DL11-W

DIAGNOSTIC

MD-11-DZDLD-A

EP-DZDLD-A-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A.

.REM ↑A

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDLD-A

PRODUCT NAME: DL11-W DIAGNOSTIC

DATE CREATED: JUNE 11, 1976

MAINTAINER: DIAGNOSTIC ENGINEERING (CC301)

AUTHOR: DAN CASALETTO

1.0 GENERAL INFORMATION

1.1 ABSTRACT

THIS DIAGNOSTIC IS A LOGIC TEST TO VERIFY THE OPERATION OF THE DL11-W SERIAL LINE/ REAL TIME CLOCK INTERFACE. THE PROGRAM WILL RUN WITHOUT ANY SPECIAL TEST FIXTURES BY DEFAULT. HOWEVER, A WRAP CABLE CAN BE USED AND TESTED BY OPTION (BIT7 OF SWR).

THIS TEST OPERATES ON THE CONSOLE DL11-W SERIAL LINE AND CLOCK INTERFACES AS WELL AS UP TO FIFTEEN(15) ADDITIONAL DL11-W SERIAL LINE INTERFACES. THE DEFAULT ADDRESSES ARE:

A. CONSOLE - 177560 SERIAL LINE
             177546 CLOCK

B. OTHER SERIAL LINE - 776500 FIRST SERIAL LINE ADDRESS OF 15 CONSECUTIVE SERIAL LINE ADDRESSES

THIS PROGRAM IS DESIGNED TO RUN ON ANY PDP-11 WITH 8K OF MEMORY AND A DL11-W MODULE. IT CAN BE RUN UNDER XXDP,APT, AND ACT MONITORS, AND ON PROCESSORS WITH NO HARDWARE SWITCH REGISTER, SOFTWARE SWITCH REGISTER = LOCATION 176

POWER FAILURE IS SUPPORTED FOR SYSTEMS WITH CORE MEMORY.

1.2 SYSTEM REQUIREMENTS

1.2.1 EQUIPMENT

STANDARD 11 FAMILY COMPUTER WITH A CONSOLE OUTPUT DEVICE AND 8K OF MEMORY.

1.2.2 STORAGE

THE PROGRAM USES MEMORY FROM 00000 TO 2151C.

1.3 ASSUMPTIONS

A. IF THE UNIT UNDER TEST (UUT) IS THE CONSOLE , THE PROGRAM WILL ASSUME THE REAL TIME CLOCK (RTC) IS ENABLED AND WILL TEST IT UNLESS THE TESTS ARE DISABLED BY BIT6 OF THE SWR.

B. IF THE UUT IS NOT THE CONSOLE, THE RTC IS NOT TESTED FOR THAT DEVICE.

C. THE PROGRAM WILL ASSUME THE ERROR FLAG BITS AND THE BREAK FUNCTION OF THE DL11-W ARE DISABLED AND WILL NOT TEST THESE FUNCTIONS UNLESS ENABLED BY BIT10 (FOR ERROR FLAGS) AND BIT8 (FOR BREAK) OF THE SWR.

2.0 OPERATING INSTRUCTIONS

2.1 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED TAPES.

2.2 STARTING PROCEDURE

LOAD THE SWITCH REGISTER WITH SETTING
(SOFTWARE SWITCH REGISTER LOCATION = 176)

A. START AT 200.
AFTER CHECKING THE TRANSMITTER, THE PROGRAM WILL PRINT ITS IDENTIFICATION AND REPORT THE NUMBER OF DEVICES UNDER TEST (NUMBER IS OCTAL).
"END PASS" IS PRINTED AFTER A FULL PASS HAS BEEN MADE ON ALL DEVICES UNDER TEST.

B. START AT 204.
THE "ECHO" TEST WILL BE EXECUTED. AN "*" IS PRINTED AT THE BEGINNING OF THE TEST. THE ECHO TEST READS A CHARACTER FROM THE TERMINAL,WRITES THAT CHARACTER TO THE TERMINAL AND REPORTS ANY ERROR FLAGS SET IN THE RECEIVER BUFFER.
A CONTROL-C HALTS THE TEST AND PRINTS "STOP" AT THE TERMINAL
CONTINUING RESTARTS THE ECHO TEST.

C. START AT 210.
THE TERMINAL OUTPUT TEST WILL BE EXECUTED. DEPRESSING ANY CHARACTER AT THE TERMINAL, HALTS THE TEST. CONTINUING RESTARTS THE TEST.
THE TEST OUTPUTS 32 CHARACTERS ON A LINE AND REPEATS THE PATTERN EVERY THREE LINES. THE PATTERN IS AS FOLLOWS
(OCTAL CODE 040 --> 377):

	(OCTAL CODE)
!"#$%&'()*+,-./0123456789:;<=>?	(040 --> 077)
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←	(100 --> 137)
`abcdefghijklmnopqrstuvwxyz{|	(140 --> 177) [LOWER CASE ALPHA]

THIS BOTTOM LINE COULD BE THE FOLLOWING IF THE TERMINAL DOES NOT HAVE LOWER CASE:

E01

@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\ [UPPER CASE ALPHA] SEQ 0004

2.3 OPERATING PROCEDURE

2.3.1 OPERATIONAL SWITCH SETTINGS

NOTE: IF NO HARDWARE SWITCH REGISTER IS AVAILABLE THE PROGRAM WILL AUTOMATICALLY USE THE CONTENTS OF LOCATION 176 AS THE SOFTWARE SWITCH REGISTER. THE USER SHOULD SET THIS LOCATION BEFORE STARTING THE PROGRAM. IF A HARDWARE SWITCH IS AVAILABLE AND A SOFTWARE SWR IS DESIRED, PUT ALL SWITCHES UP.

BIT15	- HALT ON ERROR
BIT14	- SCOPE LOOP
BIT13	- INHIBIT ERROR TYPEOUT
BIT12	- UNUSED
BIT11	- UNUSED
BIT10	- ENABLE ERROR FLAGS TESTS
BIT09	- LOOP ON ERROR
BIT08	- ENABLE BREAK FUNCTION TESTS
BIT07	- ENABLE DATA TEST WITH WRAP CABLE
BIT06	- INHIBIT RTC TESTS
BIT05	- ALLOW MANUAL SETTING OF "SDEVM" (DEVICE MAP)

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS. BECAUSE THIS DIAGNOSTIC USES THE MAINTENANCE BIT OF THE SERIAL LINE, THE CONTROL-G SHOULD BE ISSUED DURING PROGRAM TYPEOUTS AT THAT TIME THE MAINTENANCE BIT IS SURE TO BE CLEAR.

IF A CONTROL-G IS DETECTED, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

```
(IE)    SWR=XXXXXX  NEW=
```

POSSIBLE RESPONSES ARE:

1.	<CR>	IF NO CHANGES ARE TO BE MADE
2.	& DIGITS 0-7	TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ;LAST DIGIT FOLLOWED BY <CR>.
3.	↑U	TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

2.3.2 RUNNING UNDER APT

THE APT MAILBOX IS LOCATED AT LOCATION 500, TO ALLOW ADDITIONAL SERIAL LINE VECTOR ASSIGNMENTS TO THE 400 AREA OF MEMORY.

THE EXECUTION TIMES PROVIDED ARE FOR EXECUTION WITH AN 11/34 PROCESSOR, CORE MEMORY, AND 110 BAUD.

2.3.3 RUN WITH ALTERNATE CONSOLE ADDRESS

TO USE A CONSOLE ADDRESS OTHER THAN 177560, OR VECTOR OTHER THAN 60, THE OPERATOR MUST SUPPLY THE PROGRAM WITH THE CORRECT ADDRESSES BY INSERTING THEM AT THE TAG LABELED "CRCSR":

```
CRCSR:  ADDRESS OF RECEIVER RCSR
CRBUF:  ADDRESS OF RECEIVER BUFFER
CTCSR:  ADDRESS OF TRANSMITTER CSR
CTBUF:  ADDRESS OF TRANSMITTER BUFFER
CRVECT: ADDRESS OF RECEIVER VECTOR
CRPSW:  ADDRESS OF ASSOCIATED PSW
CTVECT: ADDRESS OF TRANSMITTER VECTOR
CTPSW:  ADDRESS OF ASSOCIATED PSW
```

2.3.4 TESTING ADDITIONAL SERIAL LINES

THIS PROGRAM WILL SUPPORT TESTING OF MULTIPLE SLU'S. IT REQUIRES THE ADDRESS OF THE FIRST ADDITIONAL RCSR (STORED AT "$BASE") AND ITS INTERRUPT VECTOR (STORED AT "$VECT1"); AND WILL BE ABLE TO ADDRESS ANY SLU STARTING AT THE SPECIFIED BASE ADDRESS UP TO 15 CONSECUTIVE DEVICES.

```
EXAMPLE:        $BASE:  776500
                $VECT1: 300
```

THE PROGRAM WILL BE ABLE TO TEST THE CONSOLE PLUS ANY ADDITIONAL DL11-W SLU'S WITHIN THE RANGE 776500 --> 776660

$BASE AND $VECT1 DEFAULT TO 776500 AND 300 RESPECTIVELY.

THE PROGRAM ASSOCIATES UNIT NUMBERS TO DEVICES AS FOLLOWS:
(NUMBERS IN PARENTHESIS ARE OCTAL)

```
        UNIT# 0 -->  CONSOLE [ADDRESS STORED AT "%RCSR"]
        UNIT# 1 -->  BASE ADDRESS STORED AT "$BASE"
                     ASSOCIATED BASE VECTOR STORED AT"$VECT1"
        UNIT# 2 -->  BASE ADDRESS + (10)
                     BASE VECTOR + (10)
        UNIT# 3 -->  BASE ADDRESS + (20)
                     BASE VECTOR + (20)
        UNIT# 4 -->  BASE ADDRESS + (30)
                     BASE VECTOR + (30)
                  :
                  :
                  :
                  :
                  V
        UNIT#15 -->  BASE ADDRESS + (160)
                     BASE VECTOR + (160)
```

STARTING AT LOCATION 200 AND HAVING BITS OF SWR CLEAR, THE PROGRAM WILL SELF SIZE THE NUMBER OF DEVICES (STARTING AT THE E ; ADDRESS) AND STORE A BIT MAP AT "$DEVM" (DEVICE MAP) TO INDICATE WHICH UNIT NUMBERS ARE PRESENT AND WILL BE TESTED:

UNIT 15	UNIT 14	.....	UNIT 2	UNIT 1	CONSOLE

A BIT MAP CAN BE ENTERED AT "$DEVM" PRIOR TO STARTING THE PROGRAM SETTING BITS OF THE SWR INHIBITS THE SELF-SIZING AND DEVICE MAP GENERATION, AND USES THE VALUE STORED BY THE OPERATOR.

EXAMPLE:

```
SWR   =  000040          [BINARY  0 000 000 000 100 000]
$BASE:   776500
$VECT1:  300

$DEVM:   13      [BINARY - 0 000 000 000 001 011]

THE PROGRAM WILL TEST -
        UNIT# 0 = CONSOLE
        UNIT# 1 = 776500 ; 300
        UNIT# 3 = 776520 ; 320
```

2.4 EXECUTION TIMES - (110 BAUD)

```
LONGEST SUBTEST TIME = 50 SECONDS
PASS TIME            = 60 SECONDS
ADDITIONAL DEVICES   = 55 SECONDS/DEVICE
```

3.0 ERROR REPORTING

IF A ROUTINE FAILS AND THE INHIBIT ERROR TYPEOUT (BIT13) OF THE SWR IS NOT SET, A PRINTOUT RESULTS IN THE FORM:

```
"(SOME ASCII MESSAGE)"
TEST#   ERR PC   RCSR    [ANY APPLICABLE DAT HEADINGS]
XXXXXX  XXXXXX   XXXXXX   [ANY APPLICABLE DATA]

        NOTE: "RCSR" IS DEPENDENT ON THE FAILURE
              & THEREFORE COULD BE TCSR,RBUF,TBUF,OR LKS
```

WHERE "XXXXXX" IS AN OCTAL NUMBER.
THIS ERROR PRINTOUT OCCURS PROVIDED THE ERROR THAT EXISTS WOULD NOT HINDER THE TYPEOUT. IN CASES WHERE IT IS NOT POSSIBLE TO PRINT AN ERROR MESSAGE (I.E. FATAL CONSOLE TRANSMITTER FAILURES), A HALT OCCURS AND THE PC CAN BE EXAMINED BY THE OPERATOR TO FIND THE ERROR INFORMATION IN THE PROGRAM LISTING.

NOTE: FOR SOFTWARE SWITCH OPERATION, THE SWITCH REGISTER CAN BE CHANGED BY TYPING A CONTROL-G AT THE CONSOLE DURING ERROR PRINTOUTS. AFTER CONTINUING FROM THE ERROR HALT THE OLD SWR CONTENTS IS DISPLAYED AND THE NEW CONTENTS CAN BE ENTERED.
IF ERROR HALTS ARE DISABLED, THE CONTROL-G RESPONSE OCCURS IMMEDIATELY FOLLOWING THE TYPEOUT.

4.0 SUBROUTINE ABSTRACTS

4.1 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0 DESIGNED TO
DETECT AND ISOLATE UNEXPECTED TRAPS TO THE TRAP AND INTERRUPT VECTOR
AREA OF MEMORY.

EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS OF THE NEXT
LOCATION. THE NEXT LOCATION IS LOADED WITH A BREAK POINT TRAP (000003).
THUS AN ILLEGAL TRAP OR INTERRUPT CAUSES A TRAP THROUGH THE BPT VECTOR(14)
WHICH POINTS TO THE "CATCH" ROUTINE.

THE "CATCH" ROUTINE REPORTS THE PC THAT CAUSED THE ORIGINAL TRAP AND
THE LOCATION OF THE TRAP VECTOR (IF UNDER APT, AN ERROR IS INDICATED TO APT).
AFTER REPORTING THE ERROR THE PROGRAM HALTS. THE PROGRAM MUST BE RESTARTED
AT THIS POINT.

4.2 WRPSW

THIS ROUTINE IS USED TO WRITE THE PSW BY POPPING VALUES FROM THE
STACK. THIS METHOD IS USED TO BE COMPATIBLE WITH ALL 11 FAMILY
PROCESSORS.

4.3 SCOPE

THIS ROUTINE CALL IS PLACED BETWEEN EACH SUBTEST. IT RECORDS THE STARTING
ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED AND UPDATES THE
TEST NUMBER. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE
SUBTEST AT WHICH THE SCOPE LOOP IS REQUESTED.

4.4 ERROR

THIS ROUTINE CALL IS PLACED WHEREEVER AN ERROR REPORT
IS DESIRED. THE LOWER BYTE OF THIS CALL IS USED AS THE ERROR NUMBER
AND AS A POINTER INTO THE ERROR TABLE . THIS ROUTINE REPORTS ERRORS
TO APT USING "$APTYPE" AND TYPES ERROR REPORTS TO THE CONSOLE
USING "$ERRTYPE".

4.5 $POWER

THIS ROUTINE SAVES ALL GENERAL REGISTERS DURING POWER-DOWN
AND RESTORES THEM AT POWER-UP. IF A POWER FAILURE OCCURS
"POWER" IS PRINTED AT THE CONSOLE AFTER POWER IS RESTORED.

4.6 CKSWR

THIS ROUTINE CALL IS USED TO DETECT THE RECEPTION OF A CONTROL-G
FROM THE CONSOLE. THE CALL USES "$READ" TO PERFORM THE CONTROL-G
SEQUENCE OF DISPLAYING THE CONTENTS OF THE SOFTWARE SWITCH REGISTER
AND THE ENTERING THE NEW CONTENTS FROM THE TERMINAL.

↑A

```
                        .ENABLE ABS
                        .LIST   ME
                        .NLIST  MC,MD,CND
                        .MCALL  .STYPE,.STYPOC,.STRAP
                        .MCALL  .SETUP,STARS,PUSH,POP,SETUP,.EQUAT, SERRTYP
                        .MCALL  .SAPTHDR,.SAPTBLS,.SACT11,.$SCOPE
                        .MCALL  .SCMTAG,.SEOP,.SREAD
                        .SBTTL  BASIC DEFINITIONS

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

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

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

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

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
```

```
                        SW12=   10000
                        SW11=   4000
                        SW10=   2000
                        SW09=   1000
                        SW08=   400
                        SW07=   200
                        SW06=   100
                        SW05=   40
                        SW04=   20
                        SW03=   10
                        SW02=   4
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

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

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
```

```
                        000014          TRTVEC= 14              ;[illegible]TRAP
                        000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
                        000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                        000024          PWRVEC= 24              ;;POWER FAIL
                        000030          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
                        000034          TRAPVEC=34              ;;"TRAP" TRAP
                        000060          TKVEC=  60              ;;TTY KEYBOARD VECTOR
                        000064          TPVEC=  64              ;;TTY PRINTER VECTOR
                        000240          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
                        176500                  ABASE=  176500
                        000300                  AVECT1= 300
                        000001                  $TN=    1
                        161000                  $SWR=   161000
                        000003                  BPT=    000003          ;THIS IS THE COMMAND FOR A TRAP
                                                                        ;   THROUGH 14 (BPT TRAP)

                        000000                  .=0
                                        ;;**************************************************************
                                                ;*ALL UNUSED LOCATIONS FROM 4-776 CONTAIN A ".+2,BPT"
                                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS & INTERRUPTS
                                                ;*LOCATION " CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS


                        000014                  .=14                    ;THE BPT TRAP VECTOR POINTS TO THE
000014  011602                  .WORD   CATCH           ;   ILLEGAL TRAP HANDLER "CATCH"
000016  000340                  .WORD   340

                        000042                  .=      42
000042  000000                  .WORD   0


                        000174                  .=      174
000174  000000          DISPREG:.WORD   0
000176  000000          SWREG:  .WORD   0

                        000200                  .=200
000200  000167  002302          JMP     START           ;DO INTERFACE TEST
000204  000167  014100          JMP     ECHO            ;DO ECHO TEST
000210  000167  014314          JMP     OUTTST          ;DO OUTPUT TEST TO TERMINAL
```

```
                        000500                          .=      500
                                                .SBTTL  ACT11 HOOKS

                                                ;;**************************************************************
                                                ;HOOKS REQUIRED BY ACT11
                        000500                          $SVPC=.                 ;SAVE PC
                        000046                          .=46
000046  011500                                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
                        000052                          .=52
000052  000000                                          .WORD   0               ;;2)SET LOC.52 TO ZERO
                        000500                          .=$SVPC                 ;; RESTORE PC
                                                .SBTTL  APT PARAMETER BLOCK

                                                ;;**************************************************************
                                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                                ;;**************************************************************
                        000500                          .$X=.   ;;SAVE CURRENT LOCATION
                        000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                                          200     ;;FOR APT START UP
                        000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  000500                                          $APTHDR ;;POINT TO APT HEADER BLOCK
                        000500                          .=.$X   ;;RESET LOCATION COUNTER
                                                ;;**************************************************************
                                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                                ;INTERFACE SPEC.

000500                                          $APTHD:
000500  000000                                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
000502  001066                                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
000504  000050                                  $TSTM:  .WORD   50      ;;RUN TIM OF LONGEST TEST
000506  000060                                  $PASTM: .WORD   60      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
000510  000055                                  $UNITM: .WORD   55      ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
000512  000030                                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

C02


```
                        .SBTTL  COMMON TAGS

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

001000                  .=1000
001000          $CMTAG:                         ;;START OF COMMON TAGS
001000  000000          .WORD   0
001002  000             $TSTNM: .BYTE   0       ;;CONTAINS THE TEST NUMBER
001003  000             $ERFLG: .BYTE   0       ;;CONTAINS ERROR FLAG
001004  000C30          $ICNT:  .WORD   0       ;;CONTAINS SUBTEST ITERATION COUNT
001006  000000          $LPADR: .WORD   0       ;;CONTAINS SCOPE LOOP ADDRESS
001010  000000          $LPERR: .WORD   0       ;;CONTAINS SCOPE RETURN FOR ERRORS
001012  000000          $ERTTL: .WORD   0       ;;CONTAINS TOTAL ERRORS DETECTED
001014  000             $ITEMB: .BYTE   0       ;;CONTAINS ITEM CONTROL BYTE
001015  001             $ERMAX: .BYTE   1       ;;CONTAINS MAX. ERRORS PER TEST
001016  000000          $ERRPC: .WORD   0       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001020  000000          $GOOADR: .WORD  0       ;;CONTAINS ADDRESS OF 'GOOD' DATA
001022  000000          $BADADR: .WORD  0       ;;CONTAINS ADDRESS OF 'BAD' DATA
001024  000000          $GOODAT: .WORD  0       ;;CONTAINS 'GOOD' DATA
001026  000000          $BADDAT: .WORD  0       ;;CONTAINS 'BAD' DATA
001030  000000                  .WORD   0       ;;RESERVED--NOT TO BE USED
001032  000000                  .WORD   0
001034  000             $AUTOB: .BYTE   0       ;;AUTOMATIC MODE INDICATOR
001035  000             $INTAG: .BYTE   0       ;;INTERRUPT MODE INDICATOR
001036  000000                  .WORD   0
001040  177570          SWR:    .WORD   DSWR    ;;ADDRESS OF SWITCH REGISTER
001042  177570          DISPLAY: .WORD  DDISP   ;;ADDRESS OF DISPLAY REGISTER
001044  177560          $TKS:   177560          ;;TTY KBD STATUS
001046  177562          $TKB:   177562          ;;TTY KBD BUFFER
001050  177564          $TPS:   177564          ;;TTY PRINTER STATUS REG. ADDRESS
001052  177566          $TPB:   177566          ;;TTY PRINTER BUFFER REG. ADDRESS
001054  000             $NULL:  .BYTE   0       ;;CONTAINS NULL CHARACTER FOR FILLS
001055  002             $FILLS: .BYTE   2       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001056  012             $FILLC: .BYTE   12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001057  000             $TPFLG: .BYTE   0       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001060  000C30          $ESCAPE:0               ;;ESCAPE ON ERROR ADDRESS
001062  077             $QUES:  .ASCII  /?/     ;;QUESTION MARK
001063  015             $CRLF:  .ASCII  <15>    ;;CARRIAGE RETURN
001064  000012          $LF:    .ASCIZ  <12>    ;;LINE FEED
                        ;;**********************************************************************
                        .SBTTL  APT MAILBOX-ETABLE

                        ;;**********************************************************************
                        .EVEN
001066          $MAIL:                          ;;APT MAILBOX
001066  000000  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001070  000000  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001072  000000  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001074  000000  $PASS:  .WORD   APASS           ;;PASS COUNT
001076  000000  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001100  000000  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001102  000000  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
001104  000000  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
001106          $ETABLE:                        ;;APT ENVIRONMENT TABLE
```

D02


```
   245  001106  000                $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
   246  001107  000                $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
   247  001110  000000             $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
   248  001112  000000             $USWR:  .WORD   AUSWR   ;;USER SWITCHES
   249  001114  000000             $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
   250                                                     BITS 15-11=CPU TYPE
   251                                                            11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
   252                                                            11/70=06,PDQ=07,Q=10
   253                                                     BIT 10=REAL TIME CLOCK
   254                                                     BIT  9=FLOATING POINT PROCESSOR
   255                                                     BIT  8=MEMORY MANAGEMENT
   256  001116  000                $AMAMS1: .BYTE  AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
   257  001117  000                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
   258                                                     MEM.TYPE BYTE   --  (HIGH BYTE)
   259                                                            900 NSEC CORE=001
   260                                                            300 NSEC BIPOLAR=002
   261                                                            500 NSEC MOS=003
   262  001120  000000             $MAOR1: .WORD   AMAOR1  ;;HIGH ADDRESS,BLK#1
   263                                                       MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
   264  001122  000                $AMAMS2: .BYTE  AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
   265  001123  000                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
   266  001124  000000             $MAOR2: .WORD   AMAOR2  ;;MEM.LAST ADDRESS,BLK#2
   267  001126  000                $AMAMS3: .BYTE  AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
   268  001127  000                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
   269  001130  000000             $MAOR3: .WORD   AMAOR3  ;;MEM.LAST ADDRESS,BLK#3
   270  001132  000                $AMAMS4: .BYTE  AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
   271  001133  000                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
   272  001134  000000             $MAOR4: .WORD   AMAOR4  ;;MEM.LAST ADDRESS,BLK#4
   273  001136  000300             $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
   274  001140  000000             $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
   275  001142  176500             $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
   276  001144  000000             $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
   277  001146                     $ETEND:
   278                             .MEXIT
```

E02


```
                        .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 DF A
                        ;*      DF              ;;POINTS TO THE DATA FORMAT

001146                  $ERRTB:

001146                  .$ERRTB:
001146  014614                 EM1                     ;"CAN NOT ACCESS TCSR"
001150  020575                 DH1                     ;"TEST# ERR PC  TCSR"
001152  021340                 DT1                     ;$TESTN,$ERRPC,TCSR
001154  000000                 0

001156  014640                 EM2                     ;"CAN NOT ACCESS TBUF"
001160  020622                 DH2                     ;"TEST# ERR PC  TBUF"
001162  021350                 DT2                     ;$TESTN,$ERRPC,TBUF
001164  000000                 0

001166  014664                 EM3                     ;"TCSR DONE NOT CLEARED WITH TBUF FULL"
001170  020575                 DH1                     ;"TEST# ERR PC  TCSR"
001172  021340                 DT1                     ;$TESTN,$ERRPC,TCSR
001174  000000                 0

001176  014731                 EM4                     ;"TCSR DONE NOT SET"
001200  020575                 DH1                     ;"TEST# ERR PC  TCSR"
001202  021340                 DT1                     ;$TESTN,$ERRPC,TCSR
001204  000000                 0

001206  014753                 EM5                     ;TCSR DONE NOT SET WITH RESET
001210  020575                 DH1                     ;"TEST# ERR PC  TCSR"
001212  021340                 DT1                     ;$TESTN,$ERRPC,TCSR
001214  000000                 0

001216  015010                 EM6                     ;"CAN NOT ACCESS RCSR"
001220  020647                 DH6                     ;"TEST# ERR PC  RCSR"
001222  021360                 DT6                     ;$TESTN,$ERRPC,RCSR
001224  000000                 0

001226  015034                 EM7                     ;"CAN NOT ACCESS RBUF"
001230  020674                 DH7                     ;"TEST# ERR PC  RBUF"
001232  021370                 DT7                     ;$TESTN,$ERRPC,RBUF
001234  000000                 0

001236  015060                 EM10                    ;"CAN NOT ACCESS LKS"
001240  020721                 DH10                    ;"TEST# ERR PC  LKS"
001242  021400                 DT10                    ;$TESTN,$ERRPC,LKS
001244  000000                 0
```

F02


```
                335
001246  015103         335     EM11    ;"BIT0 OF TCSR NOT CLEAR AFTER RESET"
001250  020575         336     DH1     ;"TEST# ERR PC  TCSR
001252  021340         337     DT1     ;$TESTN,$ERRPC,TCSR
001254  000000         338     0
                339
001256  015146         340     EM12    ;"CAN NOT SET BIT0 OF TCSR"
001260  020575         341     DH1     ;"TEST# ERR PC  TCSR"
001262  021340         342     DT1     ;$TESTN,$ERRPC,TCSR
001264  000000         343     0
                344
001266  015177         345     EM13    ;"CAN NOT CLEAR BIT0 OF TCSR"
001270  020575         346     DH1     ;"TEST# ERR PC  TCSR"
001272  021340         347     DT1     ;$TESTN,$ERRPC,TCSR
001274  000000         348     0
                349
001276  015232         350     EM14    ;"RESET DID NOT CLEAR BIT0 OF TCSR"
001300  020575         351     DH1     ;"TEST# ERR PC  TCSR"
001302  021340         352     DT1     ;$TESTN,$ERRPC,TCSR
001304  000000         353     0
                354
001306  015273         355     EM15    ;"BIT2 OF TCSR NOT CLEAR AFTER RESET"
001310  020575         356     DH1     ;"TEST# ERR PC  TCSR"
001312  021340         357     DT1     ;$TESTN,$ERRPC,TCSR
001314  000000         358     0
                359
001316  015336         360     EM16    ;"CAN NOT SET BIT2 OF TCSR"
001320  020575         361     DH1     ;"TEST# ERR PC  TCSR"
001322  021340         362     DT1     ;$TESTN,$ERRPC,TCSR
001324  000000         363     0
                364
001326  015367         365     EM17    ;"CAN NOT CLEAR BIT2 OF TCSR"
001330  020575         366     DH1     ;"TEST# ERR PC  TCSR"
001332  021340         367     DT1     ;$TESTN,$ERRPC,TCSR
001334  000000         368     0
                369
001336  015422         370     EM20    ;"RESET DID NOT CLEAR BIT2 OF TCSR"
001340  020575         371     DH1     ;"TEST# ERR PC  TCSR"
001342  021340         372     DT1     ;$TESTN,$ERRPC,TCSR
001344  000000         373     0
                374
001346  015463         375     EM21    ;"BIT6 OF TCSR NOT CLEAR AFTER RESET2
001350  020575         376     DH1     ;"TEST# ERR PC  TCSR"
001352  021340         377     DT1     ;$TESTN,$ERRPC,TCSR
001354  000000         378     0
                379
001356  015527         380     EM22    ;"XMIT INTERRUPT WITH PRIORITY 7"
001360  020575         381     DH1     ;"TEST# ERR PC  TCSR"
001362  021340         382     DT1     ;$TESTN,$ERRPC,TCSR
001364  000000         383     0
                384
001366  015564         385     EM23    ;"CAN NOT SET BIT6 OF TCSR"
001370  020575         386     DH1     ;"TEST# ERR PC  TCSR"
001372  021340         387     DT1     ;$TESTN,$ERRPC,TCSR
001374  000000         388     0
                389
```

```
   391  001376  015615                  EM24            ;"CAN NOT CLEAR BIT6 OF TCSR"
   392  001400  020575                  DH1             ;"TEST# ERR PC  TCSR"
   393  001402  021340                  DT1             ;$TESTN,$ERRPC,TCSR
   394  001404  000000                  0
   395
   396  001406  015650                  EM25            ;"RESET DID NOT CLEAR BIT6 OF TCSR"
   397  001410  020575                  DH1             ;"TEST# ERR PC  TCSR"
   398  001412  021340                  DT1             ;$TESTN,$ERRPC,TCSR
   399  001414  000000                  0
   400
   401  001416  015711                  EM26            ;"BIT6 OF RCSR NOT CLEAR AFTER RESET"
   402  001420  020647                  DH6             ;"TEST# ERR PC  RCSR"
   403  001422  021360                  DT6             ;$TESTN,$ERRPC,RCSR
   404  001424  000000                  0
   405
   406  001426  015754                  EM27            ;"RCVR INTERRUPT WITH PRIORITY 7"
   407  001430  020647                  DH6             ;"TEST# ERR PC  RCSR"
   408  001432  021360                  DT6             ;$TESTN,$ERRPC,RCSR
   409  001434  000000                  0
   410
   411  001436  016013                  EM30            ;"CAN NOT SET BIT6 OF RCSR"
   412  001440  020647                  DH6             ;"TEST# ERR PC  RCSR"
   413  001442  021360                  DT6             ;$TESTN,$ERRPC,RCSR
   414  001444  000000                  0
   415
   416  001446  016044                  EM31            ;"CAN NOT CLEAR BIT6 OF RCSR"
   417  001450  020647                  DH6             ;"TEST# ERR PC  RCSR"
   418  001452  021360                  DT6             ;$TESTN,$ERRPC,RCSR
   419  001454  000000                  0
   420
   421  001456  016077                  EM32            ;"CAN NOT CLEAR BIT6 OF RCSR WITH RESET2
   422  001460  020647                  DH6             ;"TEST# ERR PC  RCSR"
   423  001462  021360                  DT6             ;$TESTN,$ERRPC,RCSR
   424  001464  000000                  0
   425
   426  001466  016145                  EM33            ;"BIT6 OF LKS NOT CLEAR AFTER RESET"
   427  001470  020721                  DH10            ;"TEST# ERR PC  LKS"
   428  001472  021400                  DT10            ;$TESTN,$ERRPC,LKS
   429  001474  000000                  0
   430
   431  001476  016207                  EM34            ;"LKS INTERRUPT WITH PRIORITY 7"
   432  001500  020721                  DH10            ;"TEST# ERR PC  LKS"
   433  001502  021400                  DT10            ;$TESTN,$ERRPC,LKS
   434  001504  000000                  0
   435
   436  001506  016245                  EM35            ;"CAN NOT SET BIT6 OF LKS"
   437  001510  020721                  DH10            ;"TEST# ERR PC  LKS"
   438  001512  021400                  DT10            ;$TESTN,$ERRPC,LKS
   439  001514  000000                  0
   440
   441  001516  016275                  EM36            ;"CAN NOT CLEAR BIT6 OF LKS"
   442  001520  020721                  DH10            ;"TEST# ERR PC  LKS"
   443  001522  021400                  DT10            ;$TESTN,$ERRPC,LKS
   444  001524  000000                  0
   445
   446  001526  016327                  EM37            ;"RESET DID NOT CLEAR BIT6 OF LKS"
```

```
447  001530  020721                    DH10        ;"TEST# ERR PC  LKS"
448  001532  021400                    DT10        ;$TESTN,$ERRPC,LKS
449  001534  000000                    0
450
451  001536  016367                    EM40        ;"DUAL ADDRESSING ERROR"
452  001540  020745                    DH40        ;"TEST# ERR PC  GOOD    BAD"
453  001542  021410                    DT40        ;$TESTN,$ERRPC,RCSR,BDCSR
454  001544  000000                    0
455
456  001546  016415                    EM41        ;"BIT7 OF LKS NOT SET AFTER RESET2
457  001550  020721                    DH10        ;"TEST# ERR PC  LKS"
458  001552  021400                    DT10        ;$TESTN,$ERRPC,LKS
459  001554  000000                    0
460
461  001556  016455                    EM42        ;"CAN NOT CLEAR BIT7 OF LKS"
462  001560  020721                    DH10        ;"TEST# ERR PC  LKS"
463  001562  021400                    DT10        ;$TESTN,$ERRPC,LKS
464  001564  000000                    0
465
466  001566  016507                    EM43        ;"BIT7 OF LKS DOES NOT SET"
467  001570  020721                    DH10        ;"TEST# ERR PC  LKS"
468  001572  021400                    DT10        ;$TESTN,$ERRPC,LKS
469  001574  000000                    0
470
471  001576  016540                    EM44        ;"RTC INTERRUPT AT PRIORITY 7"
472  001600  020721                    DH10        ;"TEST# ERR PC  LKS"
473  001602  021400                    DT10        ;$TESTN,$ERRPC,LKS
474  001604  000000                    0
475
476  001606  016574                    EM45        ;"RTC INTERRUPTS WHEN DISABLED"
477  001610  020721                    DH10        ;"TEST# ERR PC  LKS"
478  001612  021400                    DT10        ;$TESTN,$ERRPC,LKS
479  001614  000000                    0
480
481  001616  016631                    EM46        ;"RTC INTERRUPT DID NOT OCCUR"
482  001620  020721                    DH10        ;"TEST# ERR PC  LKS"
483  001622  021400                    DT10        ;$TESTN,$ERRPC,LKS
484  001624  000000                    0
485
486  001626  016631                    EM47        ;"RTC INTERRUPT DID NOT OCCUR"
487  001630  020721                    DH10        ;"TEST# ERR PC  LKS"
488  001632  021400                    DT10        ;$TESTN,$ERRPC,LKS
489  001634  000000                    0
490
491  001636  016665                    EM50        ;"RTC DOUBLE INTERRUPT"
492  001640  020721                    DH10        ;"TEST# ERR PC  LKS"
493  001642  021400                    DT10        ;$TESTN,$ERRPC,LKS
494  001644  000000                    0
495
496  001646  016712                    EM51        ;"RESET DID NOT CLEAR RTC INTERRUPT"
497  001650  020721                    DH10        ;"TEST# ERR PC  LKS"
498  001652  021400                    DT10        ;$TESTN,$ERRPC,LKS
499  001654  000000                    0
500
501  001656  016742                    EM52        ;"RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS"
502  001660  020721                    DH10        ;"TEST# ERR PC  LKS"
```

```
  503  001662  021400                    DT10            ;$TESTN,$ERRPC,LKS
  504  001664  000000                    0
  505
  506  001666  017017                    EM53
  507  001670  021001                    DH53            ;"TEST# ERR PC  LKS   CNT1   CNT2"
  508  001672  021422                    DT53            ;$TESTN,$ERRPC,LKS,FIRST,SECND
  509  001674  000000                    0
  510
  511  001676  017043                    EM54            ;"XMIT INTERRUPTS WHEN DISABLED"
  512  001700  020575                    DH1             ;"TEST# ERR PC  TCSR"
  513  001702  021340                    DT1             ;$TESTN,$ERRPC,TCSR
  514  001704  000000                    0
  515
  516  001706  017201                    EM55            ;"XMIT DID NOT INTERRUPT"
  517  001710  020575                    DH1             ;"TEST# ERR PC  TCSR"
  518  001712  021340                    DT1             ;$TESTN,$ERRPC,TCSR
  519  001714  000000                    0
  520
  521  001716  017101                    EM56            ;"XMIT INTERRUPT AT PRIORITY 7"
  522  001720  020575                    DH1             ;"TEST# ERR PC  TCSR"
  523  001722  021340                    DT1             ;$TESTN,$ERRPC,TCSR
  524  001724  000000                    0
  525
  526  001726  017137                    EM57            ;"XMIT INTERRUPTS WITH ENABLE CLEAR"
  527  001730  020575                    DH1             ;"TEST# ERR PC  TCSR"
  528  001732  021340                    DT1             ;$TESTN,$ERRPC,TCSR
  529  001734  000000                    0
  530
  531  001736  017201                    EM60            ;"XMIT DID NOT INTERRUPT"
  532  001740  020575                    DH1             ;"TEST# ERR PC  TCSR"
  533  001742  021340                    DT1             ;$TESTN,$ERRPC,TCSR
  534  001744  000000                    0
  535
  536  001746  017230                    EM61            ;"XMIT RE-INTERRUPTED"
  537  001750  020575                    DH1             ;"TEST# ERR PC  TCSR"
  538  001752  021340                    DT1             ;$TESTN,$ERRPC,TCSR
  539  001754  000000                    0
  540
  541  001756  017254                    EM62            ;"LOADING TBUF DID NOT CLEAR INTERRUPT"
  542  001760  020575                    DH1             ;"TEST# ERR PC  TCSR"
  543  001762  021340                    DT1             ;$TESTN,$ERRPC,TCSR
  544  001764  000000                    0
  545
  546  001766  017321                    EM63            ;"RCVR ACTIVE NOT SET"
  547  001770  020647                    DH6             ;"TEST# ERR PC  RCSR"
  548  001772  021360                    DT6             ;$TESTN,$ERRPC,RCSR
  549  001774  000000                    0
  550
  551  001776  017345                    EM64            ;"RECEIVER DONE NEVER SET"
  552  002000  020647                    DH6             ;"TEST# ERR PC  RCSR"
  553  002002  021360                    DT6             ;$TESTN,$ERRPC,RCSR
  554  002004  000000                    0
  555
  556  002006  017371                    EM65            ;"RCVR ACTIVE NOT CLEARED WITH DONE SET2
  557  002010  020647                    DH6             ;"TEST# ERR PC  RCSR"
  558  002012  021360                    DT6             ;$TESTN,$ERRPC,RCSR
```

```
   559  002014  000000                    0
   560
   561  002016  017437                    EM66            ;"RESET DID NOT CLEAR RCVR DONE"
   562  002020  020647                    DH6             ;"TEST# ERR PC  RCSR"
   563  002022  021360                    DT6             ;$TESTN,$ERRPC,RCSR
   564  002024  000000                    0
   565
   566  002026  017475                    EM67            ;"RDR ENABLE SET DID NOT CLEAR RCVR DONE"
   567  002030  020647                    DH6             ;"TEST# ERR PC  RCSR"
   568  002032  021360                    DT6             ;$TESTN,$ERRPC,RCSR
   569  002034  000000                    0
   570
   571  002036  017540                    EM70            ;"READING RBUF DID NOT CLEAR RCVR DONE"
   572  002040  020647                    DH6             ;"TEST# ERR PC  RCSR"
   573  002042  021360                    DT6             ;$TESTN,$ERRPC,RCSR
   574  002044  000000                    0
   575
   576  002046  017605                    EM71            ;"RCVR INTERRUPT, WITH ENABLE CLEAR"
   577  002050  020647                    DH6             ;"TEST# ERR PC  RCSR"
   578  002052  021360                    DT6             ;$TESTN,$ERRPC,RCSR
   579  002054  000000                    0
   580
   581  002056  017754                    EM72            ;"RCVR DID NOT INTERRUPT"
   582  002060  020647                    DH6             ;"TEST# ERR PC  RCSR"
   583  002062  021360                    DT6             ;$TESTN,$ERRPC,RCSR
   584  002064  000000                    0
   585
   586  002066  017647                    EM73            ;"RCVR INTERRUPTS AT PRIORITY 7"
   587  002070  020647                    DH6             ;"TEST# ERR PC  RCSR"
   588  002072  021360                    DT6             ;$TESTN,$ERRPC,RCSR
   589  002074  000000                    0
   590
   591  002076  017705                    EM74            ;"RCVR INTERRUPT REQUEST PASSED WITH ENABLE CLEAR"
   592  002100  020647                    DH6             ;"TEST# ERR PC  RCSR"
   593  002102  021360                    DT6             ;$TESTN,$ERRPC.RCSR
   594  002104  000000                    0
   595
   596  002106  017754                    EM75            ;"RCVR DID NOT INTERRUPT"
   597  002110  020647                    DH6             ;"TEST# ERR PC  RCSR"
   598  002112  021360                    DT6             ;$TESTN,$ERRPC,RCSR
   599  002114  000000                    0
   600
   601  002116  020003                    EM76            ;"RECEIVER RE-INTERRUPTED"
   602  002120  020647                    DH6             ;"TEST# ERR PC  RCSR"
   603  002122  021360                    DT6             ;$TESTN,$ERRPC,RCSR
   604  002124  000000                    0
   605
   606  002126  020027                    EM77            ;"READING RBUF DID NOT CLEAR INTERRUPT"
   607  002130  020647                    DH6             ;"TEST# ERR PC  RCSR"
   608  002132  021360                    DT6             ;$TESTN,$ERRPC,RCSR
   609  002134  000000                    0
   610
   611  002136  020074                    EM100           ;"RESET DID NOT CLEAR RCVR INTERRUPT"
   612  002140  020647                    DH6             ;"TEST# ERR PC  RCSR"
   613  002142  021360                    DT6             ;$TESTN,$ERRPC,RCSR
   614  002144  000000                    0
```

```
615
616
617  002146  020137                  EM101           ;"'OR' FLAG DID NOT SET"
618  002150  020647                  DH6
619  002152  021360                  DT6             ;$TESTN,$ERRPC,RCSR
620  002154  000000                  0
621
622  002156  020165                  EM102           ;"'ERROR' NOT SET WITH 'OR' FLAG"
623  002160  020647                  DH6             ;"TEST# ERR PC  RCSR"
624  002162  021360                  DT6             ;$TESTN,$ERRPC,RCSR
625  002164  000000                  0
626
627  002166  020224                  EM103           ;"BREAK DID NOT TRANSMIT ALL ZEROES"
628  002170  021046                  DH103           ;"TEST# ERR PC  RCSR  DATA"
629  002172  021436                  DT103           ;$TESTN,$ERRPC,RCSR,$BDDAT
630  002174  000000                  0
631
632  002176  020262                  EM104           ;"BREAK DID NOT SET FRAMING ERROR"
633  002200  020647                  DH6             ;"TEST# ERR PC  RCSR"
634  002202  021360                  DT6             ;$TESTN,$ERRPC,RCSR
635  002204  000000                  0
636
637  002206  020317                  EM105           ;"DATA COMPARE ERROR"
638  002210  021103                  DH105           ;"TEST# ERR PC  RCSR  GOOD  BAD"
639  002212  021450                  DT105           ;$TESTN,$ERRPC,RCSR,GD,BD
640  002214  000000                  0
641
642  002216  020342                  EM106           ;"DATA COMPARE ERROR WITH CABLE"
643  002220  021103                  DH105           ;"TEST# ERR PC  RCSR    GOOD    BAD"
644  002222  021450                  DT105           ;$TESTN,$ERRPC,RCSR,GD,BD
645  002224  000000                  0
646
647  002226  020400                  EM107           ;"TIMEOUT IN EXERCISER TEST"
648  002230  020647                  DH6             ;"TEST# ERR PC  RCSR"
649  002232  021360                  DT6             ;$TESTN,$ERRPC,RCSR
650  002234  000000                  0
651
652  002236  020432                  EM110           ;"INCORRECT RECEIVE COUNT
653  002240  021147                  DH110           ;"TEST# ERR PC  RCSR    TRANS   RCV"
654  002242  021464                  DT110           ;$TESTN,$ERRPC,RCSR,XMTCNT,RCVCNT
655  002244  000000                  0
656
657  002246  020462                  EM111           ;"DATA COMPARE ERROR IN EXERCISER"
660  002250  021103                  DH105           ;"TEST# ERR PC  RCSR    GOOD    BAD"
661  002252  021450                  DT105           ;$TESTN,$ERRPC,RCSR,GD,BD
660  002254  000000                  0
661
662  002256  020522                  EM112           ;"TRAP CATCHER"
663  002260  021213                  DH112           ;"TEST#   ERR PC  RCSR   OLDPC  TRAP ADR"
664  002262  021500                  DT112           ;$TESTN,$ERRPC,RCSR,OLDPC,BOVECT
665  002264  000000                  0
666
667  002266  020537                  EM113           ;"NO CLK INTERRUPT IN EXERCISER"
668  002270  020721                  DH10            ;"TEST#  ERR PC  LKS"
669  002272  021400                  DT10            ;$TESTN,$ERRPC,LKS
670  002274  000000                  0
```

```
                                ;CTSTFL: CONSLE UNDER TEST FLAG
                                ;TMP1:   TEMP LOCATION FOR TABLE OFFSETS
                                ;TMP2:   TEMP LOCATION FOR DEVICE COUNT
                                ;TMP3:   LOCATION FOR DEVICE MAP BIT TEST MASK
                                ;REGISTER AND VECTOR ADDRESSES FOR THE DL-11W UNDER TEST

                                ;CONSOLE REGISTER AND VECTOR ADDRESSES FOR THE DL-11W

                                ;REAL TIME CLOCK REGISTER AND VECTOR ADDRESSES

                                ;SUBROUTINE TO GENERATE DEVICE ADDRESS TABLE

002454  012702  002354  DEVADR: MOV     #ADRTBL,R2      ;POINT R2 TO THE DEVICE ADDRESS TABLE
002460  016700  176456          MOV     $BASE,R0        ;LOAD BASE DEVICE ADDRESS IN R0
002464  010001                  MOV     R0,R1           ;
002466  062701  000170          ADD     #170,R1         ;POINT R1 TO LAST DEVICE ADDRESS
002472  010022          1$:     MOV     R0,(R2)+        ;MOVE DEVICE ADDRESS TO TABLE
002474  062700  000010          ADD     #10,R0          ;POINT R0 TO NEXT DEVICE ADDRESS
002500  020001                  CMP     R0,R1           ;FINISHED GENERATING TABLE?
002502  003773                  BLE     1$              ;BR, IF LAST DEVICE ADDRESS NOT LOADED
002504  000207                  RTS     PC

002506  005067  176356  START:  CLR     $FATAL          ;CLEAR ERROR NO.
002512  005067  176350          CLR     $MSGTYP         ;CLEAR MESSAGE TYPE
002516  005067  176350          CLR     $TESTN          ;CLEAR TEST NO.
002522  005067  177550          CLR     CTSTFL          ;CLEAR CONSOLE UNDER TEST FLAG
002526  005067  176344          CLR     $DEVCT          ;CLEAR DEVICE COUNT
002532  005067  176342          CLR     $UNIT           ;CLEAR UNIT NUMBER
```

```
   727  002536  012737  000006  000004          MOV     #6,@#4          ;INITIALIZE TIMEOUT VECTORS TO TRAP
   728  002544  012737  000003  000006          MOV     #3,@#6          ;  CATCHER ROUTINE
   729
   730                                  .SBTTL  INITIALIZE THE COMMON TAGS
   731                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
   732  002552  012706  001000                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
   733  002556  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
   734  002560  022706  001040                  CMP     #SWR,R6 ;;DONE?
   735  002564  001374                          BNE     .-6             ;;LOOP BACK IF NO
   736  002566  012706  001000                  MOV     #1000,SP                ;;SETUP THE STACK POINTER
   737                                  ;;INITIALIZE A FEW VECTORS
   738  002572  012737  012326  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
   739  002600  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
   740  002606  012737  011632  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
   741  002614  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
   742  002622  012737  014232  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
   743  002630  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
   744  002636  012737  012150  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
   745  002644  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
   746  002652  016767  006602  006572          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
   747  002660  005067  176174                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
   748  002664  112767  000001  176123          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
   749  002672  012767  002672  176106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
   750  002700  012767  002700  176102          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
   751                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
   752                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
   753  002706  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
   754  002712  012737  002746  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
   755  002720  012767  177570  176112          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
   756  002726  012767  177570  176106          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
   757  002734  022777  177777  176076          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
   758  002742  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
   759                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
   760  002744  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
   761  002746  012716  002754          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
   762  002752  000002                          RTI
   763  002754  012767  000176  176056  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
   764  002762  012767  000174  176052          MOV     #DISPREG,DISPLAY
   765  002770  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
   766
   767  002774  005067  176074                  CLR     $PASS           ;;CLEAR PASS COUNT
   768  003000  132767  000200  176101          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
   769  003006  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
   770  003010  012767  001110  176022          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
   771  003016                          67$:
   772  003016  132767  000001  176062          BITB    #BIT0,$ENV      ;CHECK IF ON APT
   773  003024  001404                          BEQ     MANL            ;BR IF NOT APT
   774  003026  132767  000200  176053          BITB    #BIT7,$ENVM     ;DID APT SIZE
   775  003034  001056                          BNE     APTSZD          ;BR, IF APT SIZED
   776  003036  032777  000040  175774  MANL:   BIT     #BIT5,@SWR      ;WAS "$DEVM" MANUALLY SET?
   777  003044  001052                          BNE     APTSZD          ;IF YES, SKIP SELF-SIZING
   778
   779  003046  004767  177402          SIZE:   JSR     PC,DEVADR       ;GENERATE DEVICE ADDRESS TABLE
   780  003052  005067  177224                  CLR     TMP2            ;CLR TEMP LOCATION TO KEEP DEVICE COUNT
   781  003056  005067  176062                  CLR     $DEVM           ;CLEAR DEVICE MAP
   782  003062  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
```

```
 783  003066  012737  003116  000004          MOV     #4$,@#4         ;SET TIMEOUT POINTER
 784  003074  016700  176042                  MOV     $BASE,R0        ;LOAD BASE ADDRESS
 785  003100  062700  000160                  ADD     #160,R0         ;POINT R0 TO UNIT #15 (UNIT#0=CONSOLE)
 786  003104  005710                  3$:     TST     (R0)            ;CHECK FOR DEVICE EXISTANCE
 787  003106  005267  176032                  INC     $DEVM           ;INDICATE DEVICE EXISTANCE IN DEVICE MAP
 788  003112  005267  177164                  INC     TMP2            ;INCREMENT DEVICE COUNT
 789  003116  012706  001000          4$:     MOV     #1000,SP                ;RESET STACK POINTER
 790  003122  006367  176016                  ASL     $DEVM           ;ADJUST DEVICE MAP FOR NEXT UNIT CHECK
 791  003126  162700  000010                  SUB     #10,R0          ;POINT R0 TO NEXT DEVICE NUMBER
 792  003132  026700  176004                  CMP     $BASE,R0        ;FINISHED SIZING?
 793  003136  003762                          BLE     3$              ;BR, IF BASE ADDRESS HAS NOT BEEN CHECKED
 794  003140  016700  177162                  MOV     CRCSR,R0        ;LOAD CONSOLE DEVICE ADDRESS
 795  003144  012737  003164  000004          MOV     #5$,@#4         ;SET UP TIMEOUT POINTER
 796  003152  005710                          TST     (R0)            ;TEST FOR CONSOLE EXISTANCE
 797  003154  005267  175764                  INC     $DEVM           ;INDICATE CONSOLE EXISTANCE IN DEVICE MAP
 798  003160  005267  177116                  INC     TMP2            ;INCREMENT DEVICE COUNT
 799  003164  010337  000004          5$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
 800  003170  000415                          BR      VCTADR          ;BR TO GENERATE VECTOR ADDRESS TABLE
 801
 802  003172  005067  177104          APTSZD: CLR     TMP2            ;CLEAR TEMP LOCATION TO KEEP DEVICE CNT
 803  003176  016702  175742                  MOV     $DEVM,R2        ;MOVE DEVICE MAP TO R2
 804  003202  005702                  TSTDVM: TST     R2              ;TEST MSB OF DEVICE MAP
 805  003204  100002                          BPL     1$              ;BR, IF MSB IS ZERO
 806  003206  005267  177070                  INC     TMP2            ;INCREMENT DEVICE COUNT, IF MSB=1
 807  003212  006302                  1$:     ASL     R2              ;SHIFT NEXT BIT INTO MSB POSITION
 808  003214  001401                          BEQ     DVADT           ;BR, IF NO OTHER BITS ARE SET IN $DEVM
 809  003216  000771                          BR      TSTDVM          ;CONTINUE CHECKING $DEVM, IF MORE BITS SET
 810  003220  004767  177230          DVADT:  JSR     PC,DEVADR       ;GENERATE DEVICE ADDRESS TABLE
 811
 812                                  ;GENERATE VECTOR ADDRESS TABLE
 813
 814  003224  012702  002414          VCTADR: MOV     #VCTTBL,R2      ;GET LOCATION OF VECTOR TABLE
 815  003230  113700  001136                  MOVB    @#$VECT1,R0     ;COPY BASE VECTOR
 816  003234  042700  177400                  BIC     #177400,R0      ;CLEAR BYTE SIGN EXTENSION
 817  003240  010001                          MOV     R0,R1           ;
 818  003242  062701  000170                  ADD     #170,R1         ;POINT R1 TO LAST DEVICE VECTOR
 819  003246  010022          1$:     MOV     R0,(R2)+        ;PUT VECTOR ADDRESS IN TABLE
 820  003250  062700  000010                  ADD     #10,R0          ;POINT R0 TO NEXT VECTOR ADDRESS
 821  003254  020001                          CMP     R0,R1           ;FINISHED GENERATING VECTOR TABLE?
 822  003256  003773                          BLE     1$              ;BR, IF LAST VECTOR IS NOT LOADED
 823
 824                                  ;MOVE DEVICE COUNT INTO DEVICE COUNT MESSAGE
 825
 826  003260  016700  177016                  MOV     TMP2,R0         ;COPY DEVICE COUNT INTO R0
 827  003264  005001                          CLR     R1              ;CLEAR AUXILARY REGISTER
 828  003266  000300                          SWAB    R0              ;PUT DEVICE COUNT IN UPPER BYTE OF R0
 829  003270  006300                          ASL     R0              ;MOVE MSB OF COUNT INTO
 830  003272  006300                          ASL     R0              ;MSB OF R0
 831  003274  006300          SHIFT:  ASL     R0              ;PUT MSB OF COUNT INTO CARRY
 832  003276  106101                          ROLB    R1              ;MOVE MSB OF COUNT INTO R1
 833  003300  006300                          ASL     R0              ;MOVE NEXT BIT TO CARRY
 834  003302  106101                          ROLB    R1              ;MOVE INTO R1
 835  003304  006300                          ASL     R0              ;MOVE LAST BIT OF DIGIT
 836  003306  106101                          ROLB    R1              ;INTO R1
 837  003310  062701  000060                  ADD     #60,R1          ;CONVERT DIGIT TO ASCII
 838  003314  000301                          SWAB    R1              ;MOVE DIGIT TO UPPER BYTE
```

```
                                                ;HAVE BOTH DIGITS BEEN MOVED TO R1?
003316  032701  000020          BIT     #BIT4,R1
003322  001764                  BEQ     SHIFT           ;BR, IF NOT
003324  010167  015760          MOV     R1,M2A          ;MOVE DEVICE COUNT TO OUTPUT MESSAGE


003330  052767  000002  176746  BEGIN:  BIS     #BIT1,TMP3      ;SET UP BIT MASK TO TEST $DEVM FOR DEVICES EXCEPT CONSOL
003336  005067  176736          CLR     TMP1            ;CLEAR LOCATION TO STORE TABLE OFFSETS
003342  032767  000001  175574  BIT     #BIT0,$DEVM     ;IS CONSOLE TO BE TESTED?
003350  001001                  BNE     TCONS           ;BR, IF CONSOLE IS TO BE TESTED
003352  000414                  BR      TSTDEV          ;BR, TO TEST OTHER DEVICES
003354  005267  176716  TCONS:  INC     CTSTFL          ;INDICATE CONSOLE UNDER TEST
003360  012700  002326          MOV     #CRCSR,R0       ;SET UP CONSOLE DEVICE ADDRESSES
003364  012701  002306          MOV     #RCSR,R1        ;POINT R1 TO UUT ADDRESS TABLE
003370  012021          1$:     MOV     (R0)+,(R1)+     ;TRANSFER CONSOLE ADDRESSES
003372  022701  002324          CMP     #TPSW,R1        ;FINISHED TRANSFER?
003376  002374                  BGE     1$              ;BR, IF NOT
003400  000167  000122          JMP     TST1            ;GO TEST CONSOLE INTERFACE


                        ;PREPARE ADDRESSES AND VECTORS FOR UUT
003404  036767  176674  175532  TSTDEV: BIT     TMP3,$DEVM      ;CHECK TO SEE IF DEVICE IS TO BE TESTED
003412  001010                  BNE     SETADR          ;BR, IF YES
003414  006367  176664          ASL     TMP3            ;SHIFT MASK TO CHECK NEXT $DEVM BIT
003420  062767  000002  176652  ADD     #2,TMP1         ;INCREMENT TABLE INDEX
003426  005267  175446          INC     $UNIT           ;INCREMENT UNIT NUMBER
003432  000764                  BR      TSTDEV          ;GO TEST NEXT BIT OF DEVICE MAP


003434  005267  175440  SETADR: INC     $UNIT           ;UPDATE UNIT NUMBER
003440  006367  176640          ASL     TMP3            ;UPDATE DEVICE MAP TEST MASK
003444  016702  176636          MOV     TMP1,R2         ;MOVE TABLE OFFSET TO R2
003450  062767  000002  176622  ADD     #2,TMP1         ;UPDATE TABLE OFFSET FOR NEXT DEVICE
003456  016200  002354          MOV     ADRTBL(R2),R0   ;PUT UUT ADDRESS INTO R0
003462  012701  002306          MOV     #RCSR,R1        ;POINT R1 TO STORAGE AREA FOR UUT ADDRESSES
003466  010021          ADR:    MOV     R0,(R1)+        ;TRANSFER UUT ADDRESS
003470  062700  000002          ADD     #2,R0           ;POINT TO NEXT UUT REGISTER
003474  030027  000006          BIT     R0,#6           ;FINISHED TRANSFER?
003500  001372                  BNE     ADR             ;BR, IF NOT


003502  016200  002414          MOV     VCTTBL(R2),R0   ;PUT UUT VECTOR INTO R0
003506  010021          VECT:   MOV     R0,(R1)+        ;TRANSFER UUT VECTORS TO ACTIVE TABLE AREA
003510  062700  000002          ADD     #2,R0           ;POINT TO NEXT VECTOR
003514  030027  000006          BIT     R0,#6           ;FINISHED TRANSFER?
003520  001372                  BNE     VECT            ;BR, IF NOT
003522  000167  000000          JMP     TST1            ;GO TEST DEVICE
```

```
                                        ;;*****************************************************************
                                        ;;TEST 1        TEST ABILITY TO REFERENCE TCSR
                                        ;;*****************************************************************
003526  000004                TST1:   SCOPE
003530  013703  000004                MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
003534  012737  003550  000004        MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
003542  005777  176544                TST     @TCSR           ;REFERENCE THE XMIT COMMAND/STATUS REG.
003546  000412                      BR      4$              ; GO TO END OF TEST

003550  022626                1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
003552  005767  176520              TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
003556  001002                      BNE     2$              ;IF YES, SKIP ERROR TYPEOUT
003560  104001                      ERROR   1               ;REPORT ERROR TO APT & TTY
003562  000404                      BR      4$              ;BR TO END OF TEST
003564                          2$:
003564  004767  006726              JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
003570  000001                      1                               ;;THE ERROR NUMBER (FROM APT LIST)
003572  000000                3$:   HALT
003574  010337  000004        4$:   MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR


                                        ;;*****************************************************************
                                        ;;TEST 2        TEST ABILITY TO REFERENCE TBUF
                                        ;;*****************************************************************
003600  000004                TST2:   SCOPE
003602  013703  000004                MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
003606  012737  003622  000004        MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
003614  005777  176474                TST     @TBUF           ;REFERENCE THE XMIT BUFFER
003620  000412                      BR      4$              ;GO TO END OF  TEST

003622  022626                1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
003624  005767  176446              TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
003630  001002                      BNE     2$              ;IF YES, SKIP ERROR TYPEOUT
003632  104002                      ERROR   2               ;REPORT ERROR TO APT & TTY
003634  000404                      BR      4$              ;BR TO END OF TEST
003636                          2$:
003636  004767  006654              JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
003642  000002                      2                               ;;THE ERROR NUMBER (FROM APT LIST)
003644  000000                3$:   HALT
003646  010337  000004        4$:   MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
```

D03


```
                                ;;*****************************************************************
                                ;*TEST 3        TEST THAT TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
                                ;;*****************************************************************
003652  000004                  TST3:   SCOPE
003654  005077  176434                  CLR     @TBUF           ;LOAD XBUF
003660  105777  176426                  TSTB    @TCSR           ;CHECK DONE
003664  100016                          BPL     3$              ;BR IF CLEAR
                                                                ;FILL SECOND BUFFER, BECAUSE REFRESH COULD CAUSE
                                                                ;   FIRST TEST TO FAIL
003666  005077  176422                  CLR     @TBUF           ;FILL DOUBLE BUFFER
003672  105777  176414                  TSTB    @TCSR           ;CHECK DONE
003676  100011                          BPL     2$              ;BR IF CLEAR

003700  005767  176372                  TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
003704  001002                          BNE     1$              ;IF YES, SKIP ERROR TYPEOUT
003706  104003                          ERROR   3               ;DONE NOT CLEARED WITH TBUF FULL
003710  000404                          BR      3$              ;BR TO END OF TEST
003712                          1$:
003712  004767  006600                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
003716  000003                          3                               ;;THE ERROR NUMBER (FROM APT LIST)
003720  000000                  2$:     HALT                    ;TCSR "DONE" NOT CLEARED WITH TBUF FULL
003722  005000                  3$:     CLR     R0              ;CLEAR TIMER
003724  105777  176362          4$:     TSTB    @TCSR           ;CHECK FOR XMIT DONE
003730  100414                          BMI     6$              ;IF DONE SETS, BR TO NEXT TEST
003732  005200                          INC     R0              ;INCREMENT TIMER
003734  001373                          BNE     4$              ;BR IF TIMER NOT DONE

003736  005767  176334                  TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
003742  001002                          BNE     5$
003744  104004                          ERROR   4               ;TCSR "DONE" DOES NOT SET
003746  000405                          BR      6$              ;BR TO END OF TEST
003750                          5$:
003750  004767  006542                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
003754  000004                          4                               ;;THE ERROR NUMBER (FROM APT LIST)
003756  000000                          HALT
003760  000415                          BR      TST4            ;BR TO NEXT TEST, AND SKIP THE TYPEOUT THAT FOLLOWS
                                                                ;  BECAUSE OF THIS FAILURE

003762  005737  000042          6$:     TST     @#42            ;UNDER ACT11?
003766  001012                          BNE     TST4            ;IF YES, SKIP IDENT. TYPEOUT
003770  005767  175100                  TST     $PASS           ;IS THIS THE FIRST PASS?
003774  001007                          BNE     TST4            ;IF NOT BR TO NEXT TEST & SKIP THE IDENTIFICATION TYPEOU
003776  005767  175074                  TST     $DEVCT          ;IS THIS THE FIRST SUBPASS?
004002  001004                          BNE     TST4            ;IF NOT, BRTO NEXT TEST
004004  104401                          TYPE                    ;TYPE PROGRAM INDENTIFICATION
004006  021264                          M1
004010  104401                          TYPE                    ;TYPE NUMBER OF DEVICES UNDER TEST
004012  021306                          M2
```

E03


```
                                ;;*****************************************************************
                                ;*TEST 4         TEST THAT TCSR "DONE" SETS WITH RESET
                                ;;*****************************************************************
004014  000004                  TST4:   SCOPE
004016  005077  176272                  CLR     @TBUF           ;LOAD TRANSMIT BUFFER
004022  105777  176264          1$:     TSTB    @TCSR           ;WAIT FOR DONE
004026  100375                          BPL     1$
004030  005077  176260                  CLR     @TBUF           ;LOAD SECOND BUFFER
004034  000240                          NOP
004036  000005                          RESET                   ;CLEAR DONE WITH RESET
004040  105777  176246                  TSTB    @TCSR           ;CHECK FOR DONE SET
004044  100401                          BMI     TST5            ;BR TO NEXT TEST IF DONE SET

004046  104005                          ERROR   5               ;TCSR "DONE" DOES NOT SET WITH RESET



                                ;;*****************************************************************
                                ;*TEST 5         TEST ABILITY TO ACCESS RCSR
                                ;;*****************************************************************
004050  000004                  TST5:   SCOPE
004052  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
004056  012737  004072  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
004064  005777  176216                  TST     @RCSR           ;ACCESS RCSR
004070  000402                          BR      2$              ;BR TO END OF TEST

004072  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
004074  104006                          ERROR   6               ;CAN NOT ACCESS RCSR
004076  010337  000004          2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
```

```
                                        ;;*****************************************************************
                                        ;*TEST 6        TEST ABILITY TO ACCESS RBUF
                                        ;;*****************************************************************
004102  000004                TST6:   SCOPE
004104  013703  000004                MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
004110  012737  004124  000004        MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
004116  005777  176166                TST     @RBUF           ;ACCESS RBUF
004122  000402                        BR      2$              ;BR TO END OF TEST

004124  022626                1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
004126  104007                        ERROR   7               ;CAN NOT ACCESS RBUF
004130  010337  000004        2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR


                                        ;;*****************************************************************
                                        ;*TEST 7        TEST ABILITY TO ACCESS LKS
                                        ;;*****************************************************************
004134  000004                TST7:   SCOPE
004136  005767  176134                TST     CTSTFL          ;IS CONSOLE UNDER TEST?
004142  001420                        BEQ     TST10           ;IF NOT, SKIP THIS TEST
004144  032777  000100  174666        BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
004152  001014                        BNE     TST10           ;IF YES, SKIP THIS TEST
004154  013703  000004                MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
004160  012737  004174  000004        MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
004166  005777  176154                TST     @LKS            ;ACCESS LKS
004172  000402                        BR      2$              ;NO TIMEOUT -  BR TO END OF TEST

004174  022626                1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
004176  104010                        ERROR   10              ;CAN NOT ACCESS LKS

004200  010337  000004        2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
```

```
                                        ;;*********************************************************************
                                        ;*TEST 10       TEST THAT BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET
                                        ;;*********************************************************************
004204  000004                  TST10:  SCOPE
004206  032777  000400  174624            BIT     #BIT8,@SWR      ;IS BREAK FUNCTION ENABLED?
004214  001462                          BEQ     TST11           ;BR TO NEXT TEST, IF NOT
004216  000005                          RESET                   ;CLEAR EVERYTHING
004220  032777  000001  176064          BIT     #BIT0,@TCSR     ;CHECK BIT0 OF TCSR CLEAR
004226  001411                          BEQ     3$              ;BR IF CLEAR
004230  005767  176042                  TST     CTSTFL
004234  001002                          BNE     1$
004236  104011                          ERROR   11              ;BIT0 WAS NOT CLEAR AFTER RESET
004240  000404                          BR      3$
004242                          1$:
004242  004767  006250                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
004246  000011                          11                              ;;THE ERROR NUMBER (FROM APT LIST)
004250  000000                  2$:     HALT
004252  052777  000001  176032  3$:     BIS     #BIT0,@TCSR     ;SET BIT0 IN TCSR
004260  032777  000001  176024          BIT     #BIT0,@TCSR     ;TEST BIT0 OF TCSR
004266  001001                          BNE     4$              ;BR IF SET
004270  104012                          ERROR   12              ;BIT0 OF TCSR WILL NOT SET
004272  042777  000001  176012  4$:     BIC     #BIT0,@TCSR     ;CLEAR BIT0 OF TCSR
004300  032777  000001  176004          BIT     #BIT0,@TCSR     ;TEST BIT0 OF TCSR
004306  001411                          BEQ     7$
004310  005767  175762                  TST     CTSTFL
004314  001002                          BNE     5$
004316  104013                          ERROR   13              ;BIT0 OF TCSR WILL NOT CLEAR
004320  000404                          BR      7$
004322                          5$:
004322  004767  006170                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
004326  000013                          13                              ;;THE ERROR NUMBER (FROM APT LIST)
004330  000000                  6$:     HALT
004332  052777  000001  175752  7$:     BIS     #BIT0,@TCSR     ;SET BIT0 IN TCSR
004340  000005                          RESET                   ;CLEAR BIT0 WITH RESET
004342  032777  000001  175742          BIT     #BIT0,@TCSR     ;TEST BIT0 CLEAR
004350  001404                          BEQ     TST11           ;BR IF CLEAR
004352  042777  000001  175732          BIC     #BIT0,@TCSR     ;CLEAR BIT0, TO PRINT ERROR
004360  104014                          ERROR   14              ;RESET DID NOT CLEAR BIT0 OF TCSR
```

```
                                ;;*****************************************************************
                                ;*TEST 11       TEST THAT BIT2(MAINT. BIT) CAN BE SET & RESET
                                ;;*****************************************************************
004362  000004                  TST11:  SCOPE
004364  000005                          RESET                   ;CLEAR EVERYTHING
004366  032777  000004  175716          BIT     #BIT2,@TCSR     ;TEST FOR BIT2 OF TCSR CLEAR
004374  001411                          BEQ     3$              ;BR IF CLEAR

004376  005767  175674                  TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
004402  001002                          BNE     1$              ;IF YES, SKIP ERROR TYPEOUT
004404  104015                          ERROR   15              ;BIT2 OF TCSR NOT CLEAR AFTER RESET
004406  000404                          BR      3$

004410                          1$:
004410  004767  006102                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
004414  000015                          15                              ;;THE ERROR NUMBER (FROM APT LIST)
004416  000000                  2$:     HALT

004420  052777  000004  175664  3$:     BIS     #BIT2,@TCSR     ;SET BIT2 OF TCSR
004426  032777  000004  175656          BIT     #BIT2,@TCSR     ;TEST FOR BIT2 SET
004434  001001                          BNE     4$              ;BR IF SET

004436  104016                          ERROR   16              ;BIT2 OF TCSR WILL NOT SET

004440  042777  000004  175644  4$:     BIC     #BIT2,@TCSR     ;CLEAR BIT2 OF TCSR
004446  032777  000004  175636          BIT     #BIT2,@TCSR     ;TEST BIT2 CLEAR
004454  001411                          BEQ     7$              ;BR IF CLEAR

004456  005767  175614                  TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
004462  001002                          BNE     5$              ;IF YES, SKIP ERROR TYPEOUT
004464  104017                          ERROR   17
004466  000404                          BR      7$
004470                          5$:
004470  004767  006022                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
004474  000017                          17                              ;;THE ERROR NUMBER (FROM APT LIST)
004476  000000                  6$:     HALT                    ;BIT0 OF TCSR WILL NOT CLEAR

004500  052777  000004  175604  7$:     BIS     #BIT2,@TCSR     ;SET BIT2 OF TCSR
004506  000005                          RESET                   ;CLEAR BIT2 WITH RESET
004510  032777  000004  175574          BIT     #BIT2,@TCSR     ;TEST FOR BIT2 CLEAR
004516  001404                          BEQ     TST12           ;IF CLEAR, GO TO NEXT TEST

004520  042777  000004  175564          BIC     #BIT2,@TCSR     ;CLEAR BIT2, TO PRINT ERROR
004526  104020                          ERROR   20
                                                                ;RESET DID NOT CLEAR BIT2 OF TCSR
```

```
                                ;;*****************************************************************
                                ;*TEST 12       TEST THAT BIT6(XMIT INT EN) CAN BE SET & RESET
                                ;;*****************************************************************
004530  000004                  TST12:  SCOPE
004532  000005                          RESET                           ;CLEAR EVERYTHING
004534  017703  175562                  MOV     @TVECT,R3               ;SAVE XMIT VECTOR
004540  012777  004570  175554          MOV     #1$,@TVECT              ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
004546  004767  005016                  JSR     PC,4$RPSW               ;SET PSW TO PRIORITY=7
004552  000340                          .WORD   340
004554  032777  000100  175530          BIT     #BIT6,@TCSR             ;TEST BIT6 OF TCSR
004562  001404                          BEQ     2$                      ;BR IF ZERO
004564  104021                          ERROR   21
                                                                        ;BIT6 IN TCSR NOT CLEAR AFTER RESET
004566  000402                          BR      2$

004570  022626                  1$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
004572  104022                          ERROR   22
                                                                        ;XMIT INTERRUPT OCCURRED  PRIO=7

004574  052777  000100  175510  2$:     BIS     #BIT6,@TCSR             ;SET BIT6 OF TCSR
004602  032777  000100  175502          BIT     #BIT6,@TCSR             ;TEST BIT6 OF TCSR
004610  001001                          BNE     3$                      ;BR, IF SET

004612  104023                          ERROR   23
                                                                        ;CANNOT SET BIT6 OF TCSR

004614  042777  000100  175470  3$:     BIC     #BIT6,@TCSR             ;CLEAR BIT6 OF TCSR
004622  032777  000100  175462          BIT     #BIT6,@TCSR             ;TEST BIT6 OF TCSR
004630  001401                          BEQ     4$                      ;BR IF CLEAR

004632  104024                          ERROR   24
                                                                        ;CANNOT CLEAR BIT6 OF TCSR

004634  052777  000100  175450  4$:     BIS     #BIT6,@TCSR             ;SET BIT6 OF TCSR
004642  000005                          RESET                           ;CLEAR BIT6 WITH RESET
004644  032777  000100  175440          BIT     #BIT6,@TCSR             ;TEST BIT6 OF TCSR
004652  001401                          BEQ     5$                      ;BR IF CLEAR

004654  104025                          ERROR   25
                                                                        ;CANNOT CLEAR BIT6 OF TCSR WITH RESET
004656  010377  175440          5$:     MOV     R3,@TVECT               ;RESTORE XMIT VECTOR
```

```
                                ;;*****************************************************************
                                ;*TEST 13       TEST THAT BIT6 OF RCSR CAN BE SET & RESET
                                ;;*****************************************************************
004662  000004                  TST13:  SCOPE
004664  000005                          RESET                   ;CLEAR EVERYTHING
004666  017703  175424                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
004672  012777  004722  175416          MOV     #1$,@RVECT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
004700  004767  004664                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY=7
004704  000340                          .WORD   340
004706  032777  000100  175372          BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
004714  001404                          BEQ     2$
004716  104026                          ERROR   26
                                                                ;BIT6 OF RCSR NOT CLEAR AFTER RESET
004720  000402                          BR      2$

004722  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
004724  104027                          ERROR   27
                                                                ;RCVR INTERRUPT WITH PRIORITY=7

004726  052777  000100  175352  2$:     BIS     #BIT6,@RCSR     ;SET BIT6 OF RCSR
004734  032777  000100  175344          BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
004742  001001                          BNE     3$              ;BR, IF SET

004744  104030                          ERROR   30
                                                                ;CANNOT SET BIT6 OF RCSR

004746  042777  000100  175332  3$:     BIC     #BIT6,@RCSR     ;CLEAR BIT6 OF RCSR
004754  032777  000100  175324          BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
004762  001401                          BEQ     4$              ;BR, IF CLEAR

004764  104031                          ERROR   31
                                                                ;CANNOT CLEAR BIT6 OF RCSR

004766  052777  000100  175312  4$:     BIS     #BIT6,@RCSR     ;SET BIT6 OF RCSR
004774  000005                          RESET                   ;CLEAR BIT6OF RCSR WITH RESET
004776  032777  000100  175302          BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
005004  001401                          BEQ     5$              ;BR, IF CLEAR

005006  104032                          ERROR   32
                                                                ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
005010  010377  175302          5$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
```

```
                                ;;*****************************************************************
                                ;*TEST 14       TEST THAT BIT6 OF LKS CAN BE SET & RESET
                                ;;*****************************************************************
005014  000004                  TST14:  SCOPE
005016  005767  175254                  TST     CTSTFL          ;IS CONSOLE UNDER TEST?
005022  001460                  BEQ     TST15           ;IF NOT, SKIP THIS TEST
005024  032777  000100  174006  BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
005032  001054                  BNE     TST15           ;IF YES, SKIP THIS TEST
005034  000005                  RESET
005036  017703  175306          MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
005042  012777  005072  175300  MOV     #1$,@RTCVT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
005050  004767  004514          JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
005054  000340                  .WORD   340
005056  032777  000100  175262  BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
005064  001404                  BEQ     2$
005066  104033                  ERROR   33
                                                        ;BIT6 OF LKS NOT CLEAR AFTER RESET
005070  000402                  BR      2$

005072  022626          1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
005074  104034                  ERROR   34
                                                        ;LKS INTERRUPT WITH PRIORITY=7

005076  052777  000100  175242  2$:     BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
005104  032777  000100  175234          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
005112  001001                  BNE     3$              ;BR IF SET

005114  104035                  ERROR   35
                                                        ;CANNOT SET BIT6 OF LKS

005116  042777  000100  175222  3$:     BIC     #BIT6,@LKS      ;CLEAR BIT6 OF LKS
005124  032777  000100  175214          BIT     #BIT6,@LKS      ;TEST BIT6 OF LK
005132  001401                  BEQ     4$
005134  104036                  ERROR   36
                                                        ;CANNOT CLEAR BIT6 OF LKS
005136  052777  000100  175202  4$:     BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
005144  000005                  RESET                   ;CLEAR BIT6 OF LKS WITH RESET
005146  032777  000100  175172          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
005154  001401                  BEQ     5$              ;BR IF CLEAR

005156  104037                  ERROR   37
                                                        ;CANNOT CLEAR BIT6 OF LKS WITH RESET
005160  010377  175164          5$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
```

```
1264
1265
1266                                    ;;*****************************************************************
1267                                    ;*TEST 15       TEST THAT REGISTERS CAN NOT BE ADDRESSES AS ANYTHING BUT 17XXXX
1268                                    ;;*****************************************************************
1269  005164  000004                    TST15:  SCOPE
1270  005166  013703  000004                    MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1271  005172  013704  000006                    MOV     @#6,R4          ;SAVE TIMEOUT PSW
1272  005176  012737  005250  000004            MOV     #2$,@#4         ;SET UP TIMEOUT VECTOR
1273  005204  012737  000340  000006            MOV     #340,@#6        ;KEEP PRIO=7
1274  005212  000005                            RESET                   ;CLEAR EVERYTHING
1275  005214  012700  000002                    MOV     #BIT1,R0        ;SET UP BIT MASK
1276  005220  016767  175062  173574    1$:     MOV     RCSR,$GDADR     ;MOVE GOOD RCSR ADDRESS INTO TEST ADDRESS
1277  005226  040067  173570                    BIC     R0,$BDADR       ;CREATE BAD ADDRESS BY COMPLEMENTING ONE BIT
1278  005232  026767  175050  173562            CMP     RCSR,$GDADR     ;ARE ADDRESSES IDENTICAL\
1279  005240  001415                            BEQ     3$              ;IF YES,DO NOT TEST THIS ADDRESS
1280  005242  052777  000100  173552            BIS     #BIT6,@$BDADR   ;SET BIT6 USING BAD ADDRESS
1281  005250  032777  000100  175030    2$:     BIT     #BIT6,@RCSR     ;CHECK TO SEE IF GOOD ADDRESS CONTAINS BIT6
1282  005256  001014                            BNE     4$              ;BR IF SET ---> ERROR
1283  005260  022706  001000                    CMP     #1000,SP                ;DID TEST CAUSE A TIMEOUT?
1284  005264  001003                            BNE     3$              ;IF YES, SKIP RESTORE OF MEMORY
1285  005266  042777  000100  173526            BIC     #BIT6,@$BDADR   ;RESTORE ANY MEMORY LOCATION THAT WAS ALTERED
1286  005274  012706  001000            3$:     MOV     #1000,SP                ;RESTORE SP IN CASE A TIMEOUT OCCURRED
1287  005300  006300                            ASL     R0              ;SHIFT BIT MASK TO NEXT POSITION
1288  005302  005700                            TST     R0              ;COMPLEMENTED ALL BITS FROM 1 - 15?
1289  005304  001345                            BNE     1$              ;BR, IF NOT
1290  005306  000403                            BR      5$              ;BR TO NEXT TEST
1291
1292  005310  012706  001000            4$:     MOV     #1000,SP                ;RESET STACK PIONTER
1293  005314  104040                            ERROR   40
1294  005316  010337  000004            5$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1295  005322  010437  000006                    MOV     R4,@#6          ;RESTORE TIMEOUT PSW
```

```
1296
1297
1298                                    ;;*****************************************************************
1299                                    ;*TEST 16       TEST THAT BIT7 OF LKS SETS & CAN BE CLEARED
1300                                    ;;*****************************************************************
1301  005326  000004                    TST16:  SCOPE
1302  005330  005767  174742                    TST     CTSTFL          ;IS CONSOLE UNDER TEST?
1303  005334  001431                    BEQ     TST17           ;IF NOT, SKIP THIS TEST
1304  005336  032777  000100  173474    BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
1305  005344  001025                    BNE     TST17           ;IF YES, SKIP THIS TEST
1306  005346  000005                    RESET                   ;CLEAR EVERYTHING & SET BIT7 OF LKS
1307  005350  000240                    NOP
1308  005352  105777  174770    1$:     TSTB    @LKS            ;TEST FOR BIT7 OF LKS
1309  005356  100401                    BMI     2$              ;BR IF SET
1310  005360  104041                    ERROR   41              ;BIT7 OF LKS DID NOT SET WITH RESET
1311  005362  042777  000200  174756  2$:     BIC     #BIT7,@LKS      ;CLEAR BIT7 OF LKS
1312  005370  032777  000200  174750          BIT     #BIT7,@LKS      ;TEST BIT7 OF LKS
1313  005376  001401                    BEQ     3$
1314
1315  005400  104042                    ERROR   42
1316                                                            ;CAN NOT CLEAR BIT7 OF LKS
1317  005402  005000                    3$:     CLR     R0              ;CLEAR TIMER
1318  005404  105777  174736    CONT:   TSTB    @LKS            ;TEST FOR BIT7 OF LKS
1319  005410  100403                    BMI     TST17           ;BR, IF SET
1320  005412  005200                    INC     R0              ;INCREMENT TIMER
1321  005414  001373                    BNE     CONT            ;CONTINUE UNTIL TIME EXPIRES
1322
1323  005416  104043                    ERROR   43              ;BIT7 OF LKS DOES NOT SET
1324
```

```
                                ;;*****************************************************************
                                ;*TEST 17       TEST THAT THE REAL TIME CLOCK INTERRUPTS PROPERLY
                                ;;*****************************************************************
005420  000004                TST17:  SCOPE
005422  005767  174650                TST     CTSTFL          ;IS CONSOLE UNDER TEST?
005426  001477                        BEQ     TST20           ;IF NOT, SKIP THIS TEST
005430  032777  000100  173402        BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
005436  001073                        BNE     TST20           ;IF YES, SKIP THIS TEST
005440  004767  004124                JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
005444  000340                                .WORD   340
005446  017703  174676                MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
005452  012777  005514  174670        MOV     #2$,@RTCVT      ;SET RTC INTERRUPT VECTOR TO ERROR REPORT
005460  012777  000340  174664        MOV     #340,@RTCPSW    ;KEEP PRIORITY AT 7
005466  052777  000100  174652        BIS     #BIT6,@LKS      ;SET INTERRUPT ENABLE
005474  042777  000200  174644        BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
005502  105777  174640          1$:   TSTB    @LKS            ;WAIT FOR RTC DONE(INTERRUPT REQUEST)
005506  100375                        BPL     1$
005510  000240                        NOP                     ;GIVE TIME FOR ANY INTERRUPTS
005512  000402                        BR      3$              ;BR, IF NO INTERRUPT OCCURS

005514  022626            2$:         CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
005516  104044                        ERROR   44              ;RTC INTERRUPTS AT PRIORITY 7

005520  005077  174622          3$:   CLR     @LKS    ;DISABLE RTC INTERRUPTS & CLEAR DONE
005524  012777  005552  174616        MOV     #4$,@RTCVT      ;SET RTC INTERRUPT VECTOR FOR ERROR
005532  004767  004032                JSR     PC,WRPSW        ;CHANGE PSW TO PRIORITY 5
005536  000240                                .WORD   240
005540  105777  174602          20$:  TSTB    @LKS            ;WAIT FOR DONE (NTERRUPT REQUEST)
005544  100375                        BPL     20$
005546  000240                        NOP                     ;GIVE TIME FOR ANY INTERRUPT
005550  000402                        BR      5$              ;IF NO INTERRUPT - BR TO CONTINUE TEST

005552  022626            4$:         CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
005554  104045                        ERROR   45              ;RTC INTERRUPTS WITH INTERRUPTS DISABLED

005556  012777  005612  174564  5$:   MOV     #7$,@RTCVT      ;POINT RTC VECTOR TO END OF TEST
005564  052777  000100  174554        BIS     #BIT6,@LKS      ;ALLOW INTERRUPTS
005572  042777  000200  174546        BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
005600  105777  174542          6$:   TSTB    @LKS            ;WAIT FOR RTC DONE
005604  100375                        BPL     6$
005606  000240                        NOP                     ;GIVE TIME FOR INTERRUPT

005610  104046                        ERROR   46              ;RTC INTERRUPT DID NOT OCCUR

005612  022626            7$:         CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
005614  042777  000100  174524        BIC     #BIT6,@LKS      ;DISABLE INTERRUPTS
005622  010377  174522                MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
```

```
                                        ;;*****************************************************************
                                        ;*TEST 20        TEST RTC FOR DOUBLE INTERRUPTS
                                        ;;*****************************************************************
005626  000004                  TST20:  SCOPE
005630  005767  174442                  TST     @TSTFL          ;IS CONSOLE UNDER TEST?
005634  001453                          BEQ     TST21           ;IF NOT, SKIP THIS TEST
005636  032777  000100  173174          BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
005644  001047                          BNE     TST21           ;IF YES, SKIP THIS TEST
005646  000005                          RESET                   ;CLEAR EVERYTHING
005650  017703  174474                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
005654  012777  005724  174466          MOV     #2$,@RTCVT      ;SET UP RTC INTERRUPT VECTOR
005662  012777  000340  174462          MOV     #340,@RTCPSW    ;DISALLOW INTERRUPTS AFTER THE INTERRUPT
005670  004767  003674                  JSR     PC,WRPSW        ;SET PRIORITY TO 5
005674  000240                          .WORD   240
005676  052777  000100  174442          BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
005704  042777  000200  174434          BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
005712  105777  174430          1$:     TSTB    @LKS            ;WAIT FOR DONE
005716  100375                          BPL     1$
005720  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPT

005722  104047                          ERROR   47              ;RTC INTERRUPT DID NOT OCCUR

005724  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
005726  012777  005746  174414          MOV     #3$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
005734  004767  003630                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 5
005740  000240                          .WORD   240
005742  000240                          NOP                     ;GIVE SOME TIME FOR AN INTERRUPT
005744  000402                          BR      4$              ;NO INTERRUPT - BR TO END OF TEST

005746  022626                  3$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
005750  104050                          ERROR   50              ;INTERRUPT SEQUENCE DID NOT CLEAR
                                        ;INTERRUPT REQUEST

005752  042777  000100  174366  4$:     BIC     #BIT6,@LKS      ;DISABLE CLOCK INTERRUPTS
005760  010377  174364                  MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
```

```
                                ;;*****************************************************************
                                ;;TEST 21       TEST THAT INTERRUPT CLEARS WITH RESET
                                ;;*****************************************************************
005764  000004                TST21:  SCOPE
005766  005767  174304                  TST     @TSTFL          ;IS CONSOLE UNDER TEST?
005772  001442                  BEQ     TST22                   ;IF NOT, SKIP THIS TEST
005774  032777  000100  173036          BIT     #BIT6,@PSWR     ;ARE LINE CLOCK TESTS INHIBITED?
006002  001036                  BNE     TST22                   ;IF YES, SKIP THIS TEST
006004  004767  003560                  JSR     PC,WRPSW        ;SET PRIORITY TO 7
006010  000340                          .WORD   340
006012  017703  174332                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
006016  012777  006070  174324          MOV     #2$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
006024  052777  000100  174314          BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
006032  042777  000200  174306          BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
006040  105777  174302          1$:     TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
006044  100375                  BPL     1$
006046  000005                  RESET                           ;CLEAR PENDING INTERRUPT WITH RESET
006050  004767  003514                  JSR     PC,WRPSW        ;SET PRIORITY TO 5
006054  000240                          .WORD   240
006056  000240                  NOP                             ;GIVE TIME FOR ANY INTERRUPT
006060  042777  000100  174260          BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
006066  000402                  BR      3$                      ;BR TO END OF TEST

006070  022626          2$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
006072  104051                  ERROR   51                      ;RESET DID NOT CLEAR INTERRUPT

006074  010377  174250          3$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
```

```
                                        ;;*****************************************************************
                                        ;*TEST 22       TEST THAT RTC INTERRUPT CLEARS BY CLEARING BIT7 OF LKS
                                        ;;*****************************************************************
006100  000004                  TST22:  SCOPE
006102  005767  174170                  TST     CTSTFL          ;IS CONSOLE UNDER TEST?
006106  001444                  BEQ     TST23           ;IF NOT, SKIP THIS TEST
006110  032777  000100  172722          BIT     #BIT6,@SHR      ;ARE LINE CLOCK TESTS INHIBITED?
006116  001040                  BNE     TST23           ;IF YES, SKIP THIS TEST
006120  004767  003444                  JSR     PC,WRPSW        ;SET PRIORITY TO 7
006124  000340                          .WORD   340
006126  017703  174216                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
006132  012777  006210  174210          MOV     #2$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
006140  052777  000100  174200          BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
006146  042777  000200  174172          BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
006154  105777  174166          1$:     TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
006160  100375                  BPL     1$
006162  042777  000200  174156          BIC     #BIT7,@LKS      ;CLEAR DONE & INTERRUPT
006170  004767  003374                  JSR     PC,WRPSW        ;ALLOW INTERRUPTS
006174  000240                          .WORD   240
006176  000240                  NOP                     ;GIVE TIME FOR ANY INTERRUPT
006200  042777  000100  174140          BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
006206  000402                  BR      3$              ;BR TO END OF TEST

006210  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
006212  104052                  ERROR   52              ;CLEARING BIT7 OF LKS DID NOT CLEAR INTERRUPT

006214  010377  174130          3$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
```

```
                                ;;*****************************************************************
                                ;*TEST 23       TEST CLOCK REPEATABILITY
                                ;;*****************************************************************
006220  000004                  TST2J:  SCOPE
006222  005767  174050                  TST     CTSTFL          ;IS CONSOLE UNDER TEST?
006226  001452                  BEQ     TST24           ;IF NOT, SKIP THIS TEST
006230  032777  000100  172602  BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
006236  001046                  BNE     TST24           ;IF YES, SKIP THIS TEST

006240  005000                  CLR     R0              ;CLEAR A TIMER
006242  012701  177777          MOV     #-1,R1          ;SET A FLAG INDICATING FIRST PASS THRU THIS LOOP
006246  005002                  1$:     CLR     R2              ;CLEAR CLOCK COUNTER
006250  005077  174072          CLR     @LKS            ;CLEAR DONE
006254  105777  174066          2$:     TSTB    @LKS            ;SYNC ON DONE
006260  100375                  BPL     2$
006262  005077  174060          CLR     @LKS            ;CLEAR DONE
006266  105777  174054          3$:     TSTB    @LKS            ;IS CLOCK DONE?
006272  100003                  BPL     4$              ;BR IF NOT , TO INCREMENT TIMER
006274  005202                  INC     R2              ;IF DONE, INCREMENT CLOCK COUNT
006276  005077  174044          CLR     @LKS            ;CLEAR DONE
006302  005200                  4$:     INC     R0              ;INCREMENT TIMER
006304  001370                  BNE     3$              ;BR IF TIME REMAINS
006306  005201                  INC     R1              ;INCREMENT LOOP PASS FLAG
006310  001003                  BNE     CMPARE          ;BR IF TWO PASSES HAVE BEEN MADE
006312  010267  000032          MOV     R2,FIRST        ;IF NOT, STORE FIRST CLOCK COUNT
006316  000753                  BR      1$              ;DO LOOP AGAIN
006320  016701  000024          CMPARE: MOV     FIRST,R1        ;RECALL FIRST CLOCK COUNT
006324  160201                  SUB     R2,R1           ;CALCULATE DIFFERENCE OF TWO COUNTS
006326  100001                  BPL     TOLER           ;IF POSITIVE,SKIP NEGATION OF DIFFERENCE
006330  005401                  NEG     R1              ;MAKE DIFFERENCE A POSITIVE NUMBER
006332  020127  000000          TOLER:  CMP     R1,#0           ;COMPARE DIFFERENCE WITH DESIRED TOLERANCE
006336  003406                  BLE     TST24           ;BR, IF LOWER/EQUAL TO TOLERANCE

006340  010267  000006                  MOV     R2,SECND        ;STORE SECOND COUNT
006344  104053                  ERROR   53              ;CLOCK REPEATABILITY ERROR

006346  000402                  BR      TST24
006350  000000                  FIRST:  .WORD   0
006352  000000                  SECND:  .WORD   0
```

```
1513
1514
1515                                    ;;*****************************************************************
1516                                    ;*TEST 24       TEST THAT XMIT INTERRUPTS ONLY WHEN ENABLED
1517                                    ;;*****************************************************************
1518  006354  000004                    TST24:  SCOPE
1519  006356  042777  000100  173726            BIC     #BIT6,@TCSR     ;CLEAR TRANSMIT INTERRUPT ENABLE
1520  006364  105777  173722            1$:     TSTB    @TCSR           ;WAIT FOR DONE
1521  006370  100375                            BPL     1$
1522  006372  017703  173724                    MOV     @TVECT,R3       ;SAVE XMIT VECTOR
1523  006376  012777  006420  173716            MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
1524  006404  005077  173714                    CLR     @TPSW
1525  006410  004767  003154                    JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
1526  006414  000140                            .WORD   140
1527  006416  000402                            BR      3$
1528
1529  006420  022626                    2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1530  006422  104054                            ERROR   54
1531                                                                    ;XMIT INTERRUPTS WITH INTERRUPT ENABLE CLEAR
1532  006424  012777  006444  173670    3$:     MOV     #4$,@TVECT      ;SET XMIT VECTOR TO END OF TEST
1533  006432  052777  000100  173652            BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
1534  006440  000240                            NOP
1535
1536  006442  104055                            ERROR   55              ;XMIT DID NOT INTERRUPT
1537
1538  006444  042777  000100  173640    4$:     BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
1539  006452  022626                            CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1540  006454  010377  173642                    MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
1541
```

```
                                ;;*****************************************************************
                                ;;TEST 25       TEST THAT XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
                                ;;*****************************************************************
006460  000004                  TST25:  SCOPE
006462  042777  000100  173622          BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
006470  004767  003074                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
006474  000340                          .WORD   340
006476  017703  173620                  MOV     @TVECT,R3       ;SAVE XMIT VECTOR
006502  012777  006530  173612          MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
006510  105777  173576          1$:     TSTB    @TCSR           ;WAIT FOR DONE
006514  100375                          BPL     1$
006516  052777  000100  173566          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPT
006524  000240                          NOP
006526  000402                          BR      3$              ;CONTINUE TEST

006530  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
006532  104056                          ERROR   56
                                                                ;XMIT INTERRUPTS AT PRIORITY=7
006534  042777  000100  173550  3$:     BIC     #BIT6,@TCSR     ;CLEAR INTERRUPT ENABLE
006542  012777  006562  173552          MOV     #4$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
006550  004767  003014                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
006554  000140                          .WORD   140
006556  000240                          NOP
006560  000402                          BR      5$              ;BR TO END OF TEST-NO INTERRUPT

006562  022626                  4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
006564  104057                          ERROR   57
                                                                ;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
006566  010377  173530          5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
```

```
                                ;;*****************************************************************
                                ;*TEST 26       TEST TRANSMITTER FOR DOUBLE INTERRUPTS
                                ;;*****************************************************************
006572  000004                  TST26:  SCOPE
006574  042777  000100  173510          BIC     #BIT6,@TCSR     ;CLEAR INTERRUPT ENABLE
006602  017703  173514          MOV     @TVECT,R3       ;SAVE XMIT VECTOR
006606  012777  006642  173506          MOV     #2$,@TVECT      ;SET UP XMIT VECTOR
006614  004767  002750                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
006620  000140                          .WORD   140
006622  105777  173464          1$:     TSTB    @TCSR           ;WAIT FOR DONE
006626  100375                          BPL     1$
006630  052777  000100  173454          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
006636  000240                          NOP
006640  104060                          ERROR   60
                                                                ;XMIT INTERRUPT DID NOT OCCUR
006642  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
006644  012777  006664  173450          MOV     #4$,@TVECT      ;POINT XMIT VECTOR TO ERROR
006652  000240                          NOP
006654  042777  000100  173430          BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
006662  000402                          BR      5$              ;BR TO END OF TEST

006664  022626                  4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
006666  104061                          ERROR   61
                                                                ;XMIT RE-INTERRUPTED
006670  010377  173426          5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR

                                ;;*****************************************************************
                                ;*TEST 27       TEST THAT XMIT INTERRUPT CLEARS WITH LOADING TBUF
                                ;;*****************************************************************
006674  000004                  TST27:  SCOPE
006676  042777  000100  173406          BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
006704  004767  002660                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
006710  000340                          .WORD   340
006712  052777  000100  173372          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
006720  017703  173376          MOV     @TVECT,R3       ;SAVE XMIT VECTOR
006724  012777  006770  173370          MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR
006732  005077  173356          CLR     @TBUF           ;LOAD TBUF
006736  105777  173350          1$:     TSTB    @TCSR           ;WAIT FOR DONE (INTERRUPT)
006742  100375                          BPL     1$
006744  005077  173344          CLR     @TBUF           ;FILL SECOND BUFFER TO RESET INT.
006750  004767  002614          JSR     PC,WRPSW        ;ALLOW INTERRUPTS
006754  000140                          .WORD   140
006756  000240                  NOP
006760  042777  000100  173324          BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
006766  000402                          BR      3$              ;BR TO END OF TEST

006770  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
006772  104062                          ERROR   62
                                                                ;LOADING TBUF DID NOT CLEAR INTERRUPT.
006774  010377  173322          3$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
```

```
                                ;;*****************************************************************
                                ;*TEST 30       TEST THAT RCVR ACTIVE (BIT11) OF RCSR SETS WHILE RECEIVING
                                ;;*****************************************************************
007000  000004                  TST30:  SCOPE
007002  000005                          RESET                           ;CLEAR EVERYTHING
007004  052777  000004  173300          BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
007012  005077  173276                  CLR     @TBUF                   ;LOAD TRANSMIT BUFFER
007016  005000                          CLR     R0                      ;CLEAR A TIMER
007020  032777  004000  173260  WACTV:  BIT     #BIT11,@RCSR            ;TEST RCVR ACTIVE BIT
007026  001006                          BNE     2$                      ;BR IF SET
007030  005200                          INC     R0                      ;INCREMENT TIMER IF NOT SET
007032  001372                          BNE     WACTV                   ;CONTINUE WAIT IF TIME REMAINS
007034  042777  000004  173250          BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT

007042  104063                          ERROR   63                      ;RCVR ACTIVE DID NOT SET WHILE RECEIVING

007044  005000                  2$:     CLR     R0                      ;CLEAR TIMER
007046  105777  173234          WDONE:  TSTB    @RCSR                   ;CHECK FOR RECEIVER DONE
007052  100406                          BMI     3$                      ;BR, IF DONE
007054  005200                          INC     R0                      ;INCREMENT TIMER, IF NOT DONE
007056  001373                          BNE     WDONE                   ;CONTINUE WAIT IF TIME REMAINS
007060  042777  000004  173224          BIC     #BIT2,@RCSR             ;CLEAR MAINTENANCE BIT
007066  104064                          ERROR   64
                                                                        ;RECEIVER DONE NEVER SET

007070  032777  004000  173210  3$:     BIT     #BIT11,@RCSR            ;CHECK FOR RCVR ACTIVE CLEAR
007076  001404                          BEQ     4$                      ;BR, IF CLEAR
007100  042777  004000  173200          BIC     #BIT11,@RCSR            ;CLEAR MAINTENANCE BIT
007106  104065                          ERROR   65
                                                                        ;RCVR ACTIVE DID NOT CLEAR WITH RCVR DONE

007110  000005                  4$:     RESET                           ;CLEAR DONE WITH RESET
007112  105777  173170                  TSTB    @RCSR                   ;CHECK FOR DONE CLEAR
007116  001404                          BEQ     5$

007120  042777  000004  173164          BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
007126  104066                          ERROR   66
                                                                        ;RESET DID NOT CLEAR RCVR DONE

007130  042777  000004  173150  5$:     BIC     #BIT2,@RCSR             ;CLEAR MAINTENANCE BIT
007136  000400                          BR      TST31                   ;BR TO NEXT TEST
```

```
                                        ;;*************************************************************
                                        ;*TEST 31       TEST THAT RDR ENABLE CLEARS RECEIVER DONE FLAG
                                        ;;*************************************************************
007140  000004                  TST31:  SCOPE
007142  000005                          RESET                           ;CLEAR EVERYTHING
007144  052777  000004  173140          BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
007152  005077  173136                  CLR     @TBUF                   ;LOAD TRANSMITTER
007156  105777  173124          1$:     TSTB    @RCSR                   ;WAIT FOR RECEIVER DONE
007162  100375                          BPL     1$
007164  052777  000001  173114          BIS     #BIT0,@RCSR             ;CLEAR DONE BY SETTING RDR ENABLE
007172  105777  173110                  TSTB    @RCSR                   ;CHECK FOR DONE CLEAR
007176  001404                          BEQ     TST32                   ;BR, IF CLEAR TO NEXT TEST
007200  042777  000004  173104          BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
007206  104067                          ERROR   67
                                                                        ;SETTING RDR ENABLE DID NOT CLEAR RCVR DONE


                                        ;;*************************************************************
                                        ;*TEST 32       TEST THAT READING RBUF CLEARS RECEIVER DONE
                                        ;;*************************************************************
007210  000004                  TST32:  SCOPE
007212  000005                          RESET                           ;CLEAR EVERYTHING
007214  052777  000004  173070          BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
007222  005077  173066                  CLR     @TBUF                   ;LOAD TRANSMITTER
007226  105777  173054          1$:     TSTB    @RCSR                   ;WAIT FOR RECEIVER DONE
007232  100375                          BPL     1$
007234  017700  173050                  MOV     @RBUF,R0                ;READ RECEIVE BUFFER
007240  042777  000004  173044          BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
007246  105777  173034                  TSTB    @RCSR                   ;CHECK FOR RECEIVE DONE CLEAR
007252  001401                          BEQ     TST33                   ;BR, IF CLEAR TO NEXT TEST
007254  104070                          ERROR   70
                                                                        ;READING RBUF DID NOT CLEAR RCVR DONE
```

```
                                        ;;*****************************************************************
                                        ;*TEST 33       TEST THAT RCVR INTERRUPTS ONLY WHEN ENABLED
                                        ;;*****************************************************************
007256  000004                  TST33:  SCOPE
007260  042777  000100  173024          BIC     #BIT6,@TCSR     ;DISABLE TRANSMIT INTERRUPTS
007266  042777  000100  173012          BIC     #BIT6,@RCSR     ;DISABLE RECEIVER INTERRUPTS
007274  052777  000004  173010          BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
007302  017703  173010                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
007306  012777  007350  173002          MOV     #2$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
007314  005077  173000                  CLR     @RPSW           ;PSW VECTOR CLEAR
007320  004767  002244                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
007324  000140                          .WORD   140
007326  005077  172762                  CLR     @TBUF           ;SEND A CHARACTER
007332  105777  172750          1$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
007336  100375                          BPL     1$
007340  042777  000004  172744          BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
007346  000405                          BR      3$              ;CONTINUE TEST
007350  042777  000004  172734  2$:     BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
007356  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
007360  104071                          ERROR   71
                                                                ;RECEIVER INTERRPUTS WITH INT. ENABLE CLEAR
007362  012777  007412  172726  3$:     MOV     #4$,@RVECT      ;POINT RCV VECTOR TO END OF TEST
007370  052777  000100  172710          BIS     #BIT6,@RCSR     ;ENABLE RCV INTERRUPTS
007376  000240                          NOP                     ;GIVE ANY INTERRUPTS TIME
007400  042777  000004  172704          BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
007406  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
007410  104072                          ERROR   72
                                                                ;RCVR DID NOT INTERRUPT
007412  042777  000004  172672  4$:     BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
007420  010377  172672                  MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
007424  042777  000100  172654          BIC     #BIT6,@RCSR     ;DISABLE INTERRUPTS
```

```
                                ;;*****************************************************************
                                ;*TEST 34       TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
                                ;;*****************************************************************
007432  000004          TST34:  SCOPE
007434  000005                  RESET                           ;CLEAR EVERYTHING
007436  052777  000004  172646          BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
007444  004767  002120                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
007450  000340                          .WORD   340
007452  017703  172640                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
007456  012777  007514  172632          MOV     #2$,@RVECT      ;POINT RCVR VECTOR TO ERROR REPORT
007464  005077  172630                  CLR     @RPSW           ;PSW VECTOR CLEAR
007470  005077  172620                  CLR     @TBUF           ;SEND A CHARACTER
007474  105777  172606          1$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
007500  100375                          BPL     1$
007502  052777  000100  172576          BIS     #BIT6,@RCSR     ;ENABLE INTERRUPTS
007510  000240                          NOP                     ;GIVE TIME FOR INTERRUPT
007512  000405                          BR      3$              ;CONTINUE TEST
007514  042777  000004  172570  2$:     BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
007522  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
007524  104073                          ERROR   73
                                                                ;RCVR INTERRUPTS AT PRIORITY 7
007526  042777  000100  172552  3$:     BIC     #BIT6,@RCSR     ;CLEAR INTERRUPT ENABLE
007534  012777  007562  172554          MOV     #4$,@RVECT      ;POINT RCVR VECTOR TO ERROR REPORT
007542  004767  002022                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
007546  000140                          .WORD   140
007550  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPT
007556  042777  000004  172532          BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
007560  000405                          BR      5$              ;BR TO END OF TEST, IF NO INTERRUPT

007562  042777  000004  172522  4$:     BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
007570  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
007572  104074                          ERROR   74
                                                                ;RCVR INTERRUPT REQUEST PASSED WITH BIT6 CLEAR
007574  010377  172516          5$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
```

```
                                ;;*****************************************************************
                                ;*TEST 35       TEST RECEIVER FOR DOUBLE INTERRUPTS
                                ;;*****************************************************************
007600  000004          TST35:  SCOPE
007602  000005                  RESET                           ;CLEAR EVERYTHING
007604  052777  000004  172500  BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
007612  017703  172500          MOV     @RVECT,R3               ;SAVE RECEIVE VECTOR
007616  012777  007670  172472  MOV     #2$,@RVECT              ;POINT RCV VECTOR TO CONTINUE TEST
007624  005077  172470          CLR     @RPSW                   ;CLEAR PSW VECTOR
007630  004767  001734          JSR     PC,@WRPSW               ;SET PSW TO PRIORITY 3
007634  000140                  .WORD   140
007636  005077  172452          CLR     @TBUF                   ;SEND A CHARACTER
007642  105777  172440    1$:   TSTB    @RCSR                   ;WAIT FOR RCVR DONE
007646  100375                  BPL     1$
007650  042777  000004  172434  BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
007656  052777  000100  172422  BIS     #BIT6,@RCSR             ;ENABLE RCV INTERRUPTS
007664  000240                  NOP                             ;GIVE SOME TIME
007666  104075                  ERROR   75
                                                                ;RCVR INTERRUPT DID NOT OCCUR
007670  022626          2$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
007672  012777  007716  172416  MOV     #3$,@RVECT              ;POINT RCV VECTOR TO ERROR REPORT
007700  000240                  NOP                             ;GIVE SOME TIME
007702  042777  000100  172376  BIC     #BIT6,@RCSR             ;CLEAR INTERRUPT ENABLE
007710  010377  172402          MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
007714  000402                  BR      4$                      ;BR TO END OF TEST
007716  022626          3$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
007720  104076                  ERROR   76
                                                                ;RECEIVER RE-INTERRUPTED
007722  010377  172370    4$:   MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
```

```
 1809
 1810
 1811                                   ;;*****************************************************************
 1812                                   ;*TEST 36       TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
 1813                                   ;;*****************************************************************
 1814  007726  000004                  TST36:  SCOPE
 1815  007730  000005                          RESET                           ;CLEAR EVERYTHING
 1816  007732  004767  001632                  JSR     PC,WRPSW                ;SET PSW PRIORITY TO 7
 1817  007736  000340                          .WORD   340
 1818  007740  017703  172352                  MOV     @RVECT,R3               ;SAVE RECEIVE VECTOR
 1819  007744  012777  010036  172344          MOV     #2$,@RVECT              ;POINT RCV VECTOR TO ERROR REPORT
 1820  007752  005077  172342                  CLR     @RPSW                   ;CLEAR PSW VECTOR
 1821  007756  052777  000100  172322          BIS     #BIT6,@RCSR             ;SET RCVR INTERRUPT ENABLE
 1822  007764  052777  000004  172320          BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
 1823  007772  005077  172316                  CLR     @TBUF                   ;SEND A CHARACTER
 1824  007776  105777  172304          1$:     TSTB    @RCSR                   ;WAIT FOR DONE
 1825  010002  100375                          BPL     1$
 1826  010004  042777  000004  172300          BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
 1827  010012  005077  172272                  CLR     @RBUF                   ;READ RBUF TO CLEAR PENDING INTERRPUT
 1828  010016  004767  001546                  JSR     PC,WRPSW                ;SET PSW TO PRIORITY 3
 1829  010022  000140                          .WORD   140
 1830  010024  000240                          NOP
 1831  010026  042777  000100  172252          BIC     #BIT6,@RCSR             ;NO INTERRUPT-CLEAR INT. ENABLE
 1832  010034  000402                          BR      3$
 1833
 1834  010036  022626                  2$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
 1835  010040  104077                          ERROR   77                      ;READING RBUF DID NOT CLEAR INTERRUPT
 1836
 1837  010042  010377  172250          3$:     MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
 1838
```

```
 1839
 1840
 1841                           ;;*****************************************************************
 1842                           ;*TEST 37       TEST THAT RESET CLEARS RECEIVE INTERRUPT
 1843                           ;;*****************************************************************
 1844  010046  000004          TST37:  SCOPE
 1845  010050  000005                  RESET                           ;CLEAR EVERYTHING
 1846  010052  004767  001512          JSR     PC,@RPSW                ;SET PSW TO PRIORITY 7
 1847  010056  000340                  .WORD   340
 1848  010060  017703  172232          MOV     @RVECT,R3               ;SAVE RECEIVE VECTOR
 1849  010064  012777  010150  172224  MOV     #2$,@RVECT              ;POINT RCV VECTOR TO ERROR REPORT
 1850  010072  005077  172222          CLR     @PSW                    ;CLEAR PSW VECTOR
 1851  010076  052777  000100  172202  BIS     #BIT6,@RCSR             ;SET RCV INTERRUPT ENABLE
 1852  010104  052777  000004  172200  BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
 1853  010112  012777  000377  172174  MOV     #377,@TBUF              ;SEND AN ALL 1'S CHARACTER
 1854  010120  105777  172162  1$:     TSTB    @RCSR                   ;WAIT FOR RCV DONE
 1855  010124  100375                  BPL     1$
 1856  010126  000005                  RESET                           ;CLEAR RCV INTERRUPT & RBUF
 1857  010130  004767  001434          JSR     PC,@RPSW                ;SET PSW TO PRIORITY 3
 1858  010134  000140                  .WORD   140
 1859  010136  000240                  NOP
 1860  010140  042777  000100  172140  BIC     #BIT6,@RCSR             ;NO INTERRUPT-CLEAR INT. ENABLE
 1861  010146  000402                  BR      3$                      ;CONTINUE TEST
 1862
 1863
 1864  010150  022626          2$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
 1865  010152  104100                  ERROR   100                     ;RESET DID NOT CLEAR RCVR INTERRUPT
 1866
 1867  010154  010377  172136  3$:     MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
```

```
                                        ;;*****************************************************************
                                        ;*TEST 40       TEST THAT THE "OR" ERROR (BIT14) & "ERROR" (BIT15) CAN BE SET
                                        ;;*****************************************************************
010160  000004                  TST40:  SCOPE
010162  032777  002000  170650          BIT     #BIT10,@SWR     ;IS THIS TEST ENABLED
010170  001440                          BEQ     TST41           ;IF NOT ENABLED, BR TO NEXT TEST
010172  000005                          RESET                   ;CLEAR EVERYTHING
010174  052777  000004  172110          BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
010202  012700  000003                  MOV     #3,R0           ;SET CHARACTER COUNT TO SEND 3 CHAR.
010206  005077  172102          1$:     CLR     @TBUF           ;LOAD TRANSMIT BUFFER
010212  105777  172074          2$:     TSTB    @TCSR           ;WAIT FOR TRANSMIT DONE
010216  100375                          BPL     2$
010220  005300                          DEC     R0              ;DECREMENT CHARACTER COUNT
010222  001371                          BNE     1$              ;BR IF ALL CHARACTERS NOT TRANSMITTED
010224  032777  040000  172056          BIT     #BIT14,@RBUF    ;TEST FOR "OR" ERROR FLAG
010232  001004                          BNE     3$              ;BR, IF SET
010234  042777  000004  172050          BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
010242  104101                          ERROR   101
                                                                ;"OR" ERROR FLAG DID NOT SET
010244  032777  100000  172036  3$:     BIT     #BIT15,@RBUF    ;TEST "ERROR" FLAG
010252  001004                          BNE     4$              ;BR, IF SET
010254  042777  000004  172030          BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
010262  104102                          ERROR   102
                                                                ;"ERROR" FLAG DID NOT SET WITH "OR" FLAG
010264  042777  000004  172020  4$:     BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
```

```
1896
1897
1898                                    ;;*****************************************************************
1899                                    ;*TEST 41        TEST THAT BREAK TRANSMITS ALL ZEROES
1900                                    ;;*****************************************************************
1901  010272  000004                    TST41:  SCOPE
1902  010274  032777  000400  170536            BIT     #BIT8,@SHR      ;IS BREAK FUNCTION TEST ENABLED?
1903  010302  001444                            BEQ     TST42           ;BR TO NEXT TEST, IF NOT ENABLED
1904  010304  000005                            RESET                   ;CLEAR EVERYTHING
1905  010306  052777  000004  171776            BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
1906  010314  012777  177777  171772            MOV     #-1,@TBUF       ;TRANSMIT ALL ONES TO RCVR
1907  010322  105777  171760          1$:       TSTB    @RCSR           ;WAIT FOR RCVR DONE
1908  010326  100375                            BPL     1$
1909  010330  005077  171754                    CLR     @RBUF           ;CLEAR DONE (LEAVING ALL ONES IN RBUF)
1910  010334  052777  000001  171750            BIS     #BIT0,@TCSR     ;TRANSMIT BREAK
1911  010342  005000                            CLR     R0              ;CLEAR A TIMER
1912  010344  117767  171736  170454  2$:       MOVB    @RCSR,$BODAT    ;WAIT FOR RCVR DONE
1913  010352  100406                            BMI     CONT41          ;BR IF DONE
1914  010354  005200                            INC     R0              ;IF NOT, INCREMENT TIMER
1915  010356  001372                            BNE     2$              ;BR IF TIME REMAINS
1916
1917  010360  042777  000005  171724            BIC     #BIT0!BIT2,@TCSR  ;CLEAR MAINTENANCE & BREAK BITS
1918  010366  104103                            ERROR   103             ;BREAK DID NOT TRANSMIT ANYTHING
1919
1920  010370  105777  171714          CONT41: TSTB    @RBUF           ;CHECK RECEIVE BUFFER FOR ZERO
1921  010374  001404                            BEQ     3$              ;BR, IF ZERO
1922  010376  042777  000005  171706            BIC     #BIT0!BIT2,@TCSR  ;CLEAR MAINTENANCE & BREAK BITS
1923
1924  010404  104103                            ERROR   103             ;BREAK DID NOT TRANSMIT ALL ZEROES
1925
1926  010406  042777  000005  171676  3$:       BIC     #BIT0!BIT2,@TCSR ;CLEAR MAINTENANCE & BREAK BITS
```

```
 1927
 1928
 1929                                   ;;*****************************************************************
 1930                                   ;*TEST 42       TEST THAT "FR" ERROR CAN BE SET DURING BREAK
 1931                                   ;;*****************************************************************
 1932  010414  000004                  TST42:  SCOPE
 1933  010416  032777  002000  170414           BIT     #BIT10,@SWR     ;IS THE "TEST ERROR FLAGS" BIT SET
 1934  010424  001435                          BEQ     TST43           ;BR TO NEXT TEST, IF NOT SET
 1935  010426  032777  000400  170404           BIT     #BIT8,@SWR      ;IS BREAK FUNCTION ENABLED
 1936  010434  001431                          BEQ     TST43           ;BR TO NEXT TEST, IF NOT SET
 1937  010436  000005                          RESET                   ;CLEAR EVERYTHING
 1938  010440  052777  000004  171644           BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
 1939  010446  052777  000001  171636           BIS     #BIT0,@TCSR     ;SEND BREAK
 1940  010454  005077  171634                  CLR     @TBUF           ;TRANSMIT A CHARACTER TO TIME BREAK
 1941  010460  105777  171626          1$:     TSTB    @TCSR           ;WAIT FOR XMIT DONE
 1942  010464  100375                          BPL     1$
 1943  010466  005077  171622                  CLR     @TBUF           ;FILL SECOND BUFFER
 1944  010472  105777  171614          2$:     TSTB    @TCSR           ;WAIT ONE CHARACTER TIME (DONE)
 1945  010476  100375                          BPL     2$
 1946  010500  042777  000005  171604           BIC     #BIT0!BIT2,@TCSR ;CLEAR MAINTENANCE & BREAK BITS
 1947  010506  032777  020000  171574           BIT     #BIT13,@RBUF    ;CHECK FOR FRAMING ERROR FLAG
 1948  010514  001001                          BNE     TST43           ;BR, IF SET
 1949
 1950  010516  104104                          ERROR   104
 1951                                                                   ;BREAK DID NOT SET FRAMING ERROR
```

```
                                        ;;*****************************************************************
                                        ;*TEST 43       TEST DATA PATH FROM TRANSMITTER TO RECEIVER USING MAINTENANCE WRAP
                                        ;;*****************************************************************
010520  000004                TST43:  SCOPE
010522  000005                        RESET                   ;CLEAR EVERYTHING
010524  005001                        CLR     R1              ;CLEAR REGISTER FOR TEST DATA
010526  052777  000004  171556        BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
010534  010177  171554        1$:     MOV     R1,@TBUF        ;XMIT A CHARACTER
010540  105777  171542        2$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
010544  100375                        BPL     2$
010546  017702  171536                MOV     @RBUF,R2        ;GET RECEIVED CHARACTER
010552  020102                        CMP     R1,R2           ;COMPARE DATA
010554  001003                        BNE     3$              ;BR, IF NON-COMPARE
010556  105201                        INCB    R1              ;INCREMENT TEST DATA
010560  001411                        BEQ     4$              ;BR, IF FINISHED
010562  000764                        BR      1$              ;CONTINUE IF NOT
010564  010167  170234        3$:     MOV     R1,$GDDAT               ;STORE THE EXPECTED DATA
010570  010267  170232                MOV     R2,$BDDAT               ;STORE RECEIVED DATA
010574  042777  000004  171510        BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
010602  104105                        ERROR   105             ;DATA COMPARE DATA

010604  042777  000004  171500  4$:   BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT



                                        ;;*****************************************************************
                                        ;*TEST 44       TEST DATA PATHS USING WRAP CABLE
                                        ;;*****************************************************************
010612  000004                TST44:  SCOPE
010614  032777  000200  170216        BIT     #BIT7,@SWR      ;IS THIS TEST ENABLED
010622  001427                        BEQ     TST45           ;BR, IF NOT
010624  005001                        CLR     R1              ;CLEAR REGISTER FOR TEST DATA
010626  000005                        RESET                   ;CLEAR EVERYTHING
010630  010177  171460        1$:     MOV     R1,@TBUF        ;XMIT A CHARACTER
010634  005000                        CLR     R0              ;CLEAR A TIMER
010636  105777  171444        2$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
010642  100403                        BMI     3$              ;BR IF DONE
010644  005200                        INC     R0              ;INCREMENT TIMER IF NOT
010646  001373                        BNE     2$              ;BR IF TIME REMAINS

010650  104064                        ERROR   64              ;RECEIVER DONE NOT SET

010652  017702  171432        3$:     MOV     @RBUF,R2        ;GET RECEIVED CHARACTER
010656  020102                        CMP     R1,R2           ;COMPARE DATA
010660  001003                        BNE     4$              ;BR, IF NON-COMPARE
010662  105201                        INCB    R1              ;INCREMENT TEST DATA
010664  001406                        BEQ     TST45           ;BR, IF FINISHED
010666  000760                        BR      1$              ;CONTINUE IF NOT
010670  010167  170130        4$:     MOV     R1,$GDDAT       ;STORE EXPECTED DATA
010674  010267  170126                MOV     R2,$BDDAT       ;STORE RECEIVED DATA

010700  104106                        ERROR   106             ;DATA COMPARE ERROR WITH WRAP CABLE
```

```
                                ;;*****************************************************************
                                ;*TEST 45       TEST DL11-W LOGIC BY EXERCISING THE XMIT,RECEIVE, & CLOCK(IF AVAILABLE)
                                ;;*****************************************************************
010702  000004                TST45:  SCOPE
010704  000005                        RESET                           ;CLEAR EVERYTHING
010706  017703  171410                MOV     @TVECT,R3               ;SAVE XMIT VECTOR
010712  017704  171400                MOV     @RVECT,R4               ;SAVE RECEIVE VECTOR
010716  012777  011214  171376        MOV     #XMIT,@TVECT            ;POINT TRANSMIT VECTOR TO TRANSMIT ROUTINE
010724  005077  171374                CLR     @TPSW                   ;ALLOW INTERRUPTS AFTER XMIT INTERRUPT
010730  012777  011250  171360        MOV     #RCV,@RVECT             ;POINT RECEIVE VECTOR TO RECEIVE ROUTINE
010736  005077  171356                CLR     @RPSW                   ;ALLOW INTERRPUTS AFTER RCVR INTERRUPT
010742  005767  171330                TST     CTSTFL                  ;IS CONSOLE UNDER TEST?
010746  001416                        BEQ     1$                      ;IF NOT SKIP CLOCK SET UP
010750  032777  000100  170062        BIT     #BIT6,@SWR              ;IF YES, ARE CLOCK TEST DISABLED?
010756  001012                        BNE     1$                      ;IF YES, SKIP CLOCK SET UP
010760  017705  171364                MOV     @RTCVT,R5               ;SAVE CLOCK VECTOR
010764  012777  011264  171356        MOV     #CLK,@RTCVT             ;POINT VECTOR TO CLOCK INTERRUPT ROUTINE
010772  005077  171354                CLR     @RTCPSW                 ;ALLOW INTERRUPTS AFTER CLOCK INTERRUPT
010776  052777  000100  171342        BIS     #BIT6,@LKS              ;ENABLE CLOCK INTERRUPTS
011004  052777  000004  171300  1$:    BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
011012  052777  000100  171272        BIS     #BIT6,@TCSR             ;ENABLE TRANSMIT INTERRUPTS
011020  052777  000100  171260        BIS     #BIT6,@RCSR             ;ENABLE RECEIVE INTERRUPTS
011026  005067  000242                CLR     XMTCNT                  ;CLEAR XMIT INTERRUPT COUNTER
011032  005067  000234                CLR     RCVCNT                  ;CLEAR RCV INTERRUPT COUNTER
011036  004767  000526                JSR     PC,WRPSW                ;SET PSW TO PRIORITY 3
011042  000140                                WORD    140
011044  005001                        CLR     R1                      ;CLEAR A REGISTER FOR TEST DATA USE
011046  005000                        CLR     R0                      ;CLEAR TIMER
011050  012702  011300                MOV     #BUF,R2                 ;POINT R2 TO RECEIVE DATA STORAGE
011054  005077  171234                CLR     @TBUF                   ;SEND FIRST CHARACTER
011060  005200                2$:     INC     R0                      ;WAIT FOR INTERRPUTS
011062  001376                        BNE     2$
011064  032777  000100  171220        BIT     #BIT6,@TCSR             ;FINISHED ENTIRE TRANSMISSION
011072  001402                        BEQ     3$                      ;BR, IF INTERRPUTS ARE DISABLED (FINISHED)
011074  000005                        RESET                           ;CLEAR EVERYTHING

011076  104107                        ERROR   107                     ;TRANSMIT INTERRUPT TIMEOUT IN MAIN. DATA TEST

011100  026767  000170  000164  3$:    CMP     XMTCNT,RCVCNT           ;COMPARE THE NUMBER OF INTERRUPTS
011106  001402                        BEQ     4$                      ;BR, IF EQUAL
011110  000005                        RESET                           ;CLEAR EVERYTHING

011112  104110                        ERROR   110                     ;RECEIVER DID NOT GET FULL TRANSMISSION
                                                                        ;   IF RCVCNT=0, NO DATA RECEIVED
                                                                        ;   IF RCVCNT?0, TEN (XMTCNT-RCVCNT)
                                                                        ;   EQUALS THE NO, OF INTERRUPTS LOST.
```

```
                                                ;IS CONCOLE UNDER TEST?
                                                ;IF NOT, SKIP CLOCK COUNT CHECK
                                                ;IF YES, ARE CLOCK TESTS DISABLED?
                                                ;IF YES, SKIP CLOCK COUNT CHECK
                                                ;CHECK FOR AT LESST ONE CLOCK INTERRUPT
                                                ;BR IF INTERRUPTS OCCURRED

                                                ;NO CLOCK INTERRUPTS IN EXERCISER

                                                ;CLEAR EVERYTHING
                                                ;LOAD RECEIVED DATA POINTER TO R0
                                                ;SET UP REGISTER FOR COMPARISON
                                                ;COMPARE XMIT & RCV DATA
                                                ;BR, IF NOT EQUAL
                                                ;INCREMENT COMPARE DATA
                                                ;FINISHED CHECKING RECEIVED DATA?
                                                ;BR, IF NOT FINISHED
                                                ;BR TO END OF TEST

                                                ;STORE BAD DATA FOR ERROR REPORT
                                                ;STORE GOOD DATA FOR ERROR REPORT
                                                ;DATA COMPARE ERROR IN EXERCISER

                                                ;RESTORE XMIT VECTOR
                                                ;RESTORE RECEIVE VECTOR
                                                ;BR TO END OF DEVICE TEST ROUTINE

                                                ;INCREMENT XMIT INTERRUPT COUNTER
                                                ;INCREMENT TEST DATA
                                                ;SEND DATA PATTERN FROM 00 --> 37
                                                ;BR, IF MORE DATA TO BE SENT
                                                ;CLEAR XMIT INTERRUPT ENABLE
                                                ;RETURN, WITHOUT SENDING ANY MORE DATA
                                                ;SEND NEW CHARACTER
                                                ;CLEAR TIMER
                                                ;RETURN

                                                ;STORE RECEIVED DATA
                                                ;INCREMENT RCV INTERRUPT COUNTER
                                                ;CLEAR TIMER
                                                ;RETURN

                                                ;INCREMENT CLOCK INTERRUPT COUNT
                                                ;RETURN

                                                ;RCVCNT: .WORD 0
                                                ;XMTCNT: .WORD 0
                                                ;CLKCNT: .WORD 0
                                                ;BUF:    .BLKW 40
```

```
                                        ;END OF DEVICE PASS ROUTINE

011400  005067  167376          ENDEV:  CLR     $TSTNM          ;CLEAR TEST NO. COUNT FOR SCOPE ROUTINE
011404  005267  167466                  INC     $DEVCT          ;INCREMENT DEVICE COUNTER
011410  026767  170666  167460          CMP     TMP2,$DEVCT     ;ALL DEVICES TESTED
011416  001404                          BEQ     $EOP            ;BR, IF YES
011420  005067  170652                  CLR     CTSTFL          ;CLEAR CONSOLE UNDER TEST FLAG
011424  000167  171754                  JMP     TSTDEV          ;GO TEST NEXT DEVICE


                                        .SBTTL  END OF PASS ROUTINE

                                        ;;**************************************************************
                                        ;;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;;*IF THERES A MONITOR GO TO IT
                                        ;;*IF THERE ISN'T JUMP TO GOAGIN

011430                          $EOP:
                                        SCOPE
011430  000004
011432  005067  167344                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
011436  005267  167432                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
011442  042767  100000  167424          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
011450  005327                          DEC     (PC)+           ;;LOOP?
011452  000001          $EOPCT: .WORD   1
011454  003015                          BGT     $DOAGN          ;;YES
011456  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
011460  000001          $ENDCT: .WORD   1
011462  011452                          $EOPCT
011464  104401  011520                  TYPE    ,ENDMG          ;;TYPE "END PASS"
011470  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
011474  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
011476  000005                          RESET                   ;;CLEAR THE WORLD
011500  004710                  $ENDMO: JSR     PC,(R0)         ;;GO TO MONITOR
011502  000240                          NOP                     ;;SAVE ROOM
011504  000240                          NOP                     ;;FOR
011506  000240                          NOP                     ;;ACT11
011510                          $DOAGN:
011510  000137                          JMP     @(PC)+          ;;RETURN
011512  011534          $RTNAD: .WORD   GOAGIN
011514    377     377     000   $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
        011520                          .EVEN
011520  005015  047105  020104  ENDMG:  .ASCIZ  <CR><LF>/END PASS /
011526  040520  051523  000040
```

```
 2153
 2154  011534  005067  167336          GOAGIN: CLR     $DEVCT          ;CLEAR DEVICE COUNT
 2155  011540  022767  000001  170534          CMP     #1,TMP2         ;IS THERE ONLY ONE DEVICE UNDER TEST?
 2156  011546  001004                          BNE     RSTRT           ;BR, IF NOT
 2157  011550  012706  001000                  MOV     #1000,SP                ;RESET STACK POINTER
 2158  011554  000167  171746                  JMP     TST1            ;GO DO ANOTHER PASS
 2159
 2160  011560  005067  167314          RSTRT:  CLR     $UNIT           ;CLEAR UNIT NUMBER
 2161  011564  000167  171540                  JMP     BEGIN
 2162
 2163  011570  011646                  WRPSW:  MOV(SP),-(SP)           ;COPY RETUN PC
 2164  011572  013616                          MOV     @(SP)+,(SP)     ;MOVE NEW PSW TO STACK
 2165  011574  062746  000002                  ADD     #2,-(SP)        ;ADJUST JSR RETURN
 2166  011600  000006                          RTT                     ;POP RETURN PC & NEW PSW
 2167
 2168                                  ;SUBROUTINE TO REPORT UNEXPECTED OR ERRONEOUS TRAPS OR INTERRUPTS
 2169
 2170  011602  012600                  CATCH:  MOV     (SP)+,R0        ;GET ADDRESS OF TRAP VECTOR + 4
 2171  011604  162700  000004                  SUB     #4,R0           ;ADJUST TO POINT TO TRAP ADDRESS
 2172  011610  010067  000014                  MOV     R0,BOVECT       ;STORE TRAP OR INTERRUPT ADDRESS
 2173  011614  016667  000002  000004          MOV     2(SP),OLDPC     ;GET PC WHERE TRAP OR INTERRUPT OCCURRED
 2174  011622  104112                          ERROR   112             ;REPORT ERROR
 2175
 2176  011624  000000                          HALT                    ;PROGRAM MUST BE RESTARTED AT THIS POINT
 2177  011626  000000          OLDPC:  .WORD   0
 2178  011630  000000          BOVECT: .WORD   0
 2179
```

```
                                        ;;*****************************************************************
                                        ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                        ;*SAVE THE ERROR ITEM NUMBER AND ADDRESS OF THE ERROR CALL
                                        ;*AND GO TO $ERRTYP ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1         HALT ON ERROR
                                        ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                        ;*SW09=1         LOOP IN ERROR
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
                                        ;;*****************************************************************

011632                                  $ERROR:
011632  105267  167145                  7$:     INCB    $ERFLG          ;SET THE ERROR FLAG
011636  001775                          BEQ     7$              ;DON'T LET FLAG GO TO ZERO
011640  016777  167136  167174          MOV     $TSTNM,@DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
011646  005267  167140                  INC     $ERTTL          ;INCREMENT ERROR COUNT
011652  011667  167140                  MOV     (SP),$ERRPC     ;GET ADDRESS OF ERROR INSTRUCTION
011656  162767  000002  167132          SUB     #2,$ERRPC
011664  117767  167126  167122          MOVB    @$ERRPC,$ITEMB  ;STRIP AND SAVE THE ERROR ITEM CODE
011672  032777  020000  167140          BIT     #BIT13,@SWR     ;SKIP TYPEOUT IF SET
011700  001004                          BNE     20$             ;SKIP TYPEOUTS
011702  004767  000106                  JSR     PC,$ERRTYP      ;GO TO USER ERROR ROUTINE
011706  104401  001063                  TYPE    ,$CRLF
011712                                  20$:
011712  122767  000001  167166          CMPB    #APTENV,$ENV    ;RUNNING IN APT MODE
011720  001007                          BNE     2$              ;NO, SKIP APT ERROR REPORT
011722  116767  167066  000004          MOVB    $ITEMB,21$      ;SET ITEM NUMBER AS ERROR NUMBER
011730  004767  000562                  JSR     PC,$ATY4        ;REPORT FATAL ERROR TO APT
011734     000                  21$:    .BYTE   0
011735     000                          .BYTE   0
011736  000777                  22$:    BR      22$             ;APT ERROR LOOP
011740  005777  167074          2$:     TST     @SWR            ;HALT ON ERROR
011744  100001                          BPL     3$              ;SKIP IF CONTINUE
011746  000000                          HALT                    ;HALT ON ERROR!
011750  104406                  3$:     CKSWR                   ;TEST FOR CHANGE IN SOFT-SWR
011752  032777  001000  167060          BIT     #BIT09,@SWR     ;LOOP ON ERROR SWITCH SET?
011760  001402                          BEQ     4$              ;BR IF NO
011762  016716  167022                  MOV     $LPERR,(SP)     ;FUDGE RETURN FOR LOOPING
011766  005767  167066          4$:     TST     $ESCAPE         ;CHECK FOR AN ESCAPE ADDRESS
011772  001402                          BEQ     5$              ;BR IF NONE
011774  016716  167060                  MOV     $ESCAPE,(SP)    ;FUDGE RETURN ADDRESS FOR ESCAPE
012000                                  5$:
012000  022737  011500  000042          CMP     #$ENDAD,@#42    ;ACT-11 AUTO-ACCEPT?
012006  001001                          BNE     6$              ;BR IF NO
012010  000000                          HALT                    ;YES
012012                                  6$:
012012  000002                          RTI                     ;RETURN
```

```
                                        .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

                                        ;;**************************************************************
                                        ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
                                        ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
                                        ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

012014                                  $ERRTYP:
012014  104401  001063                          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
012020  010046                          MOV     R0,-(SP)        ;;SAVE R0
012022  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
012024  153700  001014                  BISB    @#$ITEMB,R0
012030  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                ;;TYPE THE PC OF THE ERROR
012032  016746  166760                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                ;;ERROR ADDRESS
012036  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
012040  000426                          BR      6$              ;;GET OUT
012042  005300                  1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
012044  006300                          ASL     R0              ;;      WORK FOR THE ERROR TABLE
012046  006300                          ASL     R0
012050  006300                          ASL     R0
012052  062700  001146                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
012056  012067  000004                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
012062  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
012064  104401                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
012066  000000                  2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
012070  104401  001063                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
012074  012067  000004                  MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
012100  001404                  3$:     BEQ     5$              ;;SKIP TYPEOUT IF 0
012102  104401                          TYPE                    ;;TYPE THE "DATA HEADER"
012104  000000                  4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
012106  104401  001063                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
012112  011000                  5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
012114  001004                          BNE     7$              ;;GO TYPE THE DATA
012116  012600                  6$:     MOV     (SP)+,R0        ;;RESTORE R0
012120  104401  001063                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
012124  000207                          RTS     PC              ;;RETURN
012126                          7$:
012126  013046                          MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
012130  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
012132  005710                          TST     (R0)            ;;IS THERE ANOTHER NUMBER?
012134  001770                          BEQ     6$              ;;BR IF NO
012136  104401  012144                  TYPE    ,8$             ;;TYPE TWO(2) SPACES
012142  000771                          BR      7$              ;;LOOP
012144  020040      000           8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
012150                                  .EVEN
```

```
2279
2280                                    .SBTTL  POWER DOWN AND UP ROUTINES
2281                                    ;;*****************************************************************
2282                                            ;*POWER DOWN ROUTINE
2283                                    ;;*****************************************************************
2284  012150  012737  012310  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;SET FOR FAST UP
2285  012156  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;PRIO:7
2286  012164  010046                          MOV     R0,-(SP)        ;PUSH R0 ON STACK
2287  012166  010146                          MOV     R1,-(SP)        ;PUSH R1 ON STACK
2288  012170  010246                          MOV     R2,-(SP)        ;PUSH R2 ON STACK
2289  012172  010346                          MOV     R3,-(SP)        ;PUSH R3 ON STACK
2290  012174  010446                          MOV     R4,-(SP)        ;PUSH R4 ON STACK
2291  012176  010546                          MOV     R5,-(SP)        ;PUSH R5 ON STACK
2292  012200  017746  166634                  MOV     @SWR,-(SP)      ;PUSH @SWR ON STACK
2293  012204  010667  000104                  MOV     SP,$SAVR6       ;SAVE SP
2294  012210  012737  012222  000024          MOV     #$PWRUP,@#PWRVEC ;SET UP VECTOR
2295  012216  000000                          HALT
2296  012220  000776                          BR      .-2             ;HANG UP
2297
2298
2299                                    ;;*****************************************************************
2300                                            ;*POWER UP ROUTINE
2301                                    ;;*****************************************************************
2302  012222  012737  012310  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;SET FOR FAST DOWN
2303  012230  016706  000060                  MOV     $SAVR6,SP       ;GET SP
2304  012234  012677  166600                  MOV     (SP)+,@SWR      ;POP STACK INTO @SWR
2305  012240  012605                          MOV     (SP)+,R5
2306  012242  012604                          MOV     (SP)+,R4        ;POP STACK INTO R4
2307  012244  012603                          MOV     (SP)+,R3        ;POP STACK INTO R3
2308  012246  012602                          MOV     (SP)+,R2        ;POP STACK INTO R2
2309  012250  012601                          MOV     (SP)+,R1        ;POP STACK INTO R1
2310  012252  012600                          MOV     (SP)+,R0        ;POP STACK INTO R0
2311  012254  012737  012150  000024          MOV     #$PWRDN,@#PWRVEC ;SET UP THE POWER DOWN VECTOR
2312  012262  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;PRIO:7
2313  012270  005067  000020                  CLR     $SAVR6          ;WAIT LOOP FOR THE TTY
2314  012274  005267  000014          1$:     INC     $SAVR6          ;WAIT FOR THE INC
2315  012300  001375                          BNE     1$              ;OF WORD
2316  012302  104401                          TYPE                    ;REPORT THE POWER FAILURE
2317  012304  012316          $PWRMG: .WORD   $POWER                  ;POWER FAIL MESSAGE POINTER
2318  012306  000002                          RTI
2319  012310  000000          $ILLUP: HALT                            ;THE POWER UP SEQUENCE WAS STARTED
2320  012312  000776                          BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
2321  012314  000000          $SAVR6: 0                               ;PUT THE SP HERE
2322  012316  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
2323  012324  000122
2324
```

```
                                        .SBTTL  SCOPE HANDLER ROUTINE

                                        ;;**********************************************************************
                                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                        ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW14=1         LOOP ON TEST
                                        ;*SW09=1         LOOP ON ERROR
                                        ;*CALL
                                        ;*      SCOPE           ;;SCOPE=IOT

012326                                  $SCOPE:
012326  104406                                 CKSWR                           ;;TEST  FOR CHANGE IN SOFT-SWR
012330  032777  040000  166502          1$:    BIT     #BIT14,@SWR             ;;LOOP ON PRESENT TEST?
012336  001052                                 BNE     $OVER                   ;;YES IF SW14=1
                                        ;*****START OF CODE FOR THE XOR TESTER*****
012340  000416                          $XTSTR: BR      6$                      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
012342  013746  000004                         MOV     @#ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
012346  012737  012366  000004                 MOV     #5$,@#ERRVEC            ;;SET FOR TIMEOUT
012354  005737  177060                         TST     @#177060                ;;TIME OUT ON XOR?
012360  012637  000004                         MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
012364  000421                                 BR      $SVLAD                  ;;GO TO THE NEXT TEST
012366  022626                          5$:    CMP     (SP)+,(SP)+             ;;CLEAR THE STACK AFTER A TIME OUT
012370  012637  000004                         MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
012374  000407                                 BR      7$                      ;;LOOP ON THE PRESENT TEST
012376                                  6$:;*****END OF CODE FOR THE XOR TESTER*****
012376  105767  166401                  2$:    TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
012402  001412                                 BEQ     $SVLAD                  ;;BR IF NO
012404  032777  001000  166426                 BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
012412  001404                                 BEQ     4$                      ;;BR IF NO
012414  016767  166370  166364          7$:    MOV     $LPERR,$LPADR           ;;SET LOOP ADDRESS TO LAST SCOPE
012422  000420                                 BR      $OVER
012424  105067  166353                  4$:    CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
012430  105267  166346                  $SVLAD: INCB    $TSTNM                  ;;COUNT TEST NUMBERS
012434  116767  166342  166430                 MOVB    $TSTNM,$TESTN           ;;SET TEST NUMBER IN APT MAILBOX
012442  011567  166340                         MOV     (SP),$LPADR             ;;SAVE SCOPE LOOP ADDRESS
012446  011667  166336                         MOV     (SP),$LPERR             ;;SAVE ERROR LOOP ADDRESS
012452  005067  166402                         CLR     $ESCAPE                 ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
012456  112767  000001  166331                 MOVB    #1,$ERMAX               ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
012464  016777  166312  166350          $OVER: MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER
012472  016716  166310                         MOV     $LPADR,(SP)             ;;FUDGE RETURN ADDRESS
012476  000002                                 RTI                             ;;FIXES PS
```

```
                                        ;;*****************************************************************
                                        .SBTTL  APT COMMUNICATIONS ROUTINE
                                        ;;*****************************************************************

012500  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;TO REPORT FATAL ERROR
012506  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;TO TYPE A MESSAGE
012514  000403                  BR      $ATYC
012516  112767  000001  000220  $ATY4:  MOVB    #1 $FFLG        ;TO ONLY REPORT FATAL ERROR
012524                          $ATYC:
012524  010046                  MOV     R0,-(SP)        ;PUSH R0 ON STACK
012526  010146                  MOV     R1,-(SP)        ;PUSH R1 ON STACK
012530  105767  000206          TSTB    $MFLG           ;SHOULD TYPE A MESSAGE?
012534  001450                  BEQ     5$              ;IF NOT: BR
012536  122767  000001  166342  CMPB    #APTENV,$ENV    ;OPERATING UNDER APT?
012544  001031                  BNE     3$              ;IF NOT: BR
012546  132767  000100  166333  BITB    #APTSPOOL,$ENVM ;SHOULD SPOOL MESSAGE?
012554  001425                  BEQ     3$              ;IF NOT: BR
012556  017600  000004          MOV     @4(SP),R0       ;GET MESSAGE ADDRESS
012562  062766  000002  000004  ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
012570  005767  166272          1$:     TST     $MSGTYPE                ;SEE IF DONE W/ LAST XMISSION?
012574  001375                  BNE     1$              ;IF NOT: WAIT
012576  010067  166300          MOV     R0,$MSGAD       ;PUT ADDRESS IN MAILBOX
012602  105720                  2$:     TSTB    (R0)+           ;FIND END OF MESSAGE
012604  001376                  BNE     2$
012606  166700  166270          SUB     $MSGAD,R0       ;SUB START OF MESSAGE
012612  006200                  ASR     R0              ;GET MESSAGE LENGTH IN WORDS
012614  010067  166264          MOV     R0,$MSGLGT      ;PUT LENGTH IN MAILBOX
012620  012767  000004  166240  MOV     #4,$MSGTYPE     ;TELL APT TO TAKE MESSAGE
012626  000413                  BR      5$
012630  017667  000004  000016  3$:     MOV     @4(SP),4$       ;PUT MSG ADDR IN JSR LINKAGE
012636  062766  000002  000004  ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
012644  016746  155126          MOV     177776,-(SP)    ;PUSH 177776 ON STACK
012650  004767  000072          JSR     PC,$TYPE        ;CALL TYPE MACRO
012654  000000                  4$:     .WORD   0
012656                          5$:
012656  105767  000062          10$:    TSTB    $FFLG           ;SHOULD REPORT FATAL ERROR?
012662  001413                  BEQ     12$             ;IF NOT: BR
012664  005767  166216          TST     $ENV            ;RUNNING UNDER APT?
012670  001410                  BEQ     12$             ;IF NOT: BR
012672  005767  166170          11$:    TST     $MSGTYPE        ;FINISHED LAST MESSAGE?
012676  001375                  BNE     11$             ;IF NOT: WAIT
012700  017667  000004  166162  MOV     @4(SP),$FATAL   ;GET ERROR #
012706  005267  166154          INC     $MSGTYPE        ;TELL APT TO TAKE ERROR
012712  062766  000002  000004  12$:    ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
012720  105067  000020          CLRB    $FFLG           ;CLEAR FATAL FLAG
012724  105067  000013          CLRB    $LFLG           ;CLEAR LOG FLAG
012730  105167  000006          CLRB    $MFLG           ;CLEAR MESSAGE FLAG
012734  012601                  MOV     (SP)+,R1        ;POP STACK INTO R1
012736  012600                  MOV     (SP)+,R0        ;POP STACK INTO R1
012740  000207                  RTS     PC              ;RETURN
012742     000                  $MFLG:  .BYTE   0
012743     000                  $LFLG:  .BYTE   0               ;LOG FLAG
012744     000                  $FFLG:  .BYTE   0               ;FATAL FLAG

012746                          .EVEN
```

```
2427            000200                  APTSIZE=200
2428            000001                  APTENV=001
2429            000100                  APTSPOOL=100
2430            000040                  APTCSUP=040
2431
```

```
                                        .SBTTL  TYPE ROUTINE

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

012746  105767  166105          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
012752  100002                          BPL     1$              ;;BR IF YES
012754  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
012756  000430                          BR      3$              ;;LEAVE
012760  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
012762  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
012766  122767  000001  166112          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
012774  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
012776  132767  000100  166103          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
013004  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
013006  010067  000004                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
013012  004767  177470                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
013016  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
013020  132767  000040  166061  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
013026  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
013030  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
013032  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
013034  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
013036  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
013040  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
013044  000002                          RTI                     ;;RETURN
013046  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
013052  001430                          BEQ     8$
013054  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
013060  001006                          BNE     5$
013062  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
013064  104401                          TYPE                    ;;TYPE A CR AND LF
013066  001063                          $CRLF
013070  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
013074  000755                          BR      2$              ;;GET NEXT CHARACTER
013076  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
013102  126726  165750          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
013106  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
013110  016746  165740                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
013114  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
013120  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
013122  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
```

```
013126  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
013132  000770                          BR      7$              ;;LOOP


                                ;HORIZONTAL TAB PROCESSOR

013134  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
013140  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
013144  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
013152  001372                          BNE     9$              ;;TAB STOP
013154  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
013156  000724                          BR      2$              ;;GET NEXT CHARACTER
013160  105777  165664          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
013164  100375                          BPL     $TYPEC
013166  116677  000002  165656          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
013174  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
013202  001003                          BNE     1$              ;;BRANCH IF NO
013204  105067  000014                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
013210  000406                          BR      $TYPEX          ;;EXIT
013212  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
013220  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
013222  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
013224  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
013226  000207          $TYPEX: RTS     PC


                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

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

013230  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
013234  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
013242  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
013246  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
013252  000406                          BR      $TYPON
013254  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
013262  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
```

```
2544  013270  112767  000005  000154  $TYPUN: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
2545  013276  010346                          MOV     R3,-(SP)        ;;SAVE R3
2546  013300  010446                          MOV     R4,-(SP)        ;;SAVE R4
2547  013302  010546                          MOV     R5,-(SP)        ;;SAVE R5
2548  013304  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
2549  013310  005404                          NEG     R4
2550  013312  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
2551  013316  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
2552  013322  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
2553  013326  011605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
2554  013332  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
2555  013334  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
2556  013336  103404                          BR      3$              ;;GO DO MSB
2557  013340  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
2560  013342  006105                          ROL     R5
2561  013344  006105                          ROL     R5
2562  013346  010503                          MOV     R5,R3
2563  013350  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
2564  013352  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
2565  013356  100016                          BPL     7$              ;;BR IF NO
2566  013360  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
2567  013364  001002                          BNE     4$              ;;TEST FOR 0
2570  013366  005704                          TST     R4              ;;SUPPRESS THIS 0?
2571  013370  001403                          BEQ     5$              ;;BR IF YES
2572  013372  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
2573  013374  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
2574  013400  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
2575  013404  110367  000040                  MOVB    R3,B$           ;;SAVE FOR TYPING
2576  013410  104401  013450                  TYPE    ,B$             ;;GO TYPE THIS DIGIT
2577  013414  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
2600  013420  003347                          BGT     2$              ;;BR IF MORE TO DO
2601  013422  002402                          BLT     6$              ;;BR IF DONE
2602  013424  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
2603  013426  000744                          BR      2$              ;;GO DO THE LAST DIGIT
2604  013430  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
2605  013432  012604                          MOV     (SP)+,R4        ;;RESTORE R4
2606  013434  012603                          MOV     (SP)+,R3        ;;RESTORE R3
2607  013436  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
2610  013444  012616                          MOV     (SP)+,(SP)
2611  013446  000002                          RTI                     ;;RETURN
2612  013450  000000                  B$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
2613  013451  000000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
2614  013452  000000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
2615  013453  000000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
2616  013454  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
2617
```

```
                                        .SBTTL  TTY INPUT ROUTINE
                                        ;;*****************************************************************
                                        .ENABL  LSB
                                        ;;*****************************************************************
                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                        ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;*WHEN OPERATING IN TTY FLAG MODE.
013456  022767  000176  165354  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
013464  001074                          BNE     15$             ;;BRANCH IF NO
013466  105777  165352                  TSTB    @$TKS           ;;CHAR THERE?
013472  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
013474  117746  165346                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
013500  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
013504  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
013510  001062                          BNE     15$             ;;NO, RETURN TO USER
013512  126727  165316  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
013520  001456                          BEQ     15$             ;;BRANCH IF YES
                                        
013522  104401  014203                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
013526  104401  014210          $CTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
013532  016746  164440                  MOV     $SWREG,-(SP)    ;;SAVE SWREG FOR TYPEOUT
013536  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
013540  104401  014221                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
013544  005046                  19$:     CLR     -(SP)           ;;CLEAR COUNTER
013546  005046                          CLR     -(SP)           ;;THE NEW SWR
013550  105777  165270          7$:     TSTB    @$TKS           ;;CHAR THERE?
013554  100375                          BPL     7$              ;;IF NOT TRY AGAIN
                                        
013556  117746  165264                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
013562  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
                                        
                                        
                                        
013566  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
013572  001005                          BNE     10$             ;;BRANCH IF NOT
013574  104401  014176                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
013600  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
013604  000757                          BR      19$             ;;LET'S TRY IT AGAIN
                                        
013606  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
013612  001022                          BNE     16$             ;;BRANCH IF NO
013614  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
013620  001403                          BEQ     11$             ;;BRANCH IF YES
013622  016677  000002  165210          MOV     2(SP),@SWR      ;;SAVE NEW SWR
013630  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
013634  104401  001063          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
013640  126727  165171  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
013646  001003                          BNE     15$             ;;BRANCH IF NOT
013650  012777  000100  165166          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
013656  000002                  15$:    RTI                     ;;RETURN
013660  004767  177274          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
```

```
2646  013674  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
2647  013700  002420                          BLT     18$             ;;BRANCH IF YES
2648  013702  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
2649  013706  003015                          BGT     18$             ;;BRANCH IF YES
2650  013710  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
2651  013714  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
2652  013720  001403                          BEQ     17$             ;;BRANCH IF YES
2653  013722  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
2654  013724  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
2655  013726  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
2656  013730  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
2657  013734  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
2658  013740  000707                          BR      7$              ;;GET THE NEXT ONE
2659  013742  104401  001062          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
2660  013746  000720                          BR      20$             ;;SIMULATE CONTROL-U
2661                                .DSABL  LSB
2662
2663
2664                                ;;*****************************************************************
2665                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
2666                                ;*CALL:
2667                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
2668                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
2669                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
2670                                ;
2671
2672  013750  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
2673  013752  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
2674  013760  105777  165070          1$:     TSTB    @$TKS           ;;WAIT FOR
2675  013764  100375                          BPL     1$              ;;A CHARACTER
2676  013766  117766  165064  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
2677  013774  042766  177600  000004          BIC     #^C(177),4(SP)  ;;GET RID OF JUNK IF ANY
2678  014002  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
2679  014010  001013                          BNE     3$              ;;BRANCH IF NO
2680  014012  105777  165036          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
2681  014016  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
2682  014020  117746  165032                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
2683  014024  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
2684  014030  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
2685  014034  001366                          BNE     2$              ;;IF NOT DISCARD IT
2686  014036  000750                          BR      1$              ;;YES, RESUME
2687  014040  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
2688  014046  002407                          BLT     4$              ;;BRANCH IF YES
2689  014050  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
2690  014056  003003                          BGT     4$              ;;BRANCH IF YES
2691  014060  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
2692  014066  000002          4$:     RTI                     ;;GO BACK TO USER
2693                                ;;*****************************************************************
2694                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
2695                                ;*CALL:
2696                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
2697                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
2698                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
2699
2700  014070  010346          $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
2701  014072  012703  014166  1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
```

```
2702  014066  022703  014176          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
2703  014072  101405                  BLOS    4$              ;;BR IF YES
2704  014074  104407                  RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
2705  014076  112613                  MOVB    (SP)+,(R3)      ;;GET CHARACTER
2706  014100  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
2707  014104  001003                  BNE     3$              ;;SKIP IF NOT
2708  014106  104401  001062          4$:     TYPE    $QUES           ;;TYPE A '?'
2709  014112  000763                  BR      1$              ;;CLEAR THE BUFFER AND LOOP
2710  014114  111367  000044          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
2711  014120  104401  014164                  TYPE    ,9$
2712  014124  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
2713  014130  001356                  BNE     2$              ;;LOOP IF NOT RETURN
2714  014132  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
2715  014136  104401  001064                  TYPE    ,$LF            ;;TYPE A LINE FEED
2716  014142  012603                  MOV     (SP)+,R3        ;;RESTORE R3
2717  014144  011646                  MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
2718  014146  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
2719  014154  012766  014166  000004          MOV     #$TTYIN,4(SP)
2720  014162  000002                  RTI                     ;;RETURN
2721  014164     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
2722  014165     000                          .BYTE   0               ;;TERMINATOR
2723  014166  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
2724  014176  002536  005015     000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
2725  014203     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
2726  014210  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
2727  014216  020075     000
2728  014221     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
2729  014226  036440  000040
2730
```

```
                                        .SBTTL  TRAP DECODER

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

014232  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
014234  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
014240  005740                          TST     -(R0)           ;;BACKUP BY 2
014242  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
014244  006300                          ASL     R0              ;;POSITION FOR INDEXING
014246  016000  014266                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
014252  000200                          RTS     R0              ;;GO TO ROUTINE


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

014254  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
014256  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
014264  000002                          RTI                     ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

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

                                        ;       ROUTINE
                                        ;       -------

014266  014254                  $TRPAD: .WORD   $TRAP2
014270  012746                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
014272  013254                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
014274  013230                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
014276  013270                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)

014300  013526                          $GTSWR  ;;CALL=GTSWR    TRAP+5(104405)  GET SOFT-SWR SETTING

014302  013456                          $CKSWR  ;;CALL=CKSWR    TRAP+6(104406)  TEST FOR CHANGE IN SOFT-SWR
014304  013740                          $RDCHR  ;;CALL=RDCHR    TRAP+7(104407)  TTY TYPEIN CHARACTER ROUTINE
014306  014060                          $RDLIN  ;;CALL=RDLIN    TRAP+10(104410) TTY TYPEIN STRING ROUTINE
```

```
                                        .SBTTL  ECHO TEST
                                ;;**************************************************************
                                ;*THIS ROUTINE WILL ECHO ANY CHARACTER TYPED
                                ;*AT THE CONSOLE
                                ;*THE TEST IS HALTED BY TYPING A CONTROL-C
                                ;*TEST CAN BE RESTARTED AFTER HALTING BY JUST CONTINUING
                                ;;**************************************************************
014310                          ECHO:
014310  000005                          RESET                   ;CLEAR EVERYTHING
014312  112777  000052  166014          MOVB    #'*,@CTBUF      ;TRANSMIT PROMPT "*"
014320  105777  166002          2$:     TSTB    @CRCSR          ;WAIT FOR INPUT
014324  100375                          BPL     2$
014326  117777  165776  166000          MOVB    @CRBUF,@CTBUF   ;ECHO INPUT
014334  017700  165770                  MOV     @CKBUF,R0       ;STORE INPUT
014340  100023                          BPL     5$              ;BR IF "ERROR"NOT SET
014342  052701  010000                  BIS     #BIT12,R1       ;SET PARITY ERROR TEST MASK
014346  030100                          BIT     R1,R0           ;CHECK  FOR PARITY ERROR FLAG
014350  001403                          BEQ     3$              ;BR IF NOT SET
014352  004767  000056                  JSR     PC,MSG          ;REPORT PARITY ERROR
014356  014462                          MPAR
014360  006301                  3$:     ASL     R1              ;SHIFT MASK TO TEST "FR" FLAG
014362  030100                          BIT     R1,R0           ;TEST FOR FRAMING ERROR FLAG
014364  001403                          BEQ     4$              ;BR IF NOT SET
014366  004767  000042                  JSR     PC,MSG          ;REPSORT FRAMING ERROR
014372  014473                          MFR
014374  006301                  4$:     ASL     R1              ;SHIFT MASK TO TEST "OR" FLAG
014376  030100                          BIT     R1,R0           ;TEST FOR OVERFLOW ERROR
014400  001403                          BEQ     5$              ;BR IF NOT SET
014402  004767  000026                  JSR     PC,MSG          ;REPORT OVERFLOW ERROR
014406  014505                          MOR
014410  042700  000200          5$:     BIC     #BIT7,R0        ;CLEAR ANY PARITY BIT
014414  022700  000003                  CMP     #3,R0           ;WAS INPUT CONTROL-C
014420  001337                          BNE     2$              ;BR IF NOT
014422  004767  000006                  JSR     PC,MSG          ;REPORT PROGRAM STOP
014426  014520                          MSTOP
014430  000000                          HALT                    ;END OF TEST HALT
014432  000726                          BR      ECHO            ;AFTER END OF TEST HALT
                                                                ;   PRESS CONTINUE TO RESTART ECHO TEST

014434  013600                  MSG:    MOV     @(SP)+,R0       ;PICK UP MESSAGE POINTER
014436  062746  000002                  ADD     #2,-(SP)        ;ADJUST RETURN PC
014442  105777  165664          WAIT:   TSTB    @CTCSR          ;WAIT FOR XMIT DONE
014446  100375                          BPL     WAIT
014450  112077  165660                  MOVB    (R0)+,@CTBUF    ;SEND CHARACTER
014454  105710                          TSTB    (R0)            ;IS THIS END OF MESSAGE?
014456  001371                          BNE     WAIT            ;BR IF NOT
014460  000207                          RTS     PC              ;RETURN

014462  005015  040520  044522  MPAR:   .ASCIZ  <CR><LF>/PARITY/
014470  054524     000
014473     015  043012  040522  MFR:    .ASCIZ  <CR><LF>/FRAMING/
014500  044515  043516     000
014505     015  047412  042526  MOR:    .ASCIZ  <CR><LF>/OVERFLOW/
014512  043122  047514  000127
014520  005015  052123  050117  MSTOP:  .ASCIZ  <CR><LF>/STOP/
014526     000
```

```
 2831
 2832
 2833            01:530                          .EVEN
 2834
 2835                                    .SBTTL  TERMINAL OUTPUT TEST
 2836
 2837                                    ;;*****************************************************************
 2838                                            ;*THIS ROUTINE WILL OUTPUT ALL WRITABLE CHARACTERS FOR THE
 2839                                            ;*THE OCTAL CODE 040 --> 377
 2840                                            ;*32 CHARACTERS ARE PRINTED ON EACH LINE
 2841                                            ;*THE PATTERN IS REPEATED EVERY THREE LINES
 2842                                            ;*
 2843                                    ;;*****************************************************************
 2844
 2845  014530  012701  000040           OUTTST: MOV     #40,R1          ;LOAD FIRST WRITABLE CHARACTER
 2846  014534  012700  000040           1$:     MOV     #40,R0          ;LOAD CHAR COUNT PER LINE
 2847  014540  105777  165566           2$:     TSTB    @#XCTCSR        ;WAIT FOR DONE
 2848  014544  100375                          BPL     2$
 2849  014546  010177  165562                  MOV     R1,@#XCTBUF     ;TRANSMIT A CHARACTER
 2850  014552  105201                          INCB    R1              ;INCREMENT CHARACTER CODE
 2851  014554  005300                          DEC     R0              ;DECREMENT CHAR COUNT
 2852  014556  001370                          BNE     2$              ;BR IF LINE NOT COMPLETE
 2853  014560  004767  177650                  JSR     PC,MSG          ;SSUE CR,LINE FEED
 2854
 2855  014564  001063                          $CRLF
 2856  014566  105777  165534                  TSTB    @#RCRCSR        ;ANY CHARACTER RECEIVED?
 2857  014572  100404                          BMI     3$              ;BR IF YES
 2858  014574  032701  000200                  BIT     #BIT7,R1        ;FINISHED ONE PASS OF WRITABLE CHARACTERS?
 2859  014600  001353                          BNE     OUTTST          ;BR IF YES
 2860  014602  000754                          BR      1$              ;IF NOT WRITE NEXT LINE
 2861
 2862  014604  005077  165520           3$:     CLR     @#RCRBUF        ;CLEAR RECEIVER
 2863  014610  000000                          HALT                    ;STOP TEST
 2864  014612  000746                          BR      OUTTST          ;RESTART TEST IF CONTINUED
```

```
 2865
 2866
 2867  014614  040503  020116  047516  EM1:    .ASCIZ  /CAN NOT ACCESS TCSR/
 2868  014622  020124  041501  042503
 2869  014630  051523  052040  051503
 2870  014636  000122
 2871  014640  040503  020116  047516  EM2:    .ASCIZ  /CAN NOT ACCESS TBUF/
 2872  014646  020124  041501  042503
 2873  014654  051523  052040  052502
 2874  014662  000106
 2875  014664  041524  051123  042040  EM3:    .ASCIZ  /TCSR DONE NOT CLEARED WITH TBUF FULL/
 2876  014672  047117  020105  047516
 2877  014700  020124  046103  040505
 2878  014706  042522  020104  044527
 2879  014714  044124  052040  052502
 2880  014722  020106  052506  046114
 2881  014730     000
 2882  014731     124  051503  020122  EM4:    .ASCIZ  /TCSR DONE NOT SET/
 2883  014736  047504  042516  047040
 2884  014744  052117  051440  052105
 2885  014752     000
 2886  014753     124  051503  020122  EM5:    .ASCIZ  /TCSR DONE NOT SET WITH RESET/
 2887  014760  047504  042516  047040
 2888  014766  052117  051440  052105
 2889  014774  053440  052111  020110
 2890  015002  042522  042523  000124
 2891  015010  040503  020116  047516  EM6:    .ASCIZ  /CAN NOT ACCESS RCSR/
 2892  015016  020124  041501  042503
 2893  015024  051523  051040  051503
 2894  015032  000122
 2895  015034  040503  020116  047516  EM7:    .ASCIZ  /CAN NOT ACCESS RBUF/
 2896  015042  020124  041501  042503
 2897  015050  051523  051040  052502
 2898  015056  000106
 2899  015060  040503  020116  047516  EM10:   .ASCIZ  /CAN NOT ACCESS LKS/
 2900  015066  020124  041501  042503
 2901  015074  051523  046040  051513
 2902  015102     000
 2903  015103     102  052111  020060  EM11:   .ASCIZ  /BIT0 OF TCSR NOT CLEAR AFTER RESET/
 2904  015110  043117  052040  051503
 2905  015116  020122  047516  020124
 2906  015124  046103  040505  020122
 2907  015132  043101  042524  020122
 2908  015140  042522  042523  000124
 2909  015146  040503  020116  047516  EM12:   .ASCIZ  /CAN NOT SET BIT0 OF TCSR/
 2910  015154  020124  042523  020124
 2911  015162  044502  030124  047440
 2912  015170  020106  041524  051123
 2913  015176     000
 2914  015177     103  047101  047040  EM13:   .ASCIZ  /CAN NOT CLEAR BIT0 OF TCSR/
 2915  015204  052117  041440  042514
 2916  015212  051101  041040  052111
 2917  015220  020060  043117  052040
 2918  015226  051503  000122
 2919  015232  042522  042523  020124  EM14:   .ASCIZ  /RESET DID NOT CLEAR BIT0 OF TCSR/
 2920  015240  044504  020104  047516
```

```
2921  015246  020124  046103  040505
2922  015254  020122  044502  030124
2923  015262  047440  020106  041524
2924  015270  051123     000
2925  015273     102  052111  020062  EM15:   .ASCIZ  /BIT2 OF TCSR NOT CLEAR AFTER RESET/
2926  015300  043117  052040  051503
2927  015306  020122  047516  020124
2928  015314  046103  040505  020122
2929  015322  043101  042524  020122
2930  015330  042522  042523  000124
2931  015336  040503  020116  047516  EM16:   .ASCIZ  /CAN NOT SET BIT2 OF TCSR/
2932  015344  020124  042523  020124
2933  015352  044502  031124  047440
2934  015360  020106  041524  051123
2935  015366     000
2936  015367     103  047101  047040  EM17:   .ASCIZ  /CAN NOT CLEAR BIT2 OF TCSR/
2937  015374  052117  041440  042514
2938  015402  051101  041040  052111
2939  015410  020062  043117  052040
2940  015416  051503  000122
2941  015422  042522  042523  020124  EM20:   .ASCIZ  /RESET DID NOT CLEAR BIT2 OF TCSR/
2942  015430  044504  020104  047516
2943  015436  020124  046103  040505
2944  015444  020122  044502  031124
2945  015452  047440  020106  041524
2946  015460  051123     000
2947  015463     102  052111  020066  EM21:   .ASCIZ  ;BIT6 OF TCSR NOT CLEAR AFTER RESET/
2948  015470  043117  052040  051503
2949  015476  020122  047516  020124
2950  015504  046103  040505  020122
2951  015512  043101  042524  020122
2952  015520  042522  042523  027524
2953  015526     000
2954  015527     130  044515  020124  EM22:   .ASCIZ  /XMIT INTERRUPT AT PRIORITY 7/
2955  015534  047111  042524  051122
2956  015542  050125  020124  052101
2957  015550  050040  044522  051117
2958  015556  052111  020131  000067
2959  015564  040503  020116  047516  EM23:   .ASCIZ  /CAN NOT SET BIT6 OF TCSR/
2960  015572  020124  042523  020124
2961  015600  044502  033124  047440
2962  015606  020106  041524  051123
2963  015614     000
2964  015615     103  047101  047040  EM24:   .ASCIZ  /CAN NOT CLEAR BIT6 OF TCSR/
2965  015622  052117  041440  042514
2966  015630  051101  041040  052111
2967  015636  020066  043117  052040
2968  015644  051503  000122
2969  015650  042522  042523  020124  EM25:   .ASCIZ  /RESET DID NOT CLEAR BIT6 OF TCSR/
2970  015656  044504  020104  047516
2971  015664  020124  046103  040505
2972  015672  020122  044502  033124
2973  015700  047440  020106  041524
2974  015706  051123     000
2975  015711     102  052111  020066  EM26:   .ASCIZ  /BIT6 OF RCSR NOT CLEAR AFTER RESET/
2976  015716  043117  051040  051503
```

```
 2977  015724  020122  047516  020124
 2978  015732  046103  040505  020122
 2979  015740  043101  042524  020122
 2980  015746  042523  042523  000124
 2981  015754  041522  051126  044440  EM27:   .ASCIZ  /RCVR INTERRUPT WITH PRIORITY 7/
 2982  015762  052116  051105  052522
 2983  015770  052120  053440  052111
 2984  015776  020110  051120  047511
 2985  016004  044522  054524  033440
 2986  016012     000
 2987  016013     103  047101  047040  EM30:   .ASCIZ  /CAN NOT SET BIT6 OF RCSR/
 2988  016020  052117  051440  052105
 2989  016026  041040  052111  020066
 2990  016034  043117  051040  051503
 2991  016042  000122
 2992  016044  040503  020116  047516  EM31:   .ASCIZ  /CAN NOT CLEAR BIT6 OF RCSR/
 2993  016052  020124  046103  040505
 2994  016060  020122  044502  033124
 2995  016066  047440  020106  041522
 2996  016074  051123     000
 2997  016077     103  047101  047040  EM32:   .ASCIZ  /CAN NOT CLEAR BIT6 OF RCSR WITH RESET/
 2998  016104  052117  051440  042514
 2999  016112  051103  041040  052111
 3000  016120  020066  043117  051040
 3001  016126  051503  020122  044527
 3002  016134  044124  051040  051505
 3003  016142  052105     000
 3004  016145     102  052111  020066  EM33:   .ASCIZ  /BIT6 OF LKS NOT CLEAR AFTER RESET/
 3005  016152  043117  046040  051513
 3006  016160  047040  052117  041440
 3007  016166  042514  051101  040440
 3008  016174  052106  051105  051040
 3009  016202  051505  052105     000
 3010  016207     114  051513  044440  EM34:   .ASCIZ  /LKS INTERRUPT WITH PRIORITY 7/
 3011  016214  052116  051105  052522
 3012  016222  052120  053440  052111
 3013  016230  020110  051120  047511
 3014  016236  044522  054524  033440
 3015  016244     000
 3016  016245     103  047101  047040  EM35:   .ASCIZ  /CAN NOT SET BIT6 OF LKS/
 3017  016252  052117  051440  052105
 3018  016260  041040  052111  020066
 3019  016266  043117  046040  051513
 3020  016274     000
 3021  016275     103  047101  047040  EM36:   .ASCIZ  /CAN NOT CLEAR BIT6 OF LKS/
 3022  016302  052117  041440  042514
 3023  016310  051101  041040  052111
 3024  016316  020066  043117  046040
 3025  016324  051513     000
 3026  016327     122  051505  052105  EM37:   .ASCIZ  /RESET DID NOT CLEAR BIT6 OF LKS/
 3027  016334  042040  042111  047040
 3028  016342  052117  041440  042514
 3029  016350  051101  041040  052111
 3030  016356  020066  043117  046040
 3031  016364  051513     000
 3032  016367     104  040525  020114  EM40:   .ASCIZ  /DUAL ADDRESSING ERROR/
```

```
 3033  016374  042101  051104  051505
 3034  016402  044523  043516  042440
 3035  016410  051122  051117     000
 3036  016415     102  052111  020066  EM41:   .ASCIZ  /BIT6 OF LKS NOT SET AFTER RESET/
 3037  016422  043117  046040  051513
 3038  016430  047040  052117  051440
 3039  016436  052105  040440  052106
 3040  016444  051105  051040  051505
 3041  016452  052105     000
 3042  016455     103  047101  047040  EM42:   .ASCIZ  /CAN NOT CLEAR BIT7 OF LKS/
 3043  016462  052117  041440  042514
 3044  016470  051101  041040  052111
 3045  016476  020067  043117  046040
 3046  016504  051513     000
 3047  016507     102  052111  020067  EM43:   .ASCIZ  /BIT7 OF LKS DOES NOT SET/
 3048  016514  043117  046040  051513
 3049  016522  042040  042517  020123
 3050  016530  047516  020124  042523
 3051  016536  000124
 3052  016540  052122  020103  047111  EM44:   .ASCIZ  /RTC INTERRUPT AT PRIORITY 7/
 3053  016546  042524  051122  050125
 3054  016554  020124  052101  050040
 3055  016562  044522  051117  052111
 3056  016570  020131  000067
 3057  016574  052122  020103  047111  EM45:   .ASCIZ  /RTC INTERRUPTS WHEN DISABLED/
 3058  016602  042524  051122  050125
 3059  016610  051524  053440  042510
 3060  016616  020116  044504  040523
 3061  016624  046102  042105     000
 3062  016631                          EM47:
 3063  016631     122  041524  044440  EM46:   .ASCIZ  /RTC INTERRUPT DID NOT OCCUR/
 3064  016636  052116  051105  052522
 3065  016644  052120  042040  042111
 3066  016652  047040  052117  047440
 3067  016660  041503  051125     000
 3068  016665     122  041524  042040  EM50:   .ASCIZ  /RTC DOUBLE INTERRUPT/
 3069  016672  052517  046102  020105
 3070  016700  047111  042524  051122
 3071  016706  050125  000124
 3072  016712  042522  042523  020124  EM51:   .ASCIZ  /RESET DID NOT INTERRUPT/
 3073  016720  044504  020104  047516
 3074  016726  020124  047111  042524
 3075  016734  051122  050125  000124
 3076  016742  052122  020103  047111  EM52:   .ASCIZ  /RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS/
 3077  016750  042524  051122  050125
 3078  016756  020124  044504  020104
 3079  016764  047516  020124  046103
 3080  016772  040505  020122  044527
 3081  017000  044124  041040  052111
 3082  017006  020067  043117  046040
 3083  017014  051513     000
 3084  017017     103  047514  045503  EM53:   .ASCIZ  /CLOCK REPEATABILITY/
 3085  017024  051040  050105  040505
 3086  017032  040524  044502  044514
 3087  017040  054524     000
 3088  017043     130  044515  020124  EM54:   .ASCIZ  /XMIT INTERRUPTS WHEN DISABLED/
```

D07


```
 3089  017050  047111  042524  051122
 3090  017056  050125  051524  053440
 3091  017064  042510  020116  044504
 3092  017072  040523  046102  042105
 3093  017100     000
 3094  017101     130  044515  020124  EM56:   .PSCIZ  /XMIT INTERRUPTS AT PRIORITY 7/
 3095  017106  047111  042524  051122
 3096  017114  050125  051524  040440
 3097  017122  020124  051120  047511
 3098  017130  044522  054524  033440
 3099  017136     000
 3100  017137     130  044515  020124  EM57:   .ASCIZ  /XMIT INTERRUPTS WITH ENABLE CLEAR/
 3101  017144  047111  042524  051122
 3102  017152  050125  051524  053440
 3103  017160  052111  020110  047105
 3104  017166  041101  042514  041440
 3105  017174  042514  051101     000
 3106  017201                          EM55:
 3107  017201     130  044515  020124  EM60:   .ASCIZ  /XMIT DID NOT INTERRUPT/
 3108  017206  044504  020104  047516
 3109  017214  020124  047111  042524
 3110  017222  051122  050125  000124
 3111  017230  046530  052111  051040  EM61:   .ASCIZ  /XMIT RE-INTERRUPTED/
 3112  017236  026505  047111  042524
 3113  017244  051122  050125  042524
 3114  017252  000104
 3115  017254  047514  042101  047111  EM62:   .ASCIZ  /LOADING TBUF DID NOT CLEAR INTERRUPT/
 3116  017262  020107  041124  043125
 3117  017270  042040  042111  047040
 3118  017276  052117  041440  042514
 3119  017304  051101  044440  052116
 3120  017312  051105  052522  052120
 3121  017320     000
 3122  017321     122  053103  020122  EM63:   .ASCIZ  /RCVR ACTIVE NOT SET/
 3123  017326  041501  044524  042526
 3124  017334  047040  052117  051440
 3125  017342  052105     000
 3126  017345     122  053103  020122  EM64:   .ASCIZ  /RCVR DONE NEVER SET/
 3127  017352  047504  042516  047040
 3128  017360  053105  051105  051440
 3129  017366  052105     000
 3130  017371     122  053103  020122  EM65:   .ASCIZ  /RCVR ACTIVE NOT CLEARED WITH DONE SET/
 3131  017376  041501  044524  042526
 3132  017404  047040  052117  041440
 3133  017412  042514  051101  042105
 3134  017420  053440  052111  020110
 3135  017426  047504  042516  051440
 3136  017434  052105     000
 3137  017437     122  051505  052105  EM66:   .ASCIZ  /RESET DID NOT CLEAR RCVR DONE/
 3138  017444  042040  042111  047040
 3139  017452  052117  041440  042514
 3140  017460  051101  051040  053103
 3141  017466  020122  047504  042516
 3142  017474     000
 3143  017475     122  051104  042440  EM67:   .ASCIZ  /RDR ENABLE DID NOT CLEAR RCVR DONE/
 3144  017502  040516  046102  020105
```

E07


```
3145  017510  044504  020104  047516
3146  017516  020124  046103  040505
3147  017524  020122  041522  051126
3148  017532  042040  047117  000105
3149  017540  042522  042101  047111  EM70:   .ASCIZ  /READING RBUF DID NOT CLEAR RCVR DONE/
3150  017546  020107  041122  043125
3151  017554  042040  042111  047040
3152  017562  052117  041440  042514
3153  017570  051101  051040  053103
3154  017576  020122  047504  042516
3155  017604     000
3156  017605     122  053103  020122  EM71:   .ASCIZ  /RCVR INTERRUPTS WITH ENABLE CLEAR/
3157  017612  047111  042524  051122
3158  017620  050125  051524  053440
3159  017626  052111  020110  047105
3160  017634  041101  042514  041440
3161  017642  042514  051101     000
3162  017647     122  053103  020122  EM73:   .ASCIZ  /RCVR INTERRUPTS AT PRIORITY 7/
3163  017654  047111  042524  051122
3164  017662  050125  051524  040440
3165  017670  020124  051120  047511
3166  017676  044522  054524  033440
3167  017704     000
3168  017705     122  053103  020122  EM74:   .ASCIZ  /RCVR INT RQST PASSED WITH ENABLE CLEAR/
3169  017712  047111  020124  050522
3170  017720  052123  060040  051501
3171  017726  042523  020104  044527
3172  017734  044124  042440  040516
3173  017742  046102  020105  046103
3174  017750  040505  000122
3175  017754                          EM72:
3176  017754  041522  051126  042040  EM75:   .ASCIZ  /RCVR DID NOT INTERRUPT/
3177  017762  042111  047040  052117
3178  017770  044440  052116  051105
3179  017776  052522  052120     000
3180  020003     122  053103  020122  EM76:   .ASCIZ  /RCVR RE-INTERRUPTED/
3181  020010  042522  044455  052116
3182  020016  051105  052522  052120
3183  020024  042105     000
3184  020027     122  040505  044504  EM77:   .ASCIZ  /READING RBUF DID NOT CLEAR INTERRUPT/
3185  020034  043516  051040  052502
3186  020042  020106  044504  020104
3187  020050  047516  020124  046103
3188  020056  040505  020122  047111
3189  020064  042524  051122  050125
3190  020072  000124
3191  020074  042522  042523  020124  EM100:  .ASCIZ  /RESET DID NOT CLEAR RCVR INTERRUPT/
3192  020102  044504  020104  047516
3193  020110  020124  046103  040505
3194  020116  020122  041522  051126
3195  020124  044440  052116  051105
3196  020132  052522  052120     000
3197  020137     047  051117  020047  EM101:  .ASCIZ  /'OR' FLAG DID NOT SET/
3198  020144  046106  043501  042040
3199  020152  042111  047040  052117
3200  020160  051440  052105     000
```

F07


```
  3201  020165     047  051105  047522  EM102:  .ASCIZ  /'ERROR' NOT SET WITH 'OR' FLAG/
  3202  020172  023522  047040  052117
  3203  020200  051440  052105  053440
  3204  020206  052111  020110  047447
  3205  020214  023522  043040  040514
  3206  020222  000107
  3207  020224  051102  040505  020113  EM103:  .ASCIZ  /BREAK DID NOT XMIT ALL ZEROES/
  3208  020232  044504  020104  047516
  3209  020240  020124  046530  052111
  3210  020246  040440  046114  055040
  3211  020254  051105  042517  000123
  3212  020262  051102  040505  020113  EM104:  .ASCIZ  /BREAK DID NOT SET 'FR' ERROR/
  3213  020270  044504  020104  047516
  3214  020276  020124  042523  020124
  3215  020304  043047  023522  042440
  3216  020312  051122  051117     000
  3217  020317     104  052101  020101  EM105:  .ASCIZ  /DATA COMPARE ERROR/
  3218  020324  047503  050115  051101
  3219  020332  020105  051105  047522
  3220  020340  000122
  3221  020342  040504  040524  041440  EM106:  .ASCIZ  /DATA COMPARE ERROR WITH CABLE/
  3222  020350  046517  040520  042522
  3223  020356  042440  051122  051117
  3224  020364  053440  052111  020110
  3225  020372  040503  046102  000105
  3226  020400  044524  042515  052517  EM107:  .ASCIZ  /TIMEOUT IN EXERCISER TEST/
  3227  020406  020124  047111  042440
  3228  020414  042530  041522  051511
  3229  020422  051105  052040  051505
  3230  020430  000124
  3231  020432  047111  047503  051122  EM110:  .ASCIZ  /INCORRECT RECEIVE COUNT/
  3232  020440  041505  020124  042522
  3233  020446  042503  053111  020105
  3234  020454  047503  047125  000124
  3235  020462  040504  040524  041440  EM111:  .ASCIZ  /DATA COMPARE ERROR IN EXERCISER/
  3236  020470  046517  040520  042522
  3237  020476  042440  051122  051117
  3238  020504  044440  020116  054105
  3239  020512  051105  044503  042523
  3240  020520  000122
  3241  020522  051124  050101  041440  EM112:  .ASCIZ  /TRAP CATCHER/
  3242  020530  052101  044103  051105
  3243  020536     000
  3244  020537     116  020117  046103  EM113:  .ASCIZ  /NO CLK INTERRUPT IN EXERCISER/
  3245  020544  020113  047111  042524
  3246  020552  051122  050125  020124
  3247  020560  047111  042440  042530
  3248  020566  041522  051511  051105
  3249  020574     000
  3250
  3251  020575     124  051505  021524  DH1:    .ASCIZ  /TEST#   ERROR#  TCSR/
  3252  020602  020040  042440  051122
  3253  020610  051117  020043  052040
  3254  020616  051503  000122
  3255  020622  042524  052123  020043  DH2:    .ASCIZ  /TEST#   ERR PC  TBUF/
  3256  020630  020040  051105  020122
```

G07


```
3257  020632  041520  020040  041124
3258  020644  043125     000
3259  020647     124  051505  021524  DH6:    .ASCIZ  /TEST#  ERR PC  RCSR/
3260  020654  020040  042440  051122
3261  020662  050040  020103  051040
3262  020670  051503  000122
3263  020674  042524  052123  020043  DH7:    .ASCIZ  /TEST#  ERR PC  RBUF/
3264  020702  020040  051105  020122
3265  020710  041520  020040  041122
3266  020716  043125     000
3267  020721     124  051505  021524  DH10:   .ASCIZ  /TEST#  ERR PC  LKS/
3268  020726  020040  042440  051122
3269  020734  050040  020103  046040
3270  020742  051513     000
3271  020745     124  051505  021524  DH40:   .ASCIZ  /TEST#  ERR PC  GOOD    BAD/
3272  020752  020040  042440  051122
3273  020760  050040  020103  043440
3274  020766  047517  020104  020040
3275  020774  041040  042101     000
3276  021001     124  051505  021524  DH53:   .ASCIZ  /TEST#  ERR PC  LKS     CNT1    CNT2/
3277  021006  020040  042440  051122
3278  021014  050040  020103  046040
3279  021022  051513  020040  020040
3280  021030  041440  052116  020061
3281  021036  020040  041440  052116
3282  021044  000062
3283  021046  042524  052123  020043  DH103:  .ASCIZ  /TEST#  ERR PC  RCSR    DATA/
3284  021054  020040  051105  020122
3285  021062  041520  020040  041522
3286  021070  051123  020040  020040
3287  021076  040504  040524     000
3288  021103     124  051505  021524  DH105:  .ASCIZ  /TEST#  ERR PC  RCSR    GOOD    BAD/
3289  021110  020040  042440  051122
3290  021116  050040  020103  051040
3291  021124  051503  020122  020040
3292  021132  043440  047517  020104
3293  021140  020040  041040  042101
3294  021146     000
3295  021147     124  051505  021524  DH110:  .ASCIZ  /TEST#  ERR PC  RCSR    TRANS   RCV/
3296  021154  020040  042440  051122
3297  021162  050040  020103  051040
3298  021170  051503  020122  020040
3299  021176  052040  040522  051516
3300  021204  020040  051040  053103
3301  021212     000
3302  021213     124  051505  021524  DH112:  .ASCIZ  /TEST#  ERR PC  RCSR    OLDPC   TRAP ADR/
3303  021220  020040  042440  051122
3304  021226  050040  020103  051040
3305  021234  051503  020122  020040
3306  021242  047440  042114  041520
3307  021250  020040  052040  040522
3308  021256  020120  042101  000122
3309                          .EVEN
3310  021264  005015  040515  047111  M1:     .ASCIZ  <CR><LF>/MAINDEC-DZDLD-A/
3311  021272  042504  026503  055104
3312  021300  046104  026504  000101
```

```
                                                        .EVEN
021306  005015                          M2:             .ASCII  <CR><LF>
021310  020040  042040  053105          M2A:            .ASCIZ  /   DEVICES UNDER TEST  /
021316  041511  051505  052440
021324  042116  051105  052040
021332  051505  020124  000040
                                                        .EVEN
021340  001072  001016  002312  DT1:    .WORD   STESTN,SERRPC,TCSR,0
021346  000000
021350  001072  001016  002314  DT2:    .WORD   STESTN,SERRPC,TBUF,0
021356  000000
021360  001072  001016  002306  DT6:    .WORD   STESTN,SERRPC,RCSR,0
021366  000000
021370  001072  001016  002310  DT7:    .WORD   STESTN,SERRPC,RBUF,0
021376  000000
021400  001072  001016  002346  DT10:   .WORD   STESTN,SERRPC,LKS,0
021406  000000
021410  001072  001016  002306  DT40:   .WORD   STESTN,SERRPC,RCSR,$BOADR,0
021416  001022  000000
021422  001072  001016  002346  DT53:   .WORD   STESTN,SERRPC,LKS,FIRST,SECND,0
021430  006350  006352  000000
021436  001072  001016  002306  DT103:  .WORD   STESTN,SERRPC,RCSR,$BDOAT,0
021444  001026  000000
021450  001072  001016  002306  DT105:  .WORD   STESTN,SERRPC,RCSR,$GDOAT,$BDOAT,0
021456  001024  001026  000000
021464  001072  001016  002306  DT110:  .WORD   STESTN,SERRPC,RCSR,XMTCNT,RCVCNT,0
021472  011274  011272  000000
021500  001072  001016  002306  DT112:  .WORD   STESTN,SERRPC,RCSR,OLDPC,BOVECT,0
021506  011626  011630  000000
        000001                          .END
```

```
ABASE = 176500            122#    234     275
ACDW1 = 000000            234
ACDW2 = 000000            234
ACPUOP= 000000            234     249
ADDW0 = 000000            234
ADDW1 = 000000            234
ADDW10= 000000            234
ADDW11= 000000            234
ADDW12= 000000            234
ADDW13= 000000            234
ADDW14= 000000            234
ADDW15= 000000            234
ADDW2 = 000000            234
ADDW3 = 000000            234
ADDW4 = 000000            234
ADDW5 = 000000            234
ADDW6 = 000000            234
ADDW7 = 000000            234
ADDW8 = 000000            234
ADDW9 = 000000            234
ADEVCT= 000000            234     240
ADEVM = 000000            234     276
ADR     003466            871#    874
ADRTBL  002354            703#    709     869
AENV  = 000000            234     245
AENVM = 000000            234     246
AFATAL= 000000            234     237
AMADR1= 000000            234     262
AMADR2= 000000            234     266
AMADR3= 000000            234     269
AMADR4= 000000            234     272
AMAMS1= 000000            234     256
AMAMS2= 000000            234     264
AMAMS3= 000000            234     267
AMAMS4= 000000            234     270
AMSGAD= 000000            234     242
AMSGLG= 000000            234     243
AMSGTY= 000000            234     236
AMTYP1= 000000            234     257
AMTYP2= 000000            234     265
AMTYP3= 000000            234     268
AMTYP4= 000000            234     271
APASS = 000000            234     239
APRIOR= 000000            234
APTCSU= 000040           2430#   2463
APTENV= 000001           2206    2385    2428#   2456
APTSIZ= 000200            768    2427#
APTSPO= 000100           2387    2429#   2458
APTSZD  003172            775     777     802#
ASWREG= 000000            234     247
ATESTN= 000000            234     238
AUNIT = 000000            234     241
AUSWR = 000000            234     248
AVECT1= 000300            123#    234     273
AVECT2= 000000            234     274
BOVECT  011630           2172*   2178#   3341
```

```
BEGIN   003330          844#  2161
BIT0  = 000001          107#   772   846  1052  1063  1064  1069  1070  1081  1083  1085  1679  1910
                       1917  1922  1926  1939  1946
BIT00 = 000001           97#   107
BIT01 = 000002           96#   106
BIT02 = 000004           95#   105
BIT03 = 000010           94#   104
BIT04 = 000020           93#   103
BIT05 = 000040           92#   102
BIT06 = 000100           91#   101
BIT07 = 000200           90#   100
BIT08 = 000400           89#    99
BIT09 = 001000           88#    98  2217  2356
BIT1  = 000002          106#   844  1275
BIT10 = 002000           87#  1874  1933
BIT11 = 004000           86#  1635  1652  1654
BIT12 = 010000           85#  2790
BIT13 = 020000           84#  1947  2201
BIT14 = 040000           83#  1884  2340
BIT15 = 100000           82#  1890
BIT2  = 000004          105#  1094  1107  1108  1113  1114  1126  1128  1131  1632  1639  1648  1662
                       1666  1675  1682  1693  1698  1711  1720  1722  1730  1735  1745  1757  1767
                       1770  1782  1791  1822  1826  1852  1877  1886  1892  1895  1905  1917  1922
                       1926  1938  1946  1961  1973  1976  2028
BIT3  = 000010          104#  2073
BIT4  = 000020          103#   839
BIT5  = 000040          102#   776  2087
BIT6  = 000100          101#  1031  1145  1155  1156  1162  1163  1168  1170  1187  1197  1198  1204
                       1205  1211  1213  1227  1234  1244  1245  1251  1252  1256  1258  1280  1281
                       1285  1304  1332  1339  1362  1371  1384  1392  1411  1421  1427  1435  1450
                       1456  1464  1480  1519  1533  1538  1547  1554  1561  1578  1585  1593  1605
                       1608  1618  1709  1710  1728  1737  1754  1762  1792  1801  1821  1831  1851
                       1860  2022  2027  2029  2030  2041  2060  2089
BIT7  = 000200          100#   774  1311  1312  1340  1363  1393  1428  1457  1460  1984  2805  2858
BIT8  = 000400           99#  1049  1902  1935
BIT9  = 001000           98#
BPT   = 000003          126#
BPTVEC= 000014          114#
BUF     011300         2037  2068  2106#
CATCH   011602          137  2170#
CKSUM = 104406         2216  2339  2771#
CLK     011264         2025  2100#
CLKCNT  011276         2062  2100*  2105#
CMPARE  006320         1497  1500#
COMP    011152         2070#  2074
CONT    005404         1318#  1321
CONT41  010370         1913  1920#
CR    = 000015           22#  2151  2502  2512  2823  2825  2827  2829  3310  3314
CRBUF   002330          690#  2787  2788  2862*
CRCSR   002326          689#   794   850  2785  2856
CRLF  = 000200           23#  2473  2512
CRPSW   002340          694#
CRVECT  002336          693#
CTBUF   002334          692#  2784*  2787*  2818*  2849*
CTCSR   002332          691#  2816  2847
CTPSW   002344          696#
```

```
CTSTFL  002276   672#  724*  849*  896  918  942  956 1029 1054 1072 1097 1117 1225
                1302  1330  1382 1419 1448 1478 2020 2058 2115*
CTVECT  002342   695#
DOISP = 177570    29#  217   756
DEVDOR  002454   709#  779   810
DH1     020575   297   307   312  317  337  342  347  352  357  362  367  372  377
                 382   387   392  397  512  517  522  527  532  537  542 3251#
DH10    020721   332   427   432  437  442  447  457  462  467  472  477  482  487
                 492   497   502  668 3267#
DH103   021046   628  3283#
DH105   021103   638   643   658 3288#
DH110   021147   653  3295#
DH112   021213   663  3302#
DH2     020622   302  3255#
DH40    020745   452  3271#
DH53    021001   507  3276#
DH6     020647   322   402   407  412  417  422  547  552  557  562  567  572  577
                 582   587   592  597  602  607  612  618  623  633  648 3259#
DH7     020674   327  3263#
DISPLA  001042   217#  756*  764* 2196* 2367*
DISPRE  000174   147#  764
DSWR  = 177570    28#  216   755
DT1     021340   298   308   313  318  338  343  348  353  358  363  368  373  378
                 383   388   393  398  513  518  523  528  533  538  543 3321#
DT10    021400   333   428   433  438  443  448  458  463  468  473  478  483  488
                 493   498   503  669 3329#
DT103   021436   629  3335#
DT105   021450   639   644   659 3337#
DT110   021464   654  3339#
DT112   021500   664  3341#
DT2     021350   303  3323#
DT40    021410   453  3331#
DT53    021422   508  3333#
DT6     021360   323   403   408  413  418  423  548  553  558  563  568  573  578
                 583   588   593  598  603  608  613  619  624  634  649 3325#
DT7     021370   328  3327#
DVRDT   003220   808   810#
ECHO    014310   152  2782#  2811
EMTVEC= 000030   117#  740*  741*
EM1     014614   296  2867#
EM10    015060   331  2899#
EM100   020074   611  3191#
EM101   020137   617  3197#
EM102   020165   622  3201#
EM103   020224   627  3207#
EM104   020262   632  3212#
EM105   020317   637  3217#
EM106   020342   642  3221#
EM107   020400   647  3226#
EM11    015103   336  2903#
EM110   020432   652  3231#
EM111   020462   657  3236#
EM112   020522   662  3241#
EM113   020537   667  3244#
EM12    015146   341  2909#
EM13    015177   346  2914#
```

EM14	015232	351	2919#
EM15	015273	356	2925#
EM16	015336	361	2931#
EM17	015367	366	2936#
EM2	014640	301	2871#
EM20	015422	371	2941#
EM21	015453	376	2947#
EM22	015527	381	2954#
EM23	015564	386	2959#
EM24	015615	391	2964#
EM25	015650	396	2969#
EM26	015711	401	2975#
EM27	015754	406	2981#
EM3	014664	306	2875#
EM30	016013	411	2987#
EM31	016044	416	2992#
EM32	016077	421	2997#
EM33	016145	426	3004#
EM34	016207	431	3010#
EM35	016245	436	3016#
EM36	016275	441	3021#
EM37	016327	446	3026#
EM4	014731	311	2882#
EM40	016367	451	3032#
EM41	016415	456	3036#
EM42	016455	461	3042#
EM43	016507	466	3047#
EM44	016540	471	3052#
EM45	016574	476	3057#
EM46	016631	481	3063#
EM47	016631	486	3062#
EM5	014753	316	2886#
EM50	016665	491	3068#
EM51	016712	496	3072#
EM52	016742	501	3076#
EM53	017017	506	3084#
EM54	017043	511	3088#
EM55	017201	516	3106#
EM56	017101	521	3094#
EM57	017137	526	3100#
EM6	015010	321	2891#
EM60	017201	531	3107#
EM61	017230	536	3111#
EM62	017254	541	3115#
EM63	017321	546	3122#
EM64	017345	551	3126#
EM65	017371	556	3130#
EM66	017437	561	3137#
EM67	017475	566	3143#
EM7	015034	326	2895#
EM70	017540	571	3149#
EM71	017605	576	3156#
EM72	017754	581	3175#
EM73	017647	586	3162#
EM74	017705	591	3168#
EM75	017754	596	3176#

```
EM76    020003            C01   3180#
EM77    020027            606   3184#
ENDEV   011400           2083   2111#
ENDMG   011520           2138   2151#
ERRVEC= 000004            110#   753    754*   765*  2345   2346*  2348*  2351*
FIRST   006350           1498*  1500   1511#  3333
GNS     = ****** U       2764   2765   2766   2767  2769   2771   2772   2773
GOAGIN  011534           2148   2154#
GTSWR   = 104405         2769#
HT      = 000011            20#  2471   2512
IOTVEC= 000020            115#   738*   739*
LF      = 000012             21#  2151   2506   2512  2823   2825   2827   2829   3310   3314
LKS     002346            699#  1035   1234   1244* 1245   1251*  1252   1256*  1258   1308   1311*  1312   1318
                         1339*  1340*  1341   1349* 1353   1362*  1363*  1364   1371*  1392*  1393*  1394   1411*
                         1427*  1428*  1429   1435* 1456*  1457*  1458   1460*  1464*  1486*  1487   1489*  1490
                         1493*  2027*  3329   3333
MAIN    003036            773   776#
MFR     014473           2799   2825#
MOR     014505           2804   2827#
MPAR    014462           2794   2823#
MSG     014434           2793   2798   2803   2808  2814#  2853
MSTOP   014520           2809   2829#
M1      021264            974   3310#
M2      021306            976   3314#
M2A     021310            841*  3315#
OLDPC   011626           2173*  2177#  3341
OUTTST  014530            153   2845#  2859   2864
PC      =%000007           41#   717*   779*   810*  901*   923*   947*   961*   1059*  1077*  1103*  1122*  1143*
                         1185*  1232*  1334*  1351* 1390*  1402*  1423*  1432*  1452*  1461*  1525*  1548*  1563*
                         1581*  1606*  1615*  1715* 1746*  1764*  1786*  1816*  1828*  1846*  1857*  2033*  2132*
                         2135*  2142*  2147   2203* 2209*  2268*  2404*  2421*  2461*  2480*  2487*  2494*  2508*
                         2510*  2645*  2793*  2798* 2803*  2808*  2821*  2853*
PIRQ    = 177772            27#
PIRQVE= 000240            121#
PR0     = 000000             44#
PR1     = 000040             45#
PR2     = 000100             46#
PR3     = 000140             47#
PR4     = 000200             48#
PR5     = 000240             49#
PR6     = 000300             50#
PR7     = 000340             51#
PS      = 177776             24#    25
PSW     = 177776             25#
PWRVEC= 000024            116#   744*   745*  2284*  2285*  2294*  2302*  2311*  2312*
RBUF    002310            679#  1016   1697   1827*  1884   1890   1909*  1920   1947   1965   1997   2095   3327
RCSR    002306            678#   851    870   1002   1187   1197*  1198   1204*  1205   1211*  1213   1276   1278
                         1281   1635   1644   1652   1654*  1659   1666*  1677   1679*  1680   1695   1699   1710*
                         1718   1728*  1737*  1752   1754*  1762*  1789   1792*  1801*  1821*  1824   1831*  1851*
                         1854   1860*  1907   1912   1963   1990   2030*  3325   3331   3335   3337   3339   3341
RCV     011250           2018   2095#
RCVCNT  011272           2032*  2047   2096*  2103#  3339
RDCHR   = 104407         2704   2772#
RDLIN   = 104410         2773#
RESVEC= 000010            111#
RPSW    002320            683#  1714*  1750*  1785*  1820*  1850*  2019*
```

```
RSTRT   011560          2156   2160#
RTCPSW  002352           701#  1338*  1389*  2026*
RTCVT   002350           700#  1230   1231*  1263*  1336   1337*  1350*  1361*  1372*  1387   1388*  1401*  1412*
                        1425   1426*  1441*  1454   1455*  1471*  2024   2025*
RVECT   002316           682#  1183   1184*  1218*  1712   1713*  1727*  1736*  1748   1749*  1763*  1774*  1783
                        1784*  1799*  1802*  1808*  1818   1819*  1837*  1848   1849*  1667*  2015   2018*  2082*
R0      =%000000           32#   710*   711    713    714*   715    784*   785*   786    791*   792    794*   796
                         815*   816*   817    819    820*   821    826*   828*   829*   830*   831*   833*   835*
                         850*   852    869*   871    872*   873    876*   877    878*   879    950*   953*   1275*
                        1277   1287*  1288   1317*  1320*  1483*  1494*  1634*  1637*  1643*  1646*  1697*  1878*
                        1882*  1911*  1914*  1989*  1992*  2036*  2039*  2068*  2070   2077   2092*  2097*  2139*
                        2142   2170*  2171*  2172   2240   2241*  2242*  2249*  2250*  2251*  2252*  2253*  2254
                        2259   2264*  2266*  2270   2272   2286   2310*  2381   2389*  2393   2394   2396*  2397*
                        2398   2420*  2454   2455*  2460   2465   2468*  2741   2742*  2743   2744*  2745*  2746*
                        2747*  2788*  2791   2796   2801   2805*  2806   2814*  2818   2819   2846*  2851*
R1      =%000001           33#   711*   712*   715    817*   818*   821    827*   832*   834*   836*   837*   838*
                         839    841    851*   852*   853    870*   871*   877*   1484*  1496*  1500*  1501*  1503*
                        1504   1960*  1962   1966   1968*  1971   1986*  1988   1998   2000*  2003   2035*  2069*
                        2070   2072*  2073   2078   2086*  2087   2091   2287   2309*  2382   2419*  2790*  2791
                        2795*  2796   2800*  2801   2845*  2849   2850*  2858
R2      =%000002           34#   709*   713*   803*   804    807*   814*   819*   867*   869    876    1485*  1492*
                        1498   1501   1507   1965*  1966   1972   1997*  1998   2004   2037*  2095*  2288   2308*
R3      =%000003           35#   782*   799    890*   904    912*   926    1000*  1007   1014*  1021   1033*  1041
                        1141*  1175   1183*  1218   1230*  1263   1270*  1294   1336*  1372   1387*  1412   1425*
                        1441   1454*  1471   1522*  1540   1550*  1571   1579*  1599   1609*  1624   1712*  1736
                        1748*  1774   1783*  1802   1808   1818*  1837   1848*  1867   2014*  2081   2289   2307*
                        2545   2554*  2560*  2561*  2564*  2569*  2570*  2571   2580*  2700   2701*  2702   2705*
                        2706   2710   2712   2714*  2716*
R4      =%000004           36#   1271*  1295   2015*  2082   2290   2306*  2546   2548*  2549*  2550*  2551   2552*
                        2566   2568*  2576*  2579*
R5      =%000005           37#   2024*  2291   2305*  2547   2553*  2555*  2557*  2558*  2559*  2560   2578*
R6      =%000006           38#    40    732*   733*   734
R7      =%000007           39#    41
SECND   006352          1507*  1512#  3333
SETADR  003434           859    865#
SHIFT   003274           831#   840
SIZE    003046           779#
SP      =%000006           40#   736*   753*   761*   765    789*   895    917    1005   1019   1038   1151   1193
                        1240   1283   1286*  1292*  1346   1358   1370   1400   1407   1438   1468   1529   1539
                        1558   1568   1590   1596   1621   1723   1731   1758   1771   1793   1805   1834   1864
                        2157*  2163*  2164*  2165*  2170   2173   2198   2219*  2222*  2240*  2245*  2266   2270*
                        2286*  2287*  2288*  2289*  2290*  2291*  2292*  2293   2303*  2304   2305   2306   2307
                        2308   2309   2310   2345*  2348   2350   2351   2363   2364   2368*  2381*  2382*  2383
                        2390*  2401   2402*  2403*  2413   2415*  2419   2420   2454*  2455   2465*  2467   2468
                        2469*  2471   2473   2475   2481   2483*  2485*  2493*  2497   2501   2502   2506   2537*
                        2538   2539   2540*  2545*  2546*  2547*  2553   2578   2579   2580   2581*  2582*  2605*
                        2606*  2607   2614*  2617*  2618*  2622*  2623*  2627   2630*  2634   2636   2638   2639*
                        2646   2648   2650*  2651   2653*  2654*  2655*  2656*  2657*  2672*  2673*  2676*  2677*
                        2678   2682*  2683*  2684   2687   2689   2691*  2700*  2705   2716   2717*  2718*  2719*
                        2741*  2742   2752*  2753*  2814   2815*
STACK = 001100           15#
START   002506          151    721#
STKLMT= 177774           26#
SWR     001040          216#   734    755*   757    763*   770*   776    1031   1049   1227   1304   1332   1384
                        1421   1450   1480   1874   1902   1933   1935   1984   2022   2060   2201   2213   2217
                        2292   2304*  2340   2356   2601   2638*
```

```
SWREG   000176            148#    763    2601    2614
S%0   = 000001             79#
S%00  = 000001             69#     79
S%01  = 000002             68#     78
S%02  = 000004             67#     77
S%03  = 000010             66#     76
S%04  = 000020             65#     75
S%05  = 000040             64#     74
S%06  = 000100             63#     73
S%07  = 000200             62#     72
S%08  = 000400             61#     71
S%09  = 001000             60#     70
SW1   = 000002             78#
SW10  = 002000             59#
SW11  = 004000             58#
SW12  = 010000             57#
SW13  = 020000             56#
SW14  = 040000             55#
SW15  = 100000             54#
SW2   = 000004             77#
SW3   = 000010             76#
SW4   = 000020             75#
SW5   = 000040             74#
SW6   = 000100             73#
SW7   = 000200             72#
SW8   = 000400             71#
SW9   = 001000             70#
TBITVE= 000014            112#
TBUF    002314            681#    914     933#    938#    983#    986#   1611#   1614#   1633#   1676#   1694#   1717#   1751#
                         1788#   1823#   1853#   1879#   1906#   1940#   1943#   1962#   1988#   2038#   2091#   3323
TCONS   003354             847     849#
TCSR    002312             680#    892      934     939     951     984     989    1052   1063#   1064   1069#   1070   1081#
                          1083   1085#    1094   1107#   1108   1113#   1114   1126#   1128   1131#   1145   1155#   1156
                          1162#   1163    1168#   1170    1519#   1520    1533#   1538#   1547#   1552   1554#   1561#   1578#
                          1583   1585#    1593#   1605#   1608#   1612    1618#   1632#   1639#   1648#   1662#   1675#   1682#
                          1693#   1698#   1709#   1711#   1720#   1722#   1730#   1735#   1745#   1757#   1767#   1770#   1782#
                          1791#   1822#   1826#   1852#   1877#   1880    1886#   1892#   1895#   1905#   1910#   1917#   1922#
                          1926#   1938#   1939#   1941    1944    1946#   1961#   1973#   1976#   2028#   2029#   2041    2089#
                          3321
TKVEC = 000060             119#
TMP1    002300             673#    845#    861#    867     868#
TMP2    002302             674#    780#    788#    798#    802#    806#    826    2113    2155
TMP3    002304             675#    844#    858     860#    866#
TOLER   006332            1502    1504#
TPSW    002324             685#    853    1524#   2017#
TPVEC = 000064             120#
TRAPVE= 000034             118#    742#    743#
TRTVEC= 000014             113#
TSTDEV  003404             848     858#    863    2116
TSTDVM  003202             804#    809
TST1    003526             855     881     889#   2158
TST10   004204            1030    1032    1048#
TST11   004362            1050    1084    1092#
TST12   004530            1129    1139#
TST13   004662            1181#
TST14   005014            1224#
```

```
TST15   005164          1226    1228    1269#
TST16   005326          1301#
TST17   005420          1303    1305    1319    1329#
TST2    003600           911#
TST20   005626          1331    1333    1381#
TST21   005764          1383    1385    1418#
TST22   006100          1420    1422    1447#
TST23   006220          1449    1451    1477#
TST24   006354          1479    1481    1505    1510    1518#
TST25   006460          1546#
TST26   006572          1577#
TST27   006674          1604#
TST3    003652           932#
TST30   007000          1630#
TST31   007140          1667    1673#
TST32   007210          1681    1691#
TST33   007256          1700    1708#
TST34   007432          1743#
TST35   007600          1780#
TST36   007726          1814#
TST37   010046          1844#
TST4    004014           964     968     970     972     982#
TST40   010160          1873#
TST41   010272          1875    1901#
TST42   010414          1903    1932#
TST43   010520          1934    1936    1949    1957#
TST44   010612          1983#
TST45   010702          1985    2001    2012#
TST5    004050           990     999#
TST6    004102          1013#
TST7    004134          1028#
TVECT   002322           684#   1141    1142*   1175*   1522    1523*   1532*   1540*   1550    1551*   1562*   1571*   1579
                        1580#   1591*   1599*   1609    1610*   1624*   2014    2016*   2081#
TYPE  = 104401           973     975    2138    2204    2239    2256    2258    2261    2263    2267    2274    2316    2476
                        2572    2612    2613    2616    2629    2640    2659    2708    2711    2715    2764#
TYPOC = 104402          2247    2271    2615    2765#
TYPON = 104404          2767#
TYPOS = 104403          2766#
VCTADR  003224           800     814#
VCTTBL  002414           704#    814     876
VECT    003506           877#    880
WAITV   007020          1635#   1638
WAIT    014442          2816#   2817    2820
WDONE   007046          1644#   1647
WAPSW   011570          1143    1185    1232    1334    1351    1390    1402    1423    1432    1452    1461    1525    1548
                        1563    1581    1606    1615    1715    1746    1764    1786    1816    1828    1846    1857    2033
                        2163#
XCONT   011240          2088    2091#
XMIT    011214          2016    2085#
XMITCNT 011274          2031*   2047    2085*   2104#   3339
XRET    011244          2090    2092#
$APTHD  000500           175     181#
$ASTAT= ****** U         901     923     947     961    1059    1077    1103    1122
$ATYC   012524          2378    2380#
$ATY1   012500          2376#
$ATY3   012506          2377#   2461
```

```
$ATYP   012516            901    923    947    961   1059   1077   1103   1122   2209   2379#
$AUTOB  001034            213#  2609   2730
$BASE   001142            275#   710    784    792
$BGNADR 001022            209#  1276*  1277*  1278   1280*  1285*  3331
$BOOAT  001026            210#  1912*  1972*  2004*  2077*  3335   3337
$CHARC  013224           2478*  2488*  2495   2504*  2509#
$CKSUM  013456           2601#  2771
$CNTRG  001000            196#   731    732    740    746    747
$CNTB = 000000            226#
$CNTLG  014203           2612   2725#
$CNTLU  014176           2629   2724#
$CPUOP  001114            249#
$CRLF   001063            228#  2204   2239   2258   2263   2267   2477   2512   2640   2724   2855
$DEVCT  001076            240#   725*   971   2112*  2113   2154*
$DEVM   001144            276#   781*   787*   790*   797*   803    846    858
$DOAGN  011510           2134   2140   2146#
$ENDAD  011500            162   2142#  2224
$ENDCT  011460            746   2136#
$ENULL  011514           2149#
$ENV    001106            245#   772   2206   2385   2409   2456
$ENVM   001107            246#   768    774   2387   2458   2463
$EOP    011430           2114   2127#
$EOPCT  011452            746*  2132#  2137
$ERFLG  001003            199#  2124*  2331   2354   2360*  2370
$ERMAX  001015            205#   748*  2366*  2370
$ERPAR  011632            740   2193#
$ERRPC  001116            206#  2198*  2199*  2200   2245   3321   3323   3325   3327   3329   3331   3333   3335
                         3337   3339   3341
$ERRTB  001146            293#  2253
$ERRTY  012014           2203   2238#
$ERTTL  001012            203#  2197*
$ESCAP  001060            226#   747*  2220   2222   2365*
$ETABL  001106            244#
$ETEND  001146            187    277#
$FATAL  001070            237#   721*  2413*
$FFLG   012744           2376*  2379*  2407   2416*  2424#
$FILLC  001056            224#  2481   2512
$FILLS  001055            223#  2512
$GOADR  001020            207#
$GOOAT  001024            209#  1971*  2003*  2078*  3337
$GET42  011470           2139#
$GTSUM  013526           2613#  2769
$HIBTS  000500            182#
$ICNT   001004            200#
$ILLUC  012310           2284   2302   2319#
$INTRG  001035            214#  2641   2730
$ITEMB  001014            204#  2200*  2208   2242
$LF     001064            229#  2512   2715   2724
$LFLG   012743           2417*  2423#
$LPADR  001006            201#   749*  2358*  2363*  2368   2370
$LPERR  001010            202#   750*  2219   2358   2364*  2370
$MADR1  001120            262#
$MADR2  001124            266#
$MADR3  001130            269#
$MADR4  001134            272#
$MAIL   001066            183    187    235#   767   2362   2456
```

```
$MAMS1  001116            256#
$MAMS2  001122            264#
$MAMS3  001126            267#
$MAMS4  001132            270#
$MBADR  000502            183#
$MFLG   012742           2377*   2383    2419*   2422#
$MNEW   014221           2616    2728#
$MSGAD  001102            242#   2393*   2396
$MSGLG  001104            243#   2398*
$MSGTY  001066            236#    722*   2391    2399*   2411    2414*
$MSWR   014210           2613    2726#
$MTYP1  001117            257#
$MTYP2  001123            265#
$MTYP3  001127            268#
$MTYP4  001133            271#
$NULL   001054            222#   2483    2512
$NWTST= 000001            886#    908#    929#    979#    996#   1010#   1025#   1045#   1089#   1136#   1178#   1221#   1266#
                         1298#   1326#   1378#   1415#   1444#   1474#   1515#   1543#   1574#   1601#   1627#   1670#   1688#
                         1705#   1740#   1777#   1811#   1841#   1870#   1898#   1929#   1954#   1980#   2009#
$OCNT   013452           2544#   2573#   2586#
$OMODE  013454           2539#   2543#   2548    2551*   2562*   2588#
$OVER   012464           2341    2359    2367#
$PASS   001074            239#    767#    969    2130*   2131*   2149
$PASTM  000506            185#
$POWER  012316           2317    2322#
$PWRON  012150            744    2284#   2311
$PWRMG  012304           2317#
$PWRUP  012222           2294    2302#
$QUES   001062            227#   2512    2659    2708    2724
$RDCHR  013740           2672#   2772
$RDDEC= ****** U         2774
$RDLIN  014060           2700#   2773
$RDOCT= ****** U         2774
$RDSZ = 000010           2693#
$RTNAD  011512           2148#
$R2A  = ****** U         2774
$SAVRE= ****** U         2774
$SAVR6  012314           2293*   2303    2313*   2314*   2321#
$SCOPE  012326            738    2338#
$SETUP= 000137            144#    737     738     740     742     744     746     747     749    2129    2339    2596    2730
$STUP = 177777            144#
$SVLAD  012430           2349    2355    2361#
$SVPC = 000500            160#    165
$SWR  = 161000            125#    226     227     747     749     750     890     912     933     983    1000    1014    1029
                         1049    1093    1140    1182    1225    1270    1302    1330    1382    1419    1448    1478    1519
                         1547    1578    1605    1631    1674    1692    1709    1744    1781    1815    1845    1874    1902
                         1933    1958    1984    2013    2124    2130    2141    2147    2149    2332    2333    2334    2335
                         2340    2352    2354    2355    2356    2361    2364    2367    2370
$SWREG  001110            247#    770
$SWMSK= 000000           2335
$TESTN  001072            238#    723*   2362*   3321    3323    3325    3327    3329    3331    3333    3335    3337    3339
                         3341
$TKB    001046            219#   2594    2605    2622    2676    2682
$TKS    001044            218#   2594    2603    2619    2643*   2674    2680
$TN   = 000046            124#    886     890#    908     912#    929     933#    979     983#    996    1000#   1010    1014#
                         1025    1029#   1045    1049#   1089    1093#   1136    1140#   1178    1182#   1221    1225#   1266
```

```
                        1270#  1298   1302#  1326   1330#  1378   1382#  1415   1419#  1444   1448#  1474   1478#
                        1515   1519#  1543   1547#  1574   1578#  1601   1605#  1627   1631#  1670   1674#  1688
                        1692#  1705   1709#  1740   1744#  1777   1781#  1811   1815#  1841   1845#  1870   1874#
                        1898   1902#  1929   1933#  1954   1958#  1980   1984#  2009   2013#
$TPB    001052           221#  2501*  2512
$TPFLG  001057           225#  2450   2512
$TPS    001050           220#  2499   2512
$TRAP   014232           742   2741#
$TRAP2  014254          2752#  2763
$TRP  = 000011          2756#  2765#  2766#  2767#  2768#  2769   2770#  2771   2772#  2773#  2774#
$TRPAD  014266          2746   2763#
$TSTM   000504           184#
$TSTNM  001002           198#  2111*  2129*  2196   2331   2361*  2362   2367   2370
$TTYIN  014166          2701   2702   2719   2723#
$TYPBN= ****** U        2768
$TYPOS= ****** U        2768
$TYPE   012746          2404   2450#  2756   2764
$TYPEC  013160          2480   3487   2494   2499#  2500   2645
$TYPEX  013226          2505   2507   2510#
$TYPOC  013254          2542#  2765
$TYPON  013270          2541   2544#  2767
$TYPOS  013230          2537#  2766
$UNIT   001100           241#   726*   862*   865*  2160*
$UNITM  000510           186#
$USWR   001112           248#
$VECT1  001136           273#   815
$VECT2  001140           274#
$XTSTR  012340          2343#
$$GET4= 000000          2141#
$0FILL  013453          2538#  2542*  2552   2587#
$40CAT= ****** U        2340
.     = 021514           129#   135    136#   140#   146#   150#   155#   160    161#   163#   165#   171    172#
                         174#   176#   195#   230    703#   704#   735    749    750   2106#  2149   2150#  2277#
                        2296   2320   2370   2426#  2512   2594   2723#  2724   2730   2833#
.SERRT  001146           295#
.SX   = 000500           171#   176
```

G08


```
CLKTST      6#  1029  1225  1302  1330  1382  1419  1448  1478
COMMEN      1#   122#
ENDCOM      1#   122#
ENDPAS      7#  2138
ERROR      16#   898   920   944   958   992  1006  1020  1039  1056  1067  1074  1086  1099  1111
          1119  1132  1147  1152  1159  1165  1173  1189  1194  1201  1208  1216  1236  1241  1248
          1254  1261  1293  1310  1315  1323  1347  1359  1368  1398  1408  1439  1469  1508  1530
          1536  1559  1569  1569  1597  1622  1641  1649  1655  1663  1683  1701  1724  1732  1759
          1772  1795  1806  1835  1865  1887  1893  1918  1924  1950  1974  1995  2006  2045  2051
          2065  2079  2174
ESCAPE      1#   122#
GETPRI      1#   122#
GETSWR      1#   122#
MULT        1#   122#
NEWTST      1#   122#   886   908   929   979   996  1010  1025  1045  1089  1136  1178  1221  1266
          1298  1326  1378  1415  1444  1474  1515  1543  1574  1601  1627  1670  1688  1705  1740
          1777  1811  1841  1870  1898  1929  1954  1980  2009
POP         1#    9#   122#
PUSH        1#    9#   122#
REPORT      1#   122#   900   922   946   960  1058  1076  1102  1121
SCOPE      17#   889   911   932   982   999  1013  1028  1048  1092  1139  1181  1224  1269  1301
          1329  1381  1418  1447  1477  1518  1546  1577  1604  1630  1673  1691  1708  1743  1780
          1814  1844  1873  1901  1932  1957  1983  2012  2128
SETPRI      1#   122#
SETTRA   2756#  2765  2766  2767  2769  2771  2772  2773
SETUP       1#    9#   122#   730
SKIP        1#   122#
SLASH       1#   122#
SPACE     122#
STARS       1#    9#   12_#   130   158   168   170   177   191   230   233   886   888   908   910
          929   931   979   981   996   998  1010  1013  1025  1027  1045  1047  1089  1091  1136
         1138  1178  1180  1221  1223  1266  1268  1298  1300  1326  1328  1378  1380  1415  1417
         1444  1446  1474  1476  1515  1517  1543  1545  1574  1576  1601  1603  1627  1629  1670
         1672  168_  1690  1705  1707  1740  1742  1777  1779  1811  1813  1841  1843  1870  1872
         1_08  1900  1929  1931  1954  1956  1980  1982  2009  2011  2122  2181  2151  2233  2281
         2283  2299  2301  2328  2372  2374  2435  2514  2593  2596  2664  2693  2735  2776  2781
         2837  2843
SWRSU       1#   122#   751#
TRMTRP   2756#
TYPBIN      1#   122#
TYPDEC      1#   122#
TYPNOM      1#   122#
TYPNUM      1#   122#
TYPOCS      1#   122#
TYPOCT      1#   122#  2245  2269  2614
TYPTXT      1#   123#
$CLKTS      5#  1029  1225  1302  1330  1382  1419  1448  1478
$$CMRE    189#
$$CMTM    189#
$$ELSCA     1#   122#
$$NEXT      1#   122#   886   908   929   979   996  1010  1025  1045  1089  1136  1178  1221  1266
          1298  1326  1378  1415  1444  1474  1515  1543  1574  1601  1627  1670  1688  1705  1740
          1777  1811  1841  1870  1898  1929  1954  1980  2009
$$SET    2756#  2765  2766  2767  2769  2771  2772  2773
$$SETM    767#
$$SKIP      1#   122#
```

```
.EQUAT          1#       9#     12
.MERGE          1#
.KTI1           1#
.SETUP          1#       9#    144
.SWRHI          1#
.SACTI          1#      10#    156
.SAPTB          1#      10#   2318
.SAPTH          1#      10#    166
.SAPTY          1#
.SASTA          1#
.SCATC          1#
.SCMTA          1#      11#    189
.SD820          1#
.SDB20          1#
.SDIV           1#
.SEOP           1#      11#   2120
.SERRO          1#
.SERRT          1#       9#   2231
.SMULT          1#
.SPOWE          1#
.SRAND          1#
.SRODE          1#
.SRODC          1#
.SSEAD          1#      11#   2591
.SR2AZ          1#
.SSAVE          1#
.SSB2D          1#
.SSB20          1#
.SSCOP          1#      10#   2326
.SSIZE          1#
.SSUPR          1#
.STRAP          1#       8#   2733
.STYPB          1#
.STYPD          1#
.STYPE          1#       8#   2433
.STYPO          1#       8#   2512
.S40CA          1#
.1170           1#
```

```
CMPB    2206  2385  2456  2471  2473  2481  2502  2506  2609  2641  2706  2712
DEC     1882  2132  2249  2851
DECB    2485  2488  2562  2573
EMT       16
HALT     903   925   949   963  1061  1079  1105  1124  2176  2215  2226  2295  2319  2452  2810
        2863
INC      787   788   797   798   806   849   862   865   953  1320  1492  1494  1496  1637  1646
        1914  1992  2039  2085  2096  2100  2112  2130  2197  2314  2414  2568  2576  2656
INCB    1968  2000  2072  2086  2194  2361  2508  2850
IOT       17
JMP      151   152   153   855   881  2116  2147  2158  2161
JSR      779   910   901   923   947   961  1059  1077  1103  1122  1143  1185  1232  1334  1351
        1390  1402  1423  1432  1452  1461  1525  1548  1563  1581  1606  1615  1715  1746  1764
        1786  1816  1828  1846  1857  2033  2142  2203  2209  2404  2461  2480  2487  2494  2645
        2793  2798  2803  2808  2853
MOV      709   710   711   713   727   728   732   736   738   739   740   741   742   743   744
         745   746   749   750   753   754   755   756   761   763   764   765   770   782   783
         784   789   794   795   799   803   814   817   819   826   841   850   851   852   867
         869   870   871   876   877   890   891   904   912   913   926  1000  1001  1007  1014
        1015  1021  1033  1034  1041  1141  1142  1175  1183  1184  1218  1230  1231  1263  1270
        1271  1272  1273  1275  1276  1286  1292  1294  1295  1336  1337  1338  1350  1361  1372
        1387  1388  1389  1401  1412  1425  1426  1441  1454  1455  1471  1484  1498  1500  1507
        1522  1523  1532  1540  1550  1551  1562  1571  1579  1580  1591  1599  1609  1610  1624
        1697  1712  1713  1727  1736  1748  1749  1763  1774  1783  1784  1793  1802  1808  1818
        1819  1837  1848  1849  1853  1867  1878  1906  1962  1965  1971  1972  1988  1997  2003
        2004  2014  2015  2016  2018  2024  2025  2037  2068  2077  2078  2081  2082  2095  2135
        2139  2157  2163  2164  2170  2172  2173  2196  2198  2219  2222  2240  2245  2254  2259
        2264  2266  2270  2284  2285  2286  2287  2288  2289  2290  2291  2292  2293  2294  2302
        2303  2304  2305  2306  2307  2308  2309  2310  2311  2312  2345  2346  2348  2351  2358
        2363  2364  2367  2368  2381  2382  2389  2393  2398  2399  2401  2403  2413  2419  2420
        2454  2455  2460  2468  2483  2537  2545  2546  2547  2553  2560  2578  2579  2580  2581
        2582  2614  2638  2643  2672  2673  2700  2701  2716  2717  2718  2719  2741  2742  2746
        2752  2753  2788  2814  2845  2846  2849
MOVB     748   815  1912  2091  2200  2208  2362  2366  2376  2377  2379  2465  2493  2501  2538
        2539  2542  2543  2544  2548  2551  2552  2571  2605  2622  2676  2682  2705  2710  2744
        2784  2787  2818
NEG     1503  2549
NOP      987  1307  1343  1355  1366  1396  1404  1434  1463  1534  1555  1565  1586  1592  1617
        1729  1755  1766  1793  1800  1830  1859  2143  2144  2145
RESET    988  1051  1082  1093  1127  1140  1169  1182  1212  1229  1257  1274  1306  1386  1431
        1631  1658  1674  1692  1744  1781  1815  1845  1856  1876  1904  1937  1959  1987  2013
        2043  2049  2067  2141  2783
ROL     2555  2557  2558  2559  2561
ROLB     832   834   836
RTI      762  2093  2098  2101  2228  2318  2369  2470  2583  2644  2692  2720  2754
RTS      717  2268  2421  2510  2747  2821
RTT     2166
SUB      791  1501  2171  2199  2396
SUBB     828   838
TRAP    2756  2765  2766  2767  2769  2771  2772  2773
TST      786   796   804   892   896   914   918   942   956   967   969   971  1002  1016  1029
        1035  1054  1072  1097  1117  1225  1288  1302  1330  1362  1419  1448  1478  2020  2058
        2062  2213  2220  2272  2347  2391  2409  2411  2467  2475  2497  2566  2636  2651  2743
TSTB     934   939   951   984   989  1308  1318  1341  1353  1364  1394  1429  1458  1487  1490
        1520  1552  1583  1612  1644  1659  1677  1680  1695  1699  1718  1752  1789  1824  1854
        1880  1907  1920  1941  1944  1963  1990  2354  2383  2394  2407  2450  2499  2603  2619
```

```
        2674    2680    2785    2816    2819    2847    2856
.ASCII   227     228    3314
.ASCIZ   229    2151    2276    2322    2724    2725    2726    2728    2823    2825    2827    2829    2867    2871    2875
        2882    2886    2891    2895    2899    2903    2909    2914    2919    2925    2931    2936    2941    2947    2954
        2959    2964    2969    2975    2981    2987    2992    2997    3004    3010    3016    3021    3026    3032    3036
        3042    3047    3052    3057    3063    3068    3072    3076    3084    3088    3094    3100    3107    3111    3115
        3122    3126    3130    3137    3143    3149    3156    3162    3168    3176    3180    3184    3191    3197    3201
        3207    3212    3217    3221    3226    3231    3235    3241    3244    3251    3255    3259    3263    3267    3271
        3276    3283    3288    3295    3302    3310    3315
.BLKB   2723
.BLKW    703     704    2106
.BYTE    198     199     204     205     213     214     222     223     224     225     245     246     256     257     264
         265     267     268     270     271    2149    2210    2211    2422    2423    2424    2584    2585    2586    2587
        2721    2722
.DSABL  2661
.ENABL     1    2594
.END    3343
.ENDC     16     108     122     131     144     159     163     165     169     171     178     192     196     198     226
         227     231     234     256     264     267     270     273     274     275     276     279     736     737     740
         742     744     746     747     749     751     772     887     888     889     890     901     902     909     910
         911     912     923     924     930     931     932     933     947     948     961     962     980     981     982
         983     997     998     999    1000    1011    1012    1013    1014    1026    1027    1028    1029    1046    1047
        1048    1049    1059    1060    1077    1078    1090    1091    1092    1093    1103    1104    1122    1123    1137
        1138    1139    1140    1179    1180    1181    1182    1222    1223    1224    1225    1267    1268    1269    1270
        1299    1300    1301    1302    1327    1328    1329    1330    1379    1380    1381    1382    1416    1417    1418
        1419    1445    1446    1447    1448    1475    1476    1477    1478    1516    1517    1518    1519    1544    1545
        1546    1547    1575    1576    1577    1578    1602    1603    1604    1605    1628    1629    1630    1631    1671
        1672    1673    1674    1689    1690    1691    1692    1706    1707    1708    1709    1741    1742    1743    1744
        1778    1779    1780    1781    1812    1813    1814    1815    1842    1843    1844    1845    1871    1872    1873
        1874    1899    1900    1901    1902    1930    1931    1932    1953    1955    1956    1957    1958    1981    1982
        1983    1984    2010    2011    2012    2013    2123    2124    2126    2129    2134    2137    2138    2139    2141
        2147    2149    2150    2182    2192    2234    2249    2278    2282    2284    2300    2302    2329    2332    2335
        2340    2342    2353    2354    2356    2361    2363    2367    2370    2373    2375    2436    2465    2515    2594
        2595    2597    2625    2661    2665    2693    2694    2701    2703    2706    2708    2724    2730    2736    2742
        2745    2764    2765    2766    2767    2768    2769    2770    2771    2772    2773    2774    2777    2782    2808
        2844
.EQUIV    16      17      25      40      41      70      71      72      73      74      75      76      77      78      79
          98      99     100     101     102     103     104     105     106     107
.EVEN    234    2150    2277    2426    2833    3309    3313    3320
.IF       14      80     108     130     144     158     161     163     168     170     177     191     195     197     226
         227     230     231     233     256     264     267     270     273     274     275     276     277     279     731
         736     738     740     742     744     746     747     749     767     886     888     890     901     908     910
         912     923     929     931     933     947     961     979     981     983     996     998    1000    1010    1012
        1014    1025    1027    1029    1045    1047    1049    1059    1077    1089    1091    1093    1103    1122    1136
        1138    1140    1178    1180    1182    1221    1223    1225    1266    1268    1270    1298    1300    1302    1326
        1328    1330    1378    1380    1382    1415    1417    1419    1444    1446    1448    1474    1476    1478    1515
        1517    1519    1543    1545    1547    1574    1576    1578    1601    1603    1605    1627    1629    1631    1670
        1672    1674    1688    1690    1692    1705    1707    1709    1740    1742    1744    1777    1779    1781    1911
        1813    1815    1841    1843    1845    1870    1872    1874    1898    1900    1902    1929    1931    1933    1954
        1956    1958    1980    1982    1984    2009    2011    2013    2122    2123    2124    2125    2126    2128    2133
        2136    2138    2139    2141    2147    2149    2150    2181    2191    2233    2248    2264    2281    2283    2299
        2301    2328    2331    2335    2340    2352    2354    2355    2356    2361    2362    2364    2369    2370    2372
        2374    2435    2456    2514    2593    2595    2596    2597    2625    2664    2665    2693    2701    2702    2706
        2707    2723    2724    2730    2735    2741    2745    2756    2765    2766    2767    2768    2769    2771    2772
        2773    2774    2776    2781    2837    2843
.IFF      14     131     159     163     165     169     171     178     192     196     198     226     231     234     737
```

```
        887   888   889   890   901   909   910   911   912   923   930   931   932   933   947
        961   980   981   982   983   997   998   999  1000  1011  1012  1013  1014  1026  1027
       1028  1029  1046  1047  1048  1049  1059  1077  1090  1091  1092  1093  1103  1122  1137
       1138  1139  1140  1175  1180  1181  1182  1222  1223  1224  1225  1267  1268  1269  1270
       1299  1300  1301  1302  1327  1328  1329  1330  1379  1380  1381  1382  1416  1417  1418
       1419  1445  1446  1447  1448  1475  1476  1477  1478  1516  1517  1518  1519  1544  1545
       1546  1547  1575  1576  1577  1578  1602  1603  1604  1605  1628  1629  1630  1631  1671
       1672  1673  1674  1689  1690  1691  1692  1706  1707  1708  1709  1741  1742  1743  1744
       1778  1779  1780  1781  1812  1813  1814  1815  1842  1843  1844  1845  1871  1872  1873
       1874  1899  1900  1901  1902  1930  1931  1932  1933  1955  1956  1957  1958  1981  1982
       1983  1984  2010  2011  2012  2013  2123  2125  2129  2134  2137  2149  2182  2192  2234
       2249  2278  2282  2284  2300  2302  2329  2353  2354  2355  2370  2373  2375  2436  2515
       2594  2597  2665  2667  2672  2693  2694  2703  2707  2724  2736  2742  2777  2782  2838
       2844
.IFT     901   923   947   961  1059  1077  1103  1122  2361  2667  2672
.IFTF   2360  2612  2665  2668
.IIF     230   234   737   740   746   747   749   750   901   902   923   924   947   948   961
        962  1059  1060  1077  1078  1103  1104  1122  1123  2124  2129  2130  2149  2150  2246
       2271  2332  2333  2334  2335  2339  2361  2367  2370  2512  2594  2615  2716  2724  2730
       2764  2765  2766  2767  2769  2771  2772  2773
.IRP     144   886   908   929   979   996  1010  1025  1045  1089  1136  1178  1221  1266  1298
       1326  1378  1415  1444  1474  1515  1543  1574  1601  1627  1670  1688  1705  1740  1777
       1811  1841  1870  1898  1929  1954  1980  2009
.LIST      1     2   122   135   144   226   231   234   751   886   890   908   912   929   933
        979   983   996  1000  1010  1014  1025  1029  1045  1049  1089  1093  1136  1140  1178
       1182  1221  1225  1266  1270  1298  1302  1326  1330  1378  1382  1415  1419  1444  1448
       1474  1478  1515  1519  1543  1547  1574  1578  1601  1605  1627  1631  1670  1674  1688
       1692  1705  1709  1740  1744  1777  1781  1811  1815  1841  1845  1870  1874  1898  1902
       1929  1933  1954  1958  1980  1984  2009  2013  2129  2141  2335  2693  2756  2764  2765
       2766  2767  2768  2769  2770  2771  2772  2773  2774
.MACRO     1     5     6     7   189   767  2756
.MCALL     8     9    10    11   122   231   751
.MEXIT   278
.NLIST     1     3   122   135   144   226   231   234    51   886   890   908   912   929   933
        979   983   996  1000  1010  1014  1025  1029  1045  1049  1089  1093  1136  1140  1178
       1182  1221  1225  1266  1270  1298  1302  1326  1330  1378  1382  1415  1419  1444  1448
       1474  1478  1515  1519  1543  1547  1574  1578  1601  1605  1627  1631  1670  1674  1688
       1692  1705  1709  1740  1744  1777  1781  1811  1815  1841  1845  1870  1874  1898  1902
       1929  1933  1954  1958  1980  1984  2009  2013  2129  2141  2335  2693  2756  2764  2765
       2766  2767  2768  2769  2770  2771  2772  2773  2774
.PAGE    189   279
.REPT    135
.SBTTL    12   156   166   189   231   279   730   886   908   929   979   996  1010  1025  1045
       1089  1136  1178  1221  1266  1298  1326  1378  1415  1444  1474  1515  1543  1574  1601
       1627  1670  1688  1705  1740  1777  1811  1841  1870  1898  1929  1954  1980  2009  2120
       2231  2280  2326  2373  2433  2512  2591  2733  2756  2775  2835
.WORD    135   137   138   141   147   148   164   182   183   184   185   186   187   197   200
        201   202   203   206   207   208   209   210   211   212   215   216   217   236   237
        238   239   240   241   242   243   247   248   249   262   266   269   272   273   274
        275   276   672   673   674   675   678   679   680   681   682   683   684   685   699
        700   701  1144  1186  1233  1335  1352  1391  1403  1424  1433  1453  1462  1511  1512
       1526  1549  1564  1582  1607  1616  1716  1747  1765  1787  1817  1829  1847  1858  2034
       2103  2104  2105  2133  2136  2148  2177  2178  2257  2262  2317  2405  2462  2509  2588
       2763  3321  3323  3325  3327  3329  3331  3333  3335  3337  3339  3341
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DZDLDA/SOL/CRF/NL:TOC/PAGNUM=DZDLDA.SML,DZDLDA.P11
RUN-TIME: 49 51 6 SECONDS
RUN-TIME RATIO: 598/107=5.5
CORE USED:  34K  (67 PAGES)
```

N08

Spooler runtime 14 Seconds, 61 KCS, 390 disk reads, 3 disk writes, 103 pages