        5040  5041  5042  5043  5044  5045  5046  5047  5048  5052  5057  5065  5067  5068  5071
        5072  5079  5080  5092  5093  5094  5095  5096  5097  5100  5109  5118  5125  5126  5138
        5139  5140  5141  5153  5155  5169  5170  5179  5180  5181  5182  5183  5184  5190  5192
        5193  5217  5225  5226  5228  5229  5242  5243  5245  5246  5247  5265  5266  5268  5269
        5278  5281  5303  5304  5305  5308  5321  5322  5323  5324  5325  5326  5332  5334  5335
        5361  5374  5375  5376  5377  5379  5385  5393  5397  5398  5400  5401  5403  5414  5415
```

```
                5416  5417  5419  5420  5426  5428  5429  5457  5470  5471  5472  5473  5475  5482  5490
                5494  5495  5497  5498  5500  5512  5513  5514  5517  5518  5529  5533  5534  5545  5546
                5547  5566  5580  5581  5585  5586  5595  5624  5625  5626  5627  5628  5629  5630  5631
                5632  5633  5640  5641  5642  5643  5650  5663  5667  5678  5679  5681  5682  5685  5687
                5688  5758  5759  5760  5761  5762  5763  5764  5765  5766  5768  5812  5816  5821  5827
                5859  5861  5865  5867
MOVB            4895  4920  5272  5274  5279  5280  5288  5290  5558  5576  5579  5589  5592  5598  5647
                5655  5759  5760  5761  5762  5763  5764  5765  5766  5768
MTPS            4910  4976  5039  5073  5081  5099  5127  5361  5373  5404  5457  5469  5501  5511  5535
                5760  5862
NEG             5113  5763  5764
NOP             4969  5295  5381  5384  5387  5478  5481  5484  5654  5758
RESET           5307  5758  5760
ROL             4922  5693  5695  5763  5766
RTI             4950  4975  5020  5069  5122  5191  5333  5394  5427  5491  5530  5651  5664  5668  5759
                5760  5762  5763  5764  5765  5766  5829  5868
RTS             5761  5762  5768
SEC             5692  5694
SUB             5111  5680  5760  5764  5860
TRAP            5768
TST             4895  4911  4942  4943  5014  5059  5144  5145  5154  5160  5161  5185  5217  5327  5361
                5421  5457  5550  5551  5615  5697  5759  5760  5761  5762  5763  5764  5766  5768  5828
TSTB            5285  5552  5560  5574  5577  5587  5590  5596  5604  5607  5610  5759  5761  5762  5764
                5765  5825
.ASCII          4720  4859  4860  4861  4862
.ASCIZ          4720  4843  4844  4845  4846  4847  4848  4849  4850  4851  4852  4853  4854  4855  4856
                4857  4858  4864  4865  4866  4867  4868  4869  4870  4871  4872  4873  4874  4875  4876
                4896  5758  5761  5833  5834  5835  5836  5837
.BLKB           5765
.BLKW           5764
.BYTE           4720  4887  5758  5763  5765  5782  5783  5801
.ENABL             4  4701
.END            5890
.ENDC           4713  4715  4716  4717  4720  4893  4895  4896  4912  4935  4961  4987  5007  5038  5091
                5138  5178  5217  5225  5242  5264  5303  5320  5361  5372  5413  5457  5468  5510  5545
                5624  5644  5678  5758  5759  5760  5761  5762  5763  5764  5765  5766  5768
.EQUIV          4716
.EVEN           4877  4888  4896  5761  5838
.IF             4713  4715  4716  4717  4720  4893  4895  4896  4912  4935  4961  4987  5007  5038  5091
                5138  5178  5217  5225  5242  5264  5303  5320  5361  5372  5413  5457  5468  5510  5545
                5624  5644  5678  5758  5759  5760  5761  5762  5763  5764  5765  5766  5768
.IFF            4715  4716  4720  4895  4912  4935  4961  4987  5007  5038  5091  5138  5178  5217  5225
                5242  5264  5303  5320  5361  5372  5413  5457  5468  5510  5545  5624  5644  5678  5758
                5759  5760  5761  5762  5763  5764  5765  5766  5768
.IFT            4896  5759  5760  5765  5766
.IFTF           4896  5759  5760  5765  5766
.IIF            4713  4715  4717  4720  4895  4896  4994  5050  5060  5112  5115  5217  5231  5249  5276
                5283  5292  5309  5361  5457  5758  5759  5760  5761  5762  5765  5768
.IRP            4893  4935  4961  4987  5007  5038  5091  5138  5178  5195  5217  5225  5242  5264  5303
                5320  5337  5361  5372  5413  5431  5457  5468  5510  5545  5624  5678  5759  5760  5764
                5766  5773  5883  5886
.LIST              2  4693  4697  4700  4716  4717  4720  4842  4891  4892  4893  4895  4896  4935  4961
                4987  5007  5038  5091  5138  5178  5217  5225  5242  5264  5303  5320  5361  5372  5413
                5457  5468  5510  5545  5624  5678  5758  5759  5760  5765  5768  5777  5809  5810  5832
                5840  5841  5875  5883
.MACRO            39    81   166   301   478   509   580   737   783   797   825   918   944   975  1023
```

```
            1035    1079    1113    1146    1159    1180    1193    1226    1275    1321    1358    1405    1438    1468    1524
            1532    1584    1744    2002    2185    2263    2405    2502    2580    2665    2801    2973    3049    3149    3280
            3343    3405    3529    3567    3631    3736    3821    3860    3923    3961    4004    4102    4150    4420    4467
            4507    4583    4705    4709    4715    4720    4929    4955    4981    5000    5024    5083    5129    5171    5217
            5219    5236    5255    5297    5313    5361    5363    5406    5457    5459    5503    5537    5617    5670    5704
            5714    5743    5768
.MCALL      4702    4703    4704    4716    4895
.NLIST         1       3    4695    4696    4716    4717    4720    4840    4841    4890    4893    4895    4896    4935    4961
            4987    5007    5038    5091    5138    5178    5217    5225    5242    5264    5303    5320    5361    5372    5413
            5457    5468    5510    5545    5624    5678    5758    5759    5760    5765    5768    5771    5772    5807    5830
            5831    5839    5873    5883
.PAGE       4720    4816    4889    5767    5769    5811
.REPT       4717    4720    5876
.SBTTL      4715    4716    4717    4720    4935    4961    4987    5007    5038    5091    5138    5178    5217    5225    5242
            5264    5303    5320    5361    5372    5413    5457    5468    5510    5545    5624    5678    5758    5759    5760
            5761    5762    5763    5764    5765    5766    5768
.TITLE      4713
.WORD       4717    4720    4879    4880    4881    5758    5761    5762    5763    5766    5775    5778    5779    5780    5781
            5784    5785    5786    5787    5788    5789    5790    5791    5792    5793    5794    5795    5796    5797    5798
            5799    5802    5803    5804    5805
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

```
*,DWQAA/CRF=SYSMAC.B1,DWQAA.P11
RUN-TIME: 38 46 4 SECONDS
RUN-TIME RATIO: 326/89=3.6
CORE USED:  29K  (57 PAGES)
```