11/24

11/24 OPTIONS DIAG
CJKDFA0

AH-F604A-MC
FICHE 1 OF 1

B 1

SEQ 0001

000000 .REPT 0

IDENTIFICATION

PRODUCT CODE: AC-F602A-MC

PRODUCT NAME: CJKDFA0 11/24 OPTIONS DIAGNOSTIC

PRODUCT DATE: APR-81

MAINTAINER: DIAGNOSTIC ENGINEERING

```
                                HISTORY
                                -------
REVISION A                      FIRST RELEASE OF DIAGNOSTIC
```

TABLE OF CONTENTS

1.0 GENERAL PROGRAM INFORMATION

1.1 ABSTRACT

THIS PROGRAM TESTS BOTH SERIAL LINE UNITS (SLU'S) AND THE LINE TIME CLOCK (LTC) ON THE M7133 MODULE. ITS MAIN PURPOSE IS TO PROVIDE SCOPE LOOPING FOR REPAIR PERSONNEL. THE PROGRAM DETECTS FROM 85-95% OF ALL STUCK-AT-0, STUCK-AT-1 FAULTS. ERROR TYPE-OUTS IDENTIFY A FUNCTION BEING DONE OR A FUNCTION THAT FAILED AND TO WHAT LOGICAL PORTION OF THE BOARD IT FAILED ON (I.E., TRIED TO SET BIT 6 ON CSR OF SLU1). THIS PROGRAM IS BASICALLY A REWRITE OF THE DL11-W DIAGNOSTIC AND THEREFORE IS WRITTEN IN MACRO-11 USING THE SYSMAC MACRO PACKAGE. IT IS COMPATIBLE WITH ALL EXISTING MANUFACTURING AND FIELD SERVICE AUTOMATED TEST SYSTEMS.

1.2 SYSTEM REQUIREMENTS

A. HARDWARE REQUIREMENTS

- 11/24 CPU MODULE
- MINIMUM 8K OF MEMORY
- TURN AROUND JUMPER INSTALLED ON SLU2

B. SOFTWARE ENVIRONMENTS

- XXDP STAND-ALONE
- XXDP CHAIN MODE
- APT
- ACT
- SLIDE

1.3 RELATED DOCUMENTS AND STANDARDS

- DIAGNOSTIC ENGINEERING FUNCTIONAL SPECIFICATION FOR 11/24 ON BOARD OPTIONS TEST
  BGI-79-004-00-U

- KDF11-UA (M7133) MODULE SPECIFICATION REV. 0.
  25-AUGUST-78 BILL BERNSTEIN

- STANDARD APT SYSTEM TO PDP-11 DIAGNOSTIC INTERFACE REV. 15
  16-FEBRUARY-76 APT SYSTEM GROUP

- DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS
  175-003-009-02

1.4 PREREQUISITE DIAGNOSTICS

THIS PROGRAM ASSUMES THE CORRECT OPERATION OF THE CPU INSTRUCTION SET. THIS IS TO BE VERIFIED BY EITHER:

CJKDBXX DCF11-AA CPU DIAGNOSTIC
OR
CJKDEXX F11 QUICK TEST

IF AN END PASS MESSAGE IS RECEIVED FROM CJKDEXX THERE IS NO NEED TO RUN THIS DIAGNOSTIC BECAUSE ALL DEVICES HAVE BEEN FULLY TESTED.

1.5 ASSUMPTIONS

IT IS ASSUMED THAT THE CONSOLE DEVICE THAT IS CONNECTED TO SLU1 IS OPERATING CORRECTLY. IF THE TERMINAL IS NOT OPERATING CORRECTLY FALSE INDICATIONS CAN BE EXPECTED.

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

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

2.1.1 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. ****NOTE****
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 (^C) HALTS THE TEST AND PRINTS "STOP" AT THE TERMINAL CONTINUING RESTARTS THE ECHO TEST.

C. START AT 210. ****NOTE****

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:

```
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[      [UPPER CASE ALPHA]
```

2.2 PROGRAM OPTIONS

```
BIT15     - HALT ON ERROR
BIT14     - SCOPE LOOP
BIT13     - INHIBIT ERROR TYPEOUT
BIT12     - UNUSED
BIT11     - UNUSED
BIT10     - INHIBIT ERROR FLAGS TEST
BIT09     - LOOP ON ERROR
BIT08     - UNUSED
BIT07     - DISABLE SLU2 DATA TEST
BIT06     - INHIBIT LTC TESTS
BIT05     - INHIBIT ALL SLU TESTS (BOTH SLUS)
BIT04     - INHIBIT SLU1 TESTING
BIT03     - INHIBIT SLU2 TESTING
BIT02     - UNUSED
BIT01     - UNUSED
BIT00     - UNUSED
```

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF THE SOFTWARE SWITCH REGISTER DURING EXECUTION. TO DO THIS THE OPERATOR MUST TYPE A "CNTL G" (TYPING A "G" WITH THE "CTRL" KEY HELD DOWN AT THE SAME TIME). THIS IS PROCESSED AT KEY TIMES DURING THE PROGRAM (I.E. ON ERRORS, IN BETWEEN EACH TEST). A PROBLEM CAN OCCUR SINCE THE PROGRAM MAY BE TESTING THE SLU CONNECTED TO THE CONSOLE DEVICE AND HAVE THE SLU IN MAINTENANCE MODE. IF THIS HAPPENS IT SHOULD NOT CAUSE AN ERROR BUT THE "CNTL G" MAY BE LOST, SO IF THE PROGRAM DOES NOT RESPOND TO THE FIRST "CNTL G" TYPE A FEW MORE UNTIL THE RESPONSE IS RECEIVED. WHEN THE "CNTL G" IS RECEIVED THE PROGRAM WILL RESPOND WITH:

```
SWR - XXXXXX     NEW
```

WHERE XXXXXX IS EQUAL TO THE PRESENT SWITCH REGISTER CONTENTS IN OCTAL. THE OPERATOR CAN THEN TYPE:

1. <CR> IF NO CHANGES ARE TO BE MADE.
2. 6 DIGITS <CR> TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER SETTING.
3. CONTROL-U IF THE OPERATOR MAKES AN ERROR WHILE INPUTTING THE NEW SWITCH REGISTER SETTING.

2.3 EXECUTION TIMES

1ST PASS RUNTIME(WORST CASE).....15 SECONDS
LONGEST TEST TIME.....12.5 SECONDS
ADDITIONAL RUN TIME(EXTRA UNITS).....NONE
LONGEST PASS TIME.....15 SECONDS

3.0 ERROR INFORMATION

3.1 ERROR REPORTING PROCEDURES

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 DATA HEADINGS]
XXXXXX   XXXXXX   XXXXXX   [ANY APPLICABLE DATA]

     NOTE: ''RCSR'' IS DEPENDENT ON THE FAILURE  THEREFORE
           COULD BE TCSR, RBUF, TBUR, 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 CCURS. (SEE SECTION 3.2 FOR ERROR HALT INFORMATION.)

NOTE

FOR SOFTWARE SWITCH OPERATION, THE SWITCH REGISTER CAN BE HANGED BY TYPING A CONTROL-G AT THE CONSOLE DURING ERROR PRITNOUTS. 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.

3.2 ERROR HALTS

THERE ARE 5 ERRORS IN THIS PROGRAM THAT CAUSE A HALT. THESE ERRORS ARE IN TESTS 1, 2, 3, AND 7.

TEST 1 ERROR - ACCESS TO SLU1 TRANSMITTER CSR CAUSE TIME-OUT TRAP. THIS PROBLEM WILL PROBABLY CAUSE AN INABILITY OF THE MICRO-ODT TO RUN.

TEST 2 ERROR - ACCESS TO SLU1 TRANSMITTER DATA BUFFER CAUSED TIME-OUT TRAP. THIS PROBLEM WILL PROBABLY CAUSE AN INABILITY OF THE MICRO-ODT TO RUN.

TEST 3 ERROR - THE FIRST ERROR IS THAT DONE DID NOT CLEAR WHEN THE TRANSMITTER BUFFER WAS FILLED AS IT SHOULD.

THE SECOND ERROR INDICATES THAT DONE DID NOT RESET IN A REASONABLE TIME AFTER THE DATA BUFFER WAS FILLED, INDICATING THAT THE CHARACTER WAS NEVER TRANSMITTED. THIS COULD CAUSE MICRO-ODT TO NOT RUN OR GARBLED OUTPUT FROM THE MICRO-ODT.

TEST 7 ERROR - THIS ERROR INDICATES THAT THE SLU CANNOT BE TAKEN OUT OF MAINTENANCE MODE. MICRO-ODT SHOULD BE UNEFFECTED BY THIS ERROR.

4.0 PERFORMANCE AND PROGRESS REPORTS

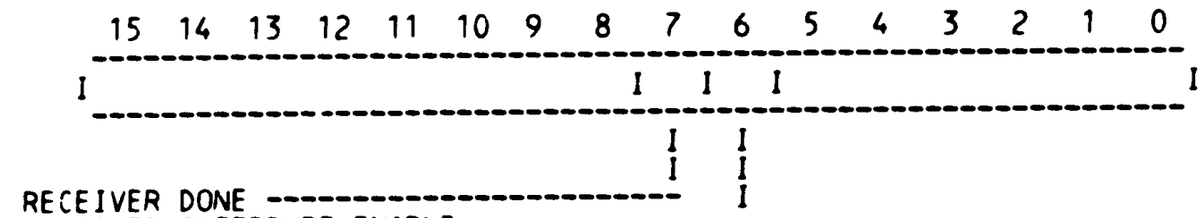
THE ONLY REPORT FROM THIS PROGRAM OTHER THAN ERROR REPORTS IS THE END PASS MESSAGE. IT IS IN THE FORM:

END PASS #XXXXX

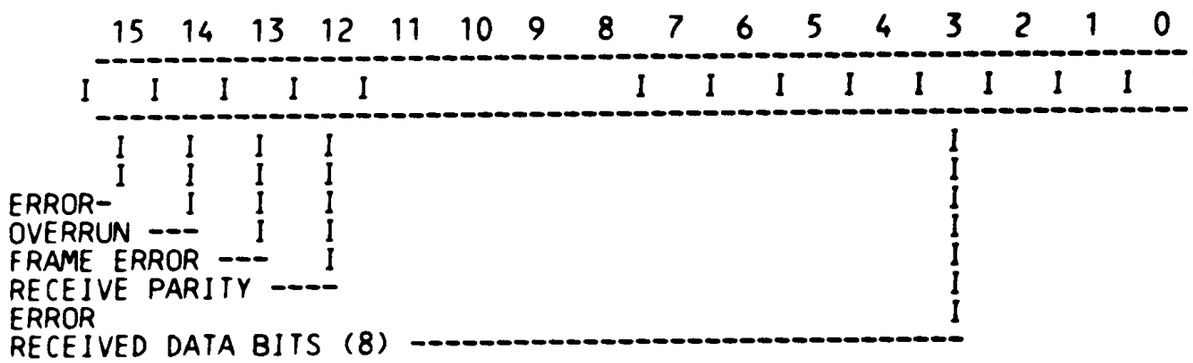
WHERE XXXXX IS THE DECIMAL NUMBER OF END OF PASSES COMPLETED.

5.0 DEVICE INFORMATION TABLES

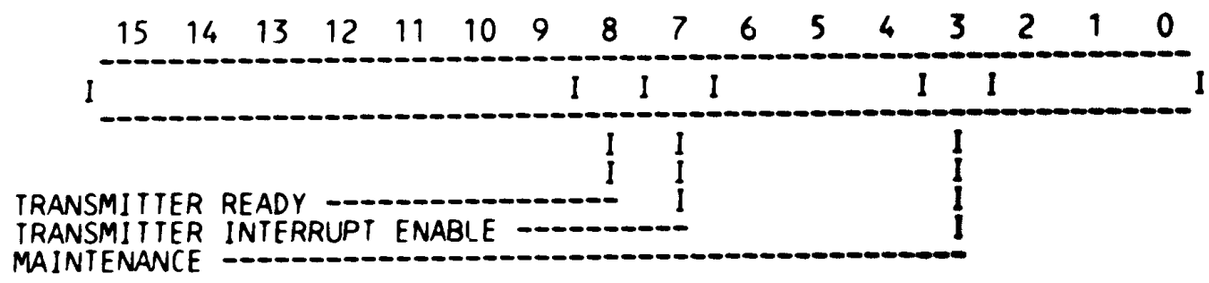
SLU1 RCSR



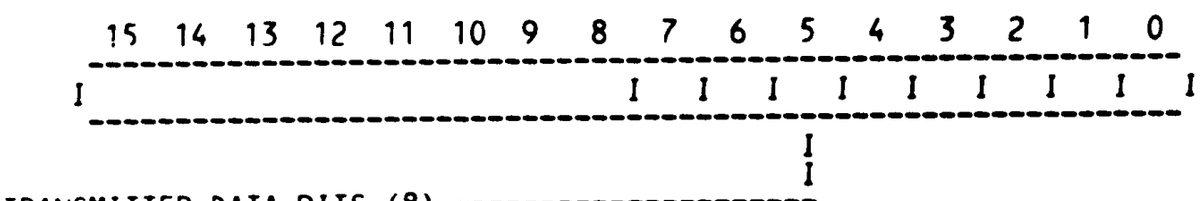
SLU1 RBUF



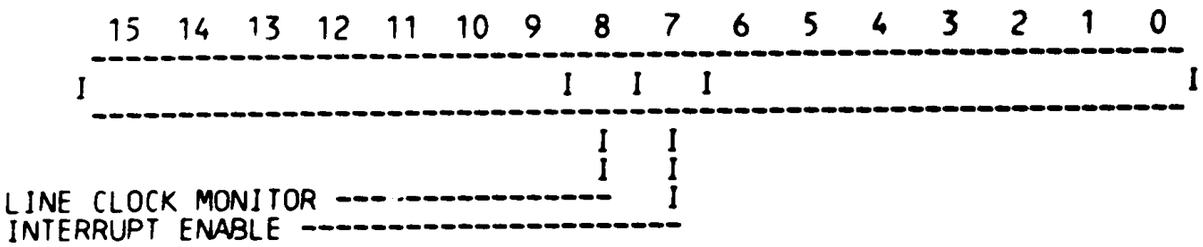
SLU1 XCSR



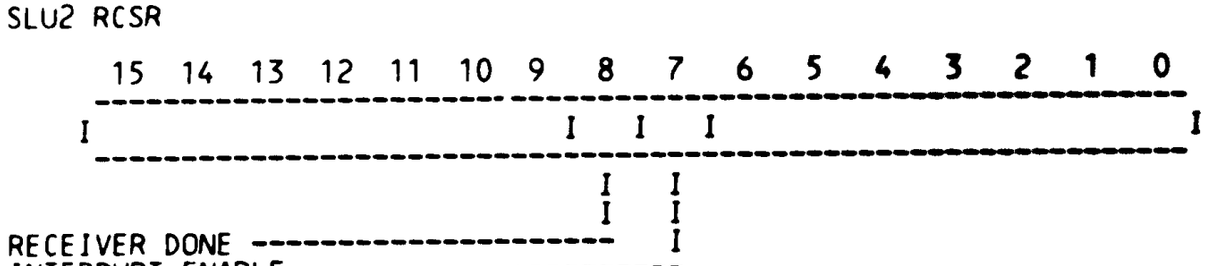
SLU1 XBUF



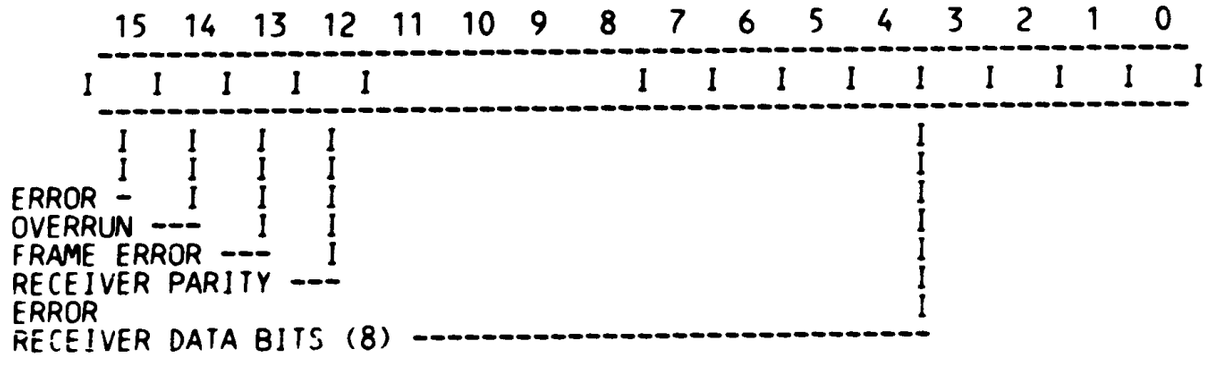
LTC CSR



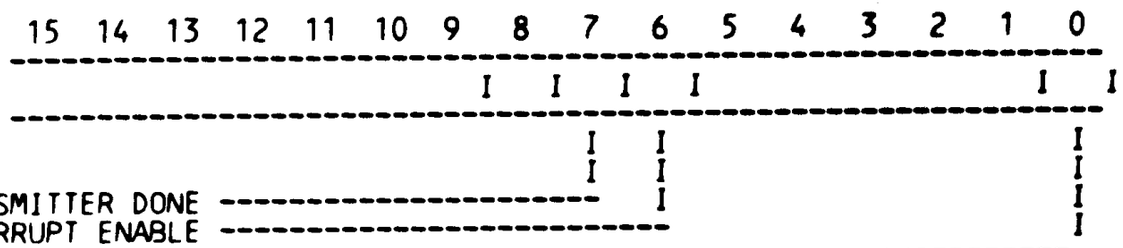
SLU2 RCSR



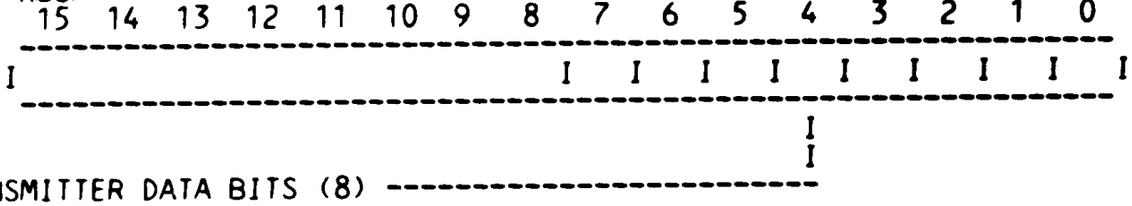
SLU2 RBUF



SLU2 XCSR



SLU2 XBUF



6.0 PROGRAM DESCRIPTION

6.1 PROGRAM EXECUTION CHARACTERISTICS

THIS PROGRAM TESTS ALL THE SELECTED DEVICES AS A SINGLE UNIT. IT FIRST VERIFIES THAT ALL REGISTERS CAN BE ACCESSED, THE WRITEABLE BITS ARE ABLE TO BE WRITTEN TO AND ARE UNIQUE ON ALL THREE DEVICES. IT THEN CHECKS EACH DEVICE FOR INTERRUPTS AND FOR DATA RELIABILITY. FINALLY IT SETS ALL SELECTED DEVICES UP TO GO AT THE SAME TIME ENABLES INTERRUPTS AND STARTS THEM OFF. THIS TEST CHECKS SYSTEM INTERACTION. WHEN ALL TESTS HAVE BEEN COMPLETED THE END OF PASS MESSAGE IS TYPED.

6.2 SUBTEST SUMMARIES

TEST1 TEST ABILITY TO ACCESS SLU1 TRANSMITTER CONTROL AND STATUS REGISTER. HALTS IF ACCESS CAUSES TIMEOUT TRAP.

TEST2 TEST ABILITY TO ACCESS SLU1 TRANSMITTER BUFFER. HALTS IF ACCESS CAUSES TIMEOUT TRAP.

TEST3 TEST SLU1 TRANSMITTER BIT7 (DONE) CLEARS WHEN TRANSMITTER BUFFER IS LOADED. THE BUFFER IS LOADED WHICH SHOULD CLEAR

THE DONE BIT. AFTER IT IS VERIFIED THAT 'DONE' CLEARS THE PROGRAM WAITS TO RECEIVE DONE BACK AFTER THE DATA IS TRANSFERRED OUT OF THE BUFFER. IF DONE DOES NOT INITIALLY CLEAR OR FAILS TO RESET THE PROGRAM HALTS.

TEST4 TEST THAT SLU1 TRANSMITTER BIT7 (DONE) SETS WITH RESET. THE TRANSMITTER BUFFER IS LOADED WITH A CHARACTER, AS SOON AS 'DONE' SETS A SECOND CHARACTER IS LOADED INTO THE BUFFER. BECAUSE THE FIRST CHARACTER IS STILL BEING SHIFTED OUT OF THE UART 'DONE' WILL NOT NORMALLY SET FOR AT LEAST 1 MS (DEPENDING ON BAUD RATE). THE PROGRAM ISSUES A RESET BEFORE THE TIME IS UP AND IMMEDIATELY CHECKS FOR 'DONE', IF IT IS SET THE PROGRAM ASSUMES IT WAS SET BY THE RESET INSTRUCTION. THIS ERROR DOES NOT CAUSE A HALT.

TEST5 TEST ABILITY TO ACCESS SLU1 RECEIVER CONTROL AND STATUS REGISTER. AN ERROR IS REPORTED IF ACCESS CAUSES A TIMEOUT TRAP.

TEST6 TEST ABILITY TO ACCESS SLU1 RECEIVER BUFFER. AN ERROR IS REPORTED IF ACCESS CAUSES A TIMEOUT TRAP.

TEST7 TEST THAT SLU1 BIT2 (MAINTENANCE) CAN BE SET AND RESET. BECAUSE AN ERROR ON THIS TEST MAY LEAVE THE SLU IN AN UNKNOWN STATE, ERRORS CAUSE THE PROGRAM TO HALT.

TEST10 TEST THAT SLU1 TRANSMITTER INTERRUPT ENABLE (BIT6) CAN BE SET AND RESET. THIS TEST CHECKS THAT THE BIT CAN BE WRITTEN INTO AND READ, CLEARED BY WRITING A ZERO TO IT AND CLEARED BY A 'RESET'.

TEST11 TEST THAT SLU1 RECEIVER INTERRUPT ENABLE (BIT6) CAN BE SET AND RESET. SAME TEST AS FOR TRANSMITTER INTERRUPT ENABLE.

TEST12 TEST THAT SLU1 RECEIVER TEST 7 (DONE) SETS AND CLEARS PROPERLY. THIS TEST PUTS THE SLU INTO MAINTENANCE MODE AND TRANSMITS A CHARACTER. RECEIVER 'DONE' SHOULD SET WHEN CHARACTER IS RECEIVED. AFTER 'DONE' SETS THE PROGRAM ATTEMPTS TO CLEAR IT WITH A 'RESET'.

TEST13 TEST SLU1 THAT READING RECEIVER BUFFER CLEAR RECEIVER 'DONE'. A CHARACTER IS TRANSMITTED IN MAINTENANCE MODE, WHEN RECEIVER 'DONE' SETS THE RECEIVER BUFFER IS READ WHICH SHOULD CLEAR DONE.

TEST14 SAME AS TEST1 BUT DONE ONSLU2 AND ERROR DOES NOT CAUSE HALT.

TEST15 SAME AS TEST2 BUT FOR SLU2 AND ERROR DOES NOT CAUSE HALT.

TEST16 SAME AS TEST3 BUT FOR SLU2 AND ERROR DOES NOT CAUSE HALT.

TEST17-21 SAME AS TEST4-6 BUT FOR SLU2.

TEST22 TEST SLU2 BREAK BIT (BIT0) CAN BE SET, CLEARED, AND RESET.

TEST23-26 SAME AS TEST 10-13 BUT FOR SLU2.

TEST27 TEST ABILITY TO ACCESS LINE CLOCK STATUS REGISTER. ERROR REPORTED IF ACCESS CAUSES TIMEOUT TRAP.

TEST30 TEST THAT LINE CLOCK INTERRUPT ENABLE (BIT6) CAN BE SET, CLEARED, AND "RESET".

TEST31 TEST THAT BIT7 (DONE) OF LINE CLOCK STATUS REGISTER SETS AND CAN BE CLEARED.

TEST32 UNIQUE INTERNAL ADDRESS TEST. THIS TEST WRITES A BIT INTO ONE OF THE DEVICE REGISTERS AND THEN VERIFIES THAT BIT IS NOT SET IN ANY OTHER REGISTER. THE TEST IS REPEATED FOR ALL THE REGISTERS.

TEST33 TEST THAT SLU1 TRANSMITTER INTERRUPTS ONLY WHEN ENABLED. THIS TEST CHECKS THAT THE TRANSMITTER ONLY INTERRUPTS WHEN ITS INTERRUPT ENABLE BIT IS SET.

TEST34 TEST SLU1 TRANSMITTER INTERRUPTS DO NOT OCCUR WHEN DISABLED. THIS TEST CHECKS THAT THE TRANSMITTER DOES NOT INTERRUPT WHEN THE PROCESSOR PRIORITY IS 7 OR THE INTERRUPT ENABLE BIT IS CLEARED.

TEST35 TEST SLU1 TRANSMITTER FOR DOUBLE INTERRUPTS. THIS TEST FIRST CHECKS THAT THE TRANSMITTER CAN INTERRUPT THEN MAKES SURE THAT ONLY ONE INTERRUPT IS REQUESTED FOR EACH SETTING OF "DONE".

TEST36 TEST THAT SLU1 TRANSMITTER INTERRUPT CLEARS WITH LOADING OF TRANSMITTER BUFFER. THIS TEST PUTS THE PROCESSOR AT 7, SETS TRANSMITTER INTERRUPT ENABLE AND FILLS THE TRANSMITTER BUFFER. WHEN "DONE" SETS THE SLU SHOULD HAVE AN INTERRUPT PENDING. THE PROGRAM THEN FILLS THE BUFFER AGAIN WHICH SHOULD CLEAR THE INTERRUPT.

TEST37-42 SAME AS TESTS FOR SLU1 RECEIVER INTERRUPTS AS TRANSMITTER INTERRUPTS.

TEST43 TEST SLU1 THAT RESET CLEARS RECEIVER INTERRUPTS. SET UP FOR RECEIVER INTERRUPT PENDING WITH PROCESSOR AT PRIOIRYT 7, ISSUE A "RESET" AND DROP PROCESSOR PRIORITY. THE RECEIVER SHOULD NOT INTERRUPT.

TEST44 TEST SLU1 THAT OVERRUN AND ERROR (BITS 14 AND 15) CAN BE SET. THIS TEST PUTS THE SLU1 IN MAINTENANCE MODE AND TRANSMITS 3 CHARACTERS WITHOUT READING THE RECEIVER BUFFER. THIS SHOULD CAUSE BITS 14 AND 15 TO SET.

TEST45 TEST SLU1 DATA PATH USING MAINTENANCE WRAP AROUND. THI

TEST TRANSMITS AN INCREMENTING DATA PATTERN AND VERIFIES THE CORRECT DATA IS RECEIVED.

TEST46-57 SAME AS TESTS 33-44 EXCEPT FOR SLU2 INSTEAD OF SLU1

TEST60 TEST THAT BREAK TRANSMITS ALL ZEROES. PUT ALL ONES INTO RECEIVER BUFFER THEN ISSUE A BREAK THE RECEIVER BUFFER SHOULD CONTAIN ZEROES.

TEST61 TEST THAT FRAMING ERROR (BIT13) CAN BE SET DURING BREAK. TRANSMIT A BREAK AND A CHARACTER JUST TO LET US KNOW WHEN TO LOOK FOR THE ERROR BIT. WHEN RECEIVER 'DONE' SETS BOTH BREAK AND CHARACTER SHOULD BE THERE, CHECK FOR BIT 13 IN RECEIVER STATUS REGISTER.

TEST62 TEST SLU2 DATA PATH USING WRAP CABLE CONNECTOR. SAME AS TEST 45 ON SLU1.

TEST63 TEST LINE TIME CLOCK INTERRUPTS PROPERLY.

TEST64 TEST LINE TIME CLOCK FOR DOUBLE INTERRUPTS.

TEST65 TEST THAT LINE TIME CLOCK INTERRUPT CLEARS WITH RESET.

TEST66 TEST THAT LINE TIME CLOCK INTERRUPT CLEARS BY CLEARING BIT7 OF LINE CLOCK STATUS REGISTER.

TEST67 TEST LINE TIME CLOCK REPEATABLILITY THIS TEST VERIFIES THAT THE PROCESSOR DOES THE SAME AMOUNT OF INSTRUCTIONS FOR TWO INTERRUPTS OF THE LINE CLOCK, THUS INDICATING EQUAL TIMES FOR EACH INTERRUPT. BECAUSE OF PROBLEMS OF SYNCHRONIZING THE PROCESSOR AND CLOCK A SMALL DEVIATION IS ALLOWED.

TEST70 BLAST TEST. THIS TEST RUNS ALL SELECTED DEVICES SIMULTANEOUSLY IN INTERRUPT MODE. AFTER 60 INTERRUPTS FROM THE LINE CLOCK OR 256 (10) BYTES HAVE BEEN TRANSFERRED BY THE SLU'S EVERYTHING IS SHUT DOWN. THE PROGRAM THEN VERIFIES THAT NO INTERRUPTS WERE LOST ON EITHER SLU AND THAT THE DATA TRANSFERRED WAS CORRECT.

NOTE: IF RUNNING UNDER THE APT ENVIRNOMENT MANY OF THE ABOVE TESTS ARE ONLY EXECUTED DURING FIRST PASS.

6.3 SPECIAL SUBROUTINE DESCRIPTIONS

6.3.1 ECHO TEST -

THIS ROUTINE WILL ECHO ANY CHARACTER TYPED ON EITHER SLU1 OR SLU2. DEFAULT IS TO THE CONSOLE DEVICE SLU1. THE TEST IS HALTED BY TYPING A CONTROL-C. TEST CAN BE RESTARTED AFTER HALTING BY JUST CONTINUING.

676
677
678
679
680
681
682
683
684
685
686
687
688
689

6.3.2 TERMINAL OUTPUT TEST -

THIS ROUTINE WILL OUTPUT ALL WRITEABLE CHARACTERS FOR THE OCTAL CODE
040 --> 377. 32 CHARACTERS ARE PRINTED ON EACH LINE. THE PATTERN IS
REPEATED EVERY THREE LINES.

7.0 LISTING

.ENDR

```
                                .TITLE  CJKDFA0 11/24 OPTIONS DIAGNOSTIC
                                ;*COPYRIGHT (C) JAN-81
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.
                                ;*
000001                          $TN=1
160000                          $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
                                .SBTTL  BASIC DEFINITIONS

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


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

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

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

```
                                PR7=    340             ;;PRIORITY LEVEL 7

                                ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
000340                          SW15=   100000
100000                          SW14=   40000
040000                          SW13=   20000
020000                          SW12=   10000
010000                          SW11=   4000
004000                          SW10=   2000
002000                          SW09=   1000
001000                          SW08=   400
000400                          SW07=   200
000200                          SW06=   100
000100                          SW05=   40
000040                          SW04=   20
000020                          SW03=   10
000010                          SW02=   4
000004                          SW01=   2
000002                          SW00=   1
000001                          .EQUIV  SW09,SW9
                                .EQUIV  SW08,SW8
                                .EQUIV  SW07,SW7
                                .EQUIV  SW06,SW6
                                .EQUIV  SW05,SW5
                                .EQUIV  SW04,SW4
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

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

```
                                .EQUIV  BIT00,BIT0

                                ;*BASIC ''CPU'' TRAP VECTOR ADDRESSES
000004                          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                          TBITVEC=14              ;;''T'' BIT
000014                          TRTVEC= 14              ;;TRACE TRAP
000014                          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                          PWRVEC= 24              ;;POWER FAIL
000030                          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                          TRAPVEC=34              ;;''TRAP'' TRAP
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
000400                                  AUSWR=  400
000001                                  $TN=    1
161000                                  $SSWR=  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 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS


000014                                  .-14                    ;THE BPT TRAP VECTOR POINTS TO THE
000014  015070                          .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  000137  003036                  JMP     START           ;DO INTERFACE TEST
000204  000137  020106                  JMP     ECHO            ;DO ECHO TEST
000210  000137  020364                  JMP     OUTTST          ;DO OUTPUT TEST TO TERMINAL
```

```
                                .=      500
                        .SBTTL  ACT11 HOOKS

                        ;;******************************************************************
                        ;HOOKS REQUIRED BY ACT11
000500  000500                  $SVPC=.                 ;SAVE PC
000500  000046                  .-46
000046  015022                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
000052  000052                  .=52
000052  000000                  .WORD   0               ;;2)SET LOC.52 TO ZERO
000500  000500                  .=$SVPC                 ;; RESTORE PC
                        .SBTTL  APT PARAMETER BLOCK

                        ;;******************************************************************
                        ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                        ;;******************************************************************
000500  000500                  .$X=.   ;;SAVE CURRENT LOCATION
000024  000024                  .-24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000200  000200                  200     ;;FOR APT START UP
000044  000044                  .-44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000500  000500                  $APTHDR ;;POINT TO APT HEADER BLOCK
000500  000500                  .=.$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  001100          $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)
```

```
                                .SBTTL  COMMON TAGS

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

                                        .=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  000000                  $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                  $GDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'GOOD' DATA
001022  000000                  $BDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'BAD' DATA
001024  000000                  $GDDAT: .WORD   0       ;;CONTAINS 'GOOD' DATA
001026  000000                  $BDDAT: .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  000000                  $TMP0:  .WORD   0       ;;USER DEFINED
001062  000000                  $TMP1:  .WORD   0       ;;USER DEFINED
001064  000000                  $TMP2:  .WORD   0       ;;USER DEFINED
001066  000000                  $TMP3:  .WORD   0       ;;USER DEFINED
001070  000000                  $TMP4:  .WORD   0       ;;USER DEFINED
001072  000000                  $ESCAPE:0               ;;ESCAPE ON ERROR ADDRESS
001074  077                     $QUES:  .ASCII  /?/     ;;QUESTION MARK
001075  015                     $CRLF:  .ASCII  <15>    ;;CARRIAGE RETURN
001076  000012                  $LF:    .ASCIZ  <12>    ;;LINE FEED
                                ;;**********************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;**********************************************************************
                                .EVEN
001100                  $MAIL:                          ;;APT MAILBOX
001100  000000                 $MSGTY: .WORD   AMSGTY  ;;MESSAGE TYPE CODE
001102  000000                 $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
001104  000000                 $TESTN: .WORD   ATESTN  ;;TEST NUMBER
001106  000000                 $PASS:  .WORD   APASS   ;;PASS COUNT
```

```
941  001110  000000
942  001112  000000
943  001114  000000
944  001116  000000
945  001120
946  001120      000
947  001121      000
948  001122  000000
949  001124  000400
950  001126  000000
951
952
953
954
955
956
957  001130      000
958  001131      000
959
960
961
962
963  001132  000000
964
965  001134      000
966  001135      000
967  001136  000000
968  001140      000
969  001141      000
970  001142  000000
971  001144      000
972  001145      000
973  001146  000000
974  001150  000300
975  001152  000000
976  001154  176500
977  001156  000000
978  001160
979
```

```
$DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
$UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
$MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
$MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
$ETABLE:                ;;APT ENVIRONMENT TABLE
$ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
$ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
$SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
$USWR:  .WORD   AUSWR   ;;USER SWITCHES
$CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
                        BITS 15-11=CPU TYPE
;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
;*                              11/70=06,PDQ=07,Q=10
;*                      BIT 10=REAL TIME CLOCK
;*                      BIT  9=FLOATING POINT PROCESSOR
;*                      BIT  8=MEMORY MANAGEMENT
$MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
$MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
;*                              900 NSEC CORE=001
;*                              300 NSEC BIPOLAR=002
;*                              500 NSEC MOS=003
$MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
$MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
$MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
$MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
$MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
$MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
$MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
$MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
$MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
$MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
$VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
$VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
$BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
$DEVM:  .WORD   ADEVM   ;;DEVICE MAP
$ETEND:
.MEXIT
```

```
                        .SBTTL  ERROR POINTER TABLE

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

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


001160                  $ERRTB:

                        .$ERRTB:
001160  000020                 .BLKW   20              ;THIS BLOCK OF 16 LOCATIONS IS HERE TO PACIFY SYSMAC


001220  020510                  EM5                     ;SLU1 TCSR DONE NOT SET WITH RESET
001222  027431                  DH5                     ;''TEST# ERR PC  TCSR''
001224  030212                  DT5                     ;$TESTN,$ERRPC,CTCSR
001226  000000                  0

001230  020552                  EM6                     ;SLU1 RCSR DID NOT RETURN SSYNC
001232  027456                  DH6                     ;''TEST# ERR PC  RCSR''
001234  030222                  DT6                     ;$TESTN,$ERRPC,CRCSR
001236  000000                  0

001240  020611                  EM7                     ;SLU1 RBUF DID NOT RETURN SSYNC
001242  027503                  DH7                     ;''TEST# ERR PC  RBUF''
001244  030232                  DT7                     ;$TESTN,$ERRPC,CRBUF
001246  000000                  0

001250  000004                  .BLKW   4               ;MORE PACIFICATION


001260  020650                  EM11                    ;''CAN NOT SET BIT2 OF SLU1 TCSR''
001262  027431                  DH5                     ;''TEST# ERR PC  TCSR''
001264  030212                  DT5                     ;$TESTN,$ERRPC,CTCSR
001266  000000                  0

001270  000004                  .BLKW   4               ;STILL MORE PACIFICATION


001300  020706                  EM13                    ;''RESET DID NOT CLEAR BIT2 OF SLU1 TCSR''
001302  027431                  DH5                     ;''TEST# ERR PC  TCSR''
001304  030212                  DT5                     ;$TESTN,$ERRPC,CTCSR
001306  000000                  0

001310  020754                  EM14                    ;''BIT6 OF SLU1 TCSR NOT CLEAR AFTER RESET2
001312  027431                  DH5                     ;''TEST# ERR PC  TCSR''
001314  030212                  DT5                     ;$TESTN,$ERRPC,CTCSR
001316  000000                  0
```

```
1036
1037  001320  021024          EM15    ;''SLU1 XMIT INTERRUPTED WITH PRIORITY 7''
1038  001322  027431          DH5     ;''TEST# ERR PC  TCSR''
1039  001324  030212          DT5     ;$TESTN,$ERRPC,CTCSR
1040  001326  000000          0
1041
1042  001330  021072          EM16    ;''CAN NOT SET BIT6 OF SLU1 TCSR''
1043  001332  027431          DH5     ;''TEST# ERR PC  TCSR''
1044  001334  030212          DT5     ;$TESTN,$ERRPC,CTCSR
1045  001336  000000          0
1046
1047  001340  021130          EM17    ;''CAN NOT CLEAR BIT6 OF SLU1 TCSR''
1048  001342  027431          DH5     ;''TEST# ERR PC  TCSR''
1049  001344  030212          DT5     ;$TESTN,$ERRPC,CTCSR
1050  001346  000000          0
1051
1052  001350  021170          EM20    ;''RESET DID NOT CLEAR BIT6 OF SLU1 TCSR''
1053  001352  027431          DH5     ;''TEST# ERR PC  TCSR''
1054  001354  030212          DT5     ;$TESTN,$ERRPC,CTCSR
1055  001356  000000          0
1056
1057  001360  021236          EM21    ;''BIT6 OF SLU1 RCSR NOT CLEAR AFTER RESET''
1058  001362  027456          DH6     ;''TEST# ERR PC  RCSR''
1059  001364  030222          DT6     ;$TESTN,$ERRPC,CRCSR
1060  001366  000000          0
1061
1062  001370  021306          EM22    ;''SLU1 RCVR INTERRUPT WITH PRIORITY 7''
1063  001372  027456          DH6     ;''TEST# ERR PC  RCSR''
1064  001374  030222          DT6     ;$TESTN,$ERRPC,CRCSR
1065  001376  000000          0
1066
1067  001400  021352          EM23    ;''CAN NOT SET BIT6 OF SLU1 RCSR''
1068  001402  027456          DH6     ;''TEST# ERR PC  RCSR''
1069  001404  030222          DT6     ;$TESTN,$ERRPC,CRCSR
1070  001406  000000          0
1071
1072  001410  021410          EM24    ;''CAN NOT CLEAR BIT6 OF SLU1 RCSR''
1073  001412  027456          DH6     ;''TEST# ERR PC  RCSR''
1074  001414  030222          DT6     ;$TESTN,$ERRPC,CRCSR
1075  001416  000000          0
1076
1077  001420  021450          EM25    ;''CAN NOT CLEAR BIT6 OF SLU1 RCSR WITH RESET2
1078  001422  027456          DH6     ;''TEST# ERR PC  RCSR''
1079  001424  030222          DT6     ;$TESTN,$ERRPC,CRCSR
1080  001426  000000          0
1081
1082  001430  021523          EM26    ;''SLU1 RECEIVER DONE NEVER SET''
1083  001432  027456          DH6     ;''TEST# ERR PC  RCSR''
1084  001434  030222          DT6     ;$TESTN,$ERRPC,CRCSR
1085  001436  000000          0
1086
1087  001440  021560          EM27    ;''RESET DID NOT CLEAR SLU1 RCVR DONE''
1088  001442  027456          DH6     ;''TEST# ERR PC  RCSR''
1089  001444  030222          DT6     ;$TESTN,$ERRPC,CRCSR
1090  001446  000000          0
1091
```

```
1092  001450  021623          EM30        ;''READING SLU1 RBUF DID NOT CLEAR RCVR DONE''
1093  001452  027456          DH6         ;''TEST# ERR PC  RCSR''
1094  001454  030222          DT6         ;$TESTN,$ERRPC,CRCSR
1095  001456  000000          0
1096
1097  001460  021675          EM31        ;SLU2 TCSR DID NOT RETURN SSYNC
1098  001462  027431          DH5         ;''TEST# ERR PC  TCSR''
1099  001464  030242          DT31        ;$TESTN,$ERRPC,TCSR
1100  001466  000000          0
1101
1102  001470  021734          EM32        ;SLU2 TBUF DID NOT RETURN SSYNC
1103  001472  027530          DH32        ;''TEST# ERR PC  TBUF''
1104  001474  030252          DT32        ;$TESTN,$ERRPC,TBUF
1105  001476  000000          0
1106
1107  001500  021773          EM33        ;''SLU2 TCSR DONE NOT CLEARED WITH TBUF FULL''
1108  001502  027431          DH5         ;''TEST# ERR PC  TCSR''
1109  001504  030242          DT31        ;$TESTN,$ERRPC,TCSR
1110  001506  000000          0
1111
1112  001510  022045          EM34        ;''SLU2 TCSR DONE NOT SET AFTER TRANSMIT''
1113  001512  027431          DH5         ;''TEST# ERR PC  TCSR''
1114  001514  030242          DT31        ;$TESTN,$ERRPC,TCSR
1115  001516  000000          0
1116
1117  001520  022113          EM35        ;''SLU2 TCSR DONE NOT SET WITH RESET''
1118  001522  027431          DH5         ;''TEST#  ERR PC  TCSR''
1119  001524  030242          DT31        ;$TESTN,ERRPC,TCSR
1120  001526  000000          0
1121
1122  001530  022155          EM36        ;''SLU2 RCSR DID NOT RETURN SSYNC''
1123  001532  027456          DH6         ;''TEST#  ERR PC  RCSR''
1124  001534  030262          DT36        ;$TESTN,ERRPC,RCSR
1125  001536  000000          0
1126
1127  001540  022214          EM37        ;SLU2 RBUF DID NOT RETURN SSYNC''
1128  001542  027503          DH7         ;''TEST#  ERR PC  RBUF''
1129  001544  030272          DT37        ;$TESTN,ERRPC,RBUF
1130  001546  000000          0
1131
1132  001550  022253          EM40        ;''BIT0 OF SLU2 TCSR NOT CLEAR AFTER RESET''
1133  001552  027431          DH5         ;''TEST# ERR PC  TCSR
1134  001554  030242          DT31        ;$TESTN,$ERRPC,TCSR
1135  001556  000000          0
1136
1137  001560  022323          EM41        ;''CAN NOT SET BIT0 OF SLU2 TCSR''
1138  001562  027431          DH5         ;''TEST# ERR PC  TCSR
1139  001564  030242          DT31        ;$TESTN,$ERRPC,TCSR
1140  001566  000000          0
1141
1142  001570  022361          EM42        ;''CAN NOT CLEAR BIT0 OF SLU2 TCSR''
1143  001572  027431          DH5         ;''TEST# ERR PC  TCSR
1144  001574  030242          DT31        ;$TESTN,$ERRPC,TCSR
1145  001576  000000          0
1146
1147  001600  022421          EM43        ;''RESET DID NOT CLEAR BIT0 OF SLU2 TCSR''
```

```
1148  001602  027431                DH5                     ;''TEST# ERR PC  TCSR
1149  001604  030242                DT31                    ;$TESTN,$ERRPC,TCSR
1150  001606  000000                0
1151
1152  001610  022467                EM44                    ;''BIT6 OF SLU2 TCSR NOT CLEAR AFTER RESET2
1153  001612  027431                DH5                     ;''TEST# ERR PC  TCSR
1154  001614  030242                DT31                    ;$TESTN,$ERRPC,TCSR
1155  001616  000000                0
1156
1157  001620  022537                EM45                    ;''SLU2 XMIT INTERRUPTED WITH PRIORITY 7''
1158  001622  027431                DH5                     ;''TEST# ERR PC  TCSR
1159  001624  030242                DT31                    ;$TESTN,$ERRPC,TCSR
1160  001626  000000                0
1161
1162  001630  022605                EM46                    ;''CAN NOT SET BIT6 OF SLU2 TCSR''
1163  001632  027431                DH5                     ;''TEST# ERR PC  TCSR
1164  001634  030242                DT31                    ;$TESTN,$ERRPC,TCSR
1165  001636  000000                0
1166
1167  001640  022643                EM47                    ;''CAN NOT CLEAR BIT6 OF SLU2 TCSR''
1168  001642  027431                DH5                     ;''TEST# ERR PC  TCSR
1169  001644  030242                DT31                    ;$TESTN,$ERRPC,TCSR
1170  001646  000000                0
1171
1172  001650  022703                EM50                    ;''RESET DID NOT CLEAR BIT6 OF SLU2 TCSR''
1173  001652  027431                DH5                     ;''TEST# ERR PC  TCSR
1174  001654  030242                DT31                    ;$TESTN,$ERRPC,TCSR
1175  001656  000000                0
1176
1177  001660  022751                EM51                    ;''BIT6 OF SLU2 RCSR NOT CLEAR AFTER RESET''
1178  001662  027456                DH6                     ;''TEST#  ERR PC  RCSR''
1179  001664  030262                DT36                    ;$TESTN,ERRPC,RCSR
1180  001666  000000                0
1181
1182  001670  023021                EM52                    ;''SLU2 RCVR INTERRUPT WITH PRIORITY 7''
1183  001672  027456                DH6                     ;''TEST#  ERR PC  RCSR''
1184  001674  030262                DT36                    ;$TESTN,ERRPC,RCSR
1185  001676  000000                0
1186
1187  001700  023065                EM53                    ;''CAN NOT SET BIT6 OF SLU2 RCSR''
1188  001702  027456                DH6                     ;''TEST#  ERR PC  RCSR''
1189  001704  030262                DT36                    ;$TESTN,ERRPC,RCSR
1190  001706  000000                0
1191
1192  001710  023123                EM54                    ;''CAN NOT CLEAR BIT6 OF SLU2 RCSR''
1193  001712  027456                DH6                     ;''TEST#  ERR PC  RCSR''
1194  001714  030262                DT36                    ;$TESTN,ERRPC,RCSR
1195  001716  000000                0
1196
1197  001720  023163                EM55                    ;''CAN NOT CLEAR BIT6 OF SLU2 RCSR WITH RESET2
1198  001722  027456                DH6                     ;''TEST#  ERR PC  RCSR''
1199  001724  030262                DT36                    ;$TESTN,ERRPC,RCSR
1200  001726  000000                0
1201
1202  001730  023236                EM56                    ;''SLU2 RECEIVER DONE NEVER SET''
1203  001732  027456                DH6                     ;''TEST#  ERR PC  RCSR''
```

```
1204  001734  030262                  DT36                    ;$TESTN,ERRPC,RCSR
1205  001736  000000                  0
1206
1207  001740  023273                  EM57                    ;''RESET DID NOT CLEAR SLU2 RCVR DONE''
1208  001742  027456                  DH6                     ;''TEST#  ERR PC  RCSR''
1209  001744  030262                  DT36                    ;$TESTN,ERRPC,RCSR
1210  001746  000000                  0
1211
1212  001750  023336                  EM60                    ;''READING SLU2 RBUF DID NOT CLEAR RCVR DONE''
1213  001752  027456                  DH6                     ;''TEST#  ERR PC  RCSR''
1214  001754  030262                  DT36                    ;$TESTN,ERRPC,RCSR
1215  001756  000000                  0
1216
1217  001760  023410                  EM61                    ;LKS DID NOT RETURN SSYNC
1218  001762  027555                  DH61                    ;''TEST#  ERR PC  LKS''
1219  001764  030302                  DT61                    ;$TESTN,ERRPC,LKS
1220  001766  000000                  0
1221
1222  001770  023441                  EM62                    ;''BIT6 OF LKS NOT CLEAR AFTER RESET''
1223  001772  027555                  DH61                    ;''TEST# ERR PC  LKS''
1224  001774  030302                  DT61                    ;$TESTN,$ERRPC,LKS
1225  001776  000000                  0
1226
1227  002000  023503                  EM63                    ;''LKS INTERRUPT WITH PRIORITY 7''
1228  002002  027555                  DH61                    ;''TEST# ERR PC  LKS''
1229  002004  030302                  DT61                    ;$TESTN,$ERRPC,LKS
1230  002006  000000                  0
1231
1232  002010  023541                  EM64                    ;''CAN NOT SET BIT6 OF LKS''
1233  002012  027555                  DH61                    ;''TEST# ERR PC  LKS''
1234  002014  030302                  DT61                    ;$TESTN,$ERRPC,LKS
1235  002016  000000                  0
1236
1237  002020  023571                  EM65                    ;''CAN NOT CLEAR BIT6 OF LKS''
1238  002022  027555                  DH61                    ;''TEST# ERR PC  LKS''
1239  002024  030302                  DT61                    ;$TESTN,$ERRPC,LKS
1240  002026  000000                  0
1241
1242  002030  023623                  EM66                    ;''RESET DID NOT CLEAR BIT6 OF LKS''
1243  002032  027555                  DH61                    ;''TEST# ERR PC  LKS''
1244  002034  030302                  DT61                    ;$TESTN,$ERRPC,LKS
1245  002036  000000                  0
1246
1247  002040  023663                  EM67                    ;''BIT7 OF LKS NOT SET AFTER RESET2
1248  002042  027555                  DH61                    ;''TEST# ERR PC  LKS''
1249  002044  030302                  DT61                    ;$TESTN,$ERRPC,LKS
1250  002046  000000                  0
1251
1252  002050  023723                  EM70                    ;''CAN NOT CLEAR BIT7 OF LKS''
1253  002052  027555                  DH61                    ;''TEST# ERR PC  LKS''
1254  002054  030302                  DT61                    ;$TESTN,$ERRPC,LKS
1255  002056  000000                  0
1256
1257  002060  023755                  EM71                    ;''BIT7 OF LKS DOES NOT SET''
1258  002062  027555                  DH61                    ;''TEST# ERR PC  LKS''
1259  002064  030302                  DT61                    ;$TESTN,$ERRPC,LKS
```

```
1260  002066  000000                    0
1261
1262  002070  024006                    EM72                    ;WRITING TO ONE INTERNAL ADDRESS MODIFIED ANOTHER
1263  002072  027601                    DH72                    ;''TEST# ERR PC  GOOD  BAD  GDDATA BDDATA
1264  002074  030312                    DT72                    ;$TESTN,$ERRPC,$GDADR,$BDADR,$GDDAT,$BDDAT
1265  002076  000000                    0
1266
1267  002100  000004                    .BLKW   4               ;THE LAST IN A LONG LINE OF PACIFICATION
1268
1269
1270  002110  024067                    EM74                    ;''SLU1 XMIT INTERRUPTS WHEN DISABLED''
1271  002112  027431                    DH5                     ;''TEST# ERR PC  TCSR''
1272  002114  030212                    DT5                     ;$TESTN,$ERRPC,CTCSR
1273  002116  000000                    0
1274
1275  002120  024132                    EM75                    ;''SLU1 XMIT DID NOT INTERRUPT''
1276  002122  027431                    DH5                     ;''TEST# ERR PC  TCSR''
1277  002124  030212                    DT5                     ;$TESTN,$ERRPC,CTCSR
1278  002126  000000                    0
1279
1280  002130  024166                    EM76                    ;''SLU1 XMIT INTERRUPT AT PRIORITY 7''
1281  002132  027431                    DH5                     ;''TEST# ERR PC  TCSR''
1282  002134  030212                    DT5                     ;$TESTN,$ERRPC,CTCSR
1283  002136  000000                    0
1284
1285  002140  024230                    EM77                    ;''SLU1 XMIT INTERRUPTS WITH ENABLE CLEAR''
1286  002142  027431                    DH5                     ;''TEST# ERR PC  TCSR''
1287  002144  030212                    DT5                     ;$TESTN,$ERRPC,CTCSR
1288  002146  000000                    0
1289
1290  002150  024277                    EM100                   ;''SLU1 XMIT DID NOT INTERRUPT''
1291  002152  027431                    DH5                     ;''TEST# ERR PC  TCSR''
1292  002154  030212                    DT5                     ;$TESTN,$ERRPC,CTCSR
1293  002156  000000                    0
1294
1295  002160  024333                    EM101                   ;''SLU1 XMIT RE-INTERRUPTED''
1296  002162  027431                    DH5                     ;''TEST# ERR PC  TCSR''
1297  002164  030212                    DT5                     ;$TESTN,$ERRPC,CTCSR
1298  002166  000000                    0
1299
1300  002170  024364                    EM102                   ;''LOADING SLU1 TBUF DID NOT CLEAR INTERRUPT''
1301  002172  027431                    DH5                     ;''TEST# ERR PC  TCSR''
1302  002174  030212                    DT5                     ;$TESTN,$ERRPC,CTCSR
1303  002176  000000                    0
1304
1305  002200  024436                    EM103                   ;''SLU1 RCVR INTERRUPTS WITH ENABLE CLEAR''
1306  002202  027456                    DH6                     ;''TEST# ERR PC  RCSR''
1307  002204  030222                    DT6                     ;$TESTN,$ERRPC,CRCSR
1308  002206  000000                    0
1309
1310  002210  024505                    EM104                   ;''SLU1 RCVR DID NOT INTERRUPT''
1311  002212  027456                    DH6                     ;''TEST# ERR PC  RCSR''
1312  002214  030222                    DT6                     ;$TESTN,$ERRPC,CRCSR
1313  002216  000000                    0
1314
1315  002220  024541                    EM105                   ;''SLU1 RCVR INTERRUPTS AT PRIORITY 7''
```

```
1316  002222  027456          DH6                     ;''TEST# ERR PC  RCSR''
1317  002224  030222          DT6                     ;$TESTN,$ERRPC,CRCSR
1318  002226  000000          0
1319
1320  002230  024604          EM106                   ;''SLU1 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR''
1321  002232  027456          DH6                     ;''TEST# ERR PC  RCSR''
1322  002234  030222          DT6                     ;$TESTN,$ERRPC,CRCSR
1323  002236  000000          0
1324
1325  002240  024665          EM107                   ;''SLU1 RCVR DID NOT INTERRUPT''
1326  002242  027456          DH6                     ;''TEST# ERR PC  RCSR''
1327  002244  030222          DT6                     ;$TESTN,$ERRPC,CRCSR
1328  002246  000000          0
1329
1330  002250  024721          EM110                   ;''SLU1 RECEIVER RE-INTERRUPTED''
1331  002252  027456          DH6                     ;''TEST# ERR PC  RCSR''
1332  002254  030222          DT6                     ;$TESTN,$ERRPC,CRCSR
1333  002256  000000          0
1334
1335  002260  024756          EM111                   ;''SLU1 READING RBUF DID NOT CLEAR INTERRUPT''
1336  002262  027456          DH6                     ;''TEST# ERR PC  RCSR''
1337  002264  030222          DT6                     ;$TESTN,$ERRPC,CRCSR
1338  002266  000000          0
1339
1340  002270  025030          EM112                   ;''RESET DID NOT CLEAR SLU1 RCVR INTERRUPT''
1341  002272  027456          DH6                     ;''TEST# ERR PC  RCSR''
1342  002274  030222          DT6                     ;$TESTN,$ERRPC,CRCSR
1343  002276  000000          0
1344
1345  002300  025100          EM113                   ;''SLU1 'OR' FLAG DID NOT SET''
1346  002302  027456          DH6                     ;''TEST# ERR PC  RCSR''
1347  002304  030222          DT6                     ;$TESTN,$ERRPC,CRCSR
1348  002306  000000          0
1349
1350  002310  025133          EM114                   ;''SLU1 'ERROR' NOT SET WITH 'OR' FLAG''
1351  002312  027456          DH6                     ;''TEST# ERR PC  RCSR''
1352  002314  030222          DT6                     ;$TESTN,$ERRPC,CRCSR
1353  002316  000000          0
1354
1355  002320  025177          EM115                   ;''DATA COMPARE ERROR''
1356  002322  027657          DH115                   ;''TEST# ERR PC  CRCSR  GOOD  BAD''
1357  002324  030330          DT115                   ;$TESTN,$ERRPC,CRCSR,GD,BD
1358  002326  000000          0
1359
1360  002330  025222          EM116                   ;''SLU2 XMIT INTERRUPTS WHEN DISABLED''
1361  002332  027431          DH5                     ;''TEST# ERR PC  TCSR''
1362  002334  030242          DT31                    ;$TESTN,$ERRPC,TCSR
1363  002336  000000          0
1364
1365  002340  025265          EM117                   ;''SLU2 XMIT DID NOT INTERRUPT''
1366  002342  027431          DH5                     ;''TEST# ERR PC  TCSR''
1367  002344  030242          DT31                    ;$TESTN,$ERRPC,TCSR
1368  002346  000000          0
1369
1370  002350  025321          EM120                   ;''SLU2 XMIT INTERRUPT AT PRIORITY 7''
1371  002352  027431          DH5                     ;''TEST# ERR PC  TCSR''
```

ERROR POINTER TABLE

```
1372  002354  030242          DT31                    ;$TESTN,$ERRPC,TCSR
1373  002356  000000          0
1374
1375  002360  025363          EM121                   ;''SLU2 XMIT INTERRUPTS WITH ENABLE CLEAR''
1376  002362  027431          DH5                     ;''TEST# ERR PC  TCSR''
1377  002364  030242          DT31                    ;$TESTN,$ERRPC,TCSR
1378  002366  000000          0
1379
1380  002370  025432          EM122                   ;''SLU2 XMIT DID NOT INTERRUPT''
1381  002372  027431          DH5                     ;''TEST# ERR PC  TCSR''
1382  002374  030242          DT31                    ;$TESTN,$ERRPC,TCSR
1383  002376  000000          0
1384
1385  002400  025466          EM123                   ;''SLU2 XMIT RE-INTERRUPTED''
1386  002402  027431          DH5                     ;''TEST# ERR PC  TCSR''
1387  002404  030242          DT31                    ;$TESTN,$ERRPC,TCSR
1388  002406  000000          0
1389
1390  002410  025517          EM124                   ;''LOADING SLU2 TBUF DID NOT CLEAR INTERRUPT''
1391  002412  027431          DH5                     ;''TEST# ERR PC  TCSR''
1392  002414  030242          DT31                    ;$TESTN,$ERRPC,TCSR
1393  002416  000000          0
1394
1395  002420  025571          EM125                   ;''SLU2 RCVR INTERRUPTS WITH ENABLE CLEAR''
1396  002422  027456          DH6                     ;''TEST# ERR PC  RCSR''
1397  002424  030262          DT36                    ;$TESTN,$ERRPC,RCSR
1398  002426  000000          0
1399
1400  002430  025640          EM126                   ;''SLU2 RCVR DID NOT INTERRUPT''
1401  002432  027456          DH6                     ;''TEST# ERR PC  RCSR''
1402  002434  030262          DT36                    ;$TESTN,$ERRPC,RCSR
1403  002436  000000          0
1404
1405  002440  025674          EM127                   ;''SLU2 RCVR INTERRUPTS AT PRIORITY 7''
1406  002442  027456          DH6                     ;''TEST# ERR PC  RCSR''
1407  002444  030262          DT36                    ;$TESTN,$ERRPC,RCSR
1408  002446  000000          0
1409
1410  002450  025737          EM130                   ;''SLU2 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR''
1411  002452  027456          DH6                     ;''TEST# ERR PC  RCSR''
1412  002454  030262          DT36                    ;$TESTN,$ERRPC,RCSR
1413  002456  000000          0
1414
1415  002460  026020          EM131                   ;''SLU2 RCVR DID NOT INTERRUPT''
1416  002462  027456          DH6                     ;''TEST# ERR PC  RCSR''
1417  002464  030262          DT36                    ;$TESTN,$ERRPC,RCSR
1418  002466  000000          0
1419
1420  002470  026054          EM132                   ;''SLU2 RECEIVER RE-INTERRUPTED''
1421  002472  027456          DH6                     ;''TEST# ERR PC  RCSR''
1422  002474  030262          DT36                    ;$TESTN,$ERRPC,RCSR
1423  002476  000000          0
1424
1425  002500  026111          EM133                   ;''SLU2 READING RBUF DID NOT CLEAR INTERRUPT''
1426  002502  027456          DH6                     ;''TEST# ERR PC  RCSR''
1427  002504  030262          DT36                    ;$TESTN,$ERRPC,RCSR
```

```
1428  002506  000000                    0
1429
1430  002510  026163                    EM134           ;'RESET DID NOT CLEAR SLU2 RCVR INTERRUPT'
1431  002512  027456                    DH6             ;'TEST# ERR PC  RCSR'
1432  002514  030262                    DT36            ;$TESTN,$ERRPC,RCSR
1433  002516  000000                    0
1434
1435  002520  026233                    EM135           ;'SLU2 'OR' FLAG DID NOT SET'
1436  002522  027456                    DH6             ;'TEST# ERR PC  RCSR'
1437  002524  030262                    DT36            ;$TESTN,$ERRPC,RCSR
1438  002526  000000                    0
1439
1440  002530  026266                    EM136           ;'SLU2 'ERROR' NOT SET WITH 'OR' FLAG'
1441  002532  027456                    DH6             ;'TEST# ERR PC  RCSR'
1442  002534  030262                    DT36            ;$TESTN,$ERRPC,RCSR
1443  002536  000000                    0
1444
1445  002540  026332                    EM137           ;'SLU2 BREAK DID NOT TRANSMIT ALL ZEROES'
1446  002542  027723                    DH137           ;'TEST# ERR PC  RCSR  DATA'
1447  002544  030344                    DT137           ;$TESTN,$ERRPC,RCSR,$BDDAT
1448  002546  000000                    0
1449
1450  002550  026401                    EM140           ;'BREAK DID NOT SET FRAMING ERROR'
1451  002552  027456                    DH6             ;'TEST# ERR PC  RCSR'
1452  002554  030222                    DT6             ;$TESTN,$ERRPC,RCSR
1453  002556  000000                    0
1454
1455  002560  026441                    EM141           ;'SLU2 'ERROR' NOT SET WITH 'FR' FLAG'
1456  002562  027456                    DH6             ;'TEST# ERR PC  RCSR'
1457  002564  030262                    DT36            ;$TESTN,$ERRPC,RCSR
1458  002566  000000                    0
1459
1460  002570  026505                    EM142           ;'DATA COMPARE ERROR WITH CABLE'
1461  002572  027760                    DH142           ;'TEST# ERR PC  RCSR    GOOD    BAD'
1462  002574  030356                    DT142           ;$TESTN,$ERRPC,RCSR,GD,BD
1463  002576  000000                    0
1464
1465  002600  026543                    EM143           ;'RTC INTERRUPT AT PRIORITY 7'
1466  002602  027555                    DH61            ;'TEST# ERR PC  LKS'
1467  002604  030302                    DT61            ;$TESTN,$ERRPC,LKS
1468  002606  000000                    0
1469
1470  002610  026577                    EM144           ;'RTC INTERRUPTS WHEN DISABLED'
1471  002612  027555                    DH61            ;'TEST# ERR PC  LKS'
1472  002614  030302                    DT61            ;$TESTN,$ERRPC,LKS
1473  002616  000000                    0
1474
1475  002620  026634                    EM145           ;'RTC INTERRUPT DID NOT OCCUR'
1476  002622  027555                    DH61            ;'TEST# ERR PC  LKS'
1477  002624  030302                    DT61            ;$TESTN,$ERRPC,LKS
1478  002626  000000                    0
1479
1480  002630  026670                    EM146           ;'RTC INTERRUPT DID NOT OCCUR'
1481  002632  027555                    DH61            ;'TEST# ERR PC  LKS'
1482  002634  030302                    DT61            ;$TESTN,$ERRPC,LKS
1483  002636  000000                    0
```

```
1484
1485  002640  026724          EM147           ;'RTC DOUBLE INTERRUPT'
1486  002642  027555          DH61            ;'TEST# ERR PC  LKS'
1487  002644  030302          DT61            ;$TESTN,$ERRPC,LKS
1488  002646  000000          0
1489
1490  002650  026751          EM150           ;'RESET DID NOT CLEAR RTC INTERRUPT'
1491  002652  027555          DH61            ;'TEST# ERR PC  LKS'
1492  002654  030302          DT61            ;$TESTN,$ERRPC,LKS
1493  002656  000000          0
1494
1495  002660  027013          EM151           ;'RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS'
1496  002662  027555          DH61            ;'TEST# ERR PC  LKS'
1497  002664  030302          DT61            ;$TESTN,$ERRPC,LKS
1498  002666  000000          0
1499
1500  002670  027070          EM152           ;'CLOCK REPEATABILITY ERROR'
1501  002672  030014          DH152           ;'TEST# ERR PC  LKS    CNT1    CNT2'
1502  002674  030370          DT152           ;'$TESTN,$ERRPC,LKS,FIRST,SECND
1503  002676  000000          0
1504
1505  002700  027122          EM153           ;'SLU1 RECEIVER STATUS ERROR'
1506  002702  027657          DH115           ;'TEST#  ERR PC  CRCSR  GOOD  BAD'
1507  002704  030330          DT115           ;$TESTN,$ERRPC,CRCSR,$GDDTA,$BDDAT
1508  002706  000000          0
1509
1510  002710  027155          EM154           ;SLU2 RECEIVER STATUS ERROR
1511  002712  027760          DH142           ;'TESTN  ERR PC  RCSR  GOOD  BAD'
1512  002714  030356          DT142           ;$TESTN,$ERRPC,RCSR,$GDDAT,$BDDAT
1513  002716  000000          0
1514
1515  002720  027210          EM155           ;'INCORRECT RECEIVE COUNT SLU1
1516  002722  030061          DH155           ;'TEST# ERR PC  RCSR    TRANS   RCV'
1517  002724  030404          DT155           ;$TESTN,$ERRPC,CRCSR,XMTCT1,RCVCT1
1518  002726  000000          0
1519
1520  002730  027245          EM156           ;'SLU1 DATA COMPARE ERROR IN EXERCISER'
1521  002732  027657          DH115           ;'TEST# ERR PC  CRCSR   GOOD    BAD'
1522  002734  030330          DT115           ;$TESTN,$ERRPC,CRCSR,GD,BD
1523  002736  000000          0
1524
1525  002740  027312          EM157           ;'INCORRECT RECEIVE COUNT SLU2
1526  002742  030061          DH155           ;'TEST# ERR PC  RCSR    TRANS   RCV'
1527  002744  030420          DT157           ;$TESTN,$ERRPC,RCSR,XMTCT2,RCVCT2
1528  002746  000000          0
1529
1530  002750  027347          EM160           ;'SLU2 DATA COMPARE ERROR IN EXERCISER'
1531  002752  027760          DH142           ;'TEST# ERR PC  RCSR    GOOD    BAD'
1532  002754  030356          DT142           ;$TESTN,$ERRPC,RCSR,GD,BD
1533  002756  000000          0
1534
1535  002760  027414          EM161           ;'TRAP CATCHER'
1536  002762  030125          DH161           ;'TEST#   ERR PC  OLDPC  TRAP ADR'
1537  002764  030434          DT161           ;$TESTN,$ERRPC,OLDPC,BDVECT
1538  002766  000000          0
1539
```

```
 1540
 1541
 1542  002770  176500                ;REGISTER ADDRESSES OF INTERNAL ON BOARD OPTIONS
 1543  002772  176502
 1544  002774  176504
 1545  002776  176506
 1546  003000  177560
 1547  003002  177562
 1548  003004  177564
 1549  003006  177566
 1550  003010  177546
 1551
 1552
 1553
 1554
 1555  003012  000300                ;VECTOR ADDRESSES FOR ON BOARD OPTIONS
 1556  003014  000302
 1557  003016  000304
 1558  003020  000306
 1559  003022  000060
 1560  003024  000062
 1561  003026  000064
 1562  003030  000066
 1563  003032  000100
 1564  003034  000102
 1565
 1566  003036  005037  001102
 1567  003042  005037  001100
 1568  003046  005037  001104
 1569  003052  005037  001156
 1570  003056
 1571                              .SBTTL  INITIALIZE THE COMMON TAGS
 1572                              ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
 1573  003056  012706  001000
 1574  003062  005026
 1575  003064  022706  001040
 1576  003070  001374
 1577  003072  012706  001000
 1578                              ;;INITIALIZE A FEW VECTORS
 1579  003076  012737  015614  000020
 1580  003104  012737  000340  000022
 1581  003112  012737  015120  000030
 1582  003120  012737  000340  000032
 1583  003126  012737  020026  000034
 1584  003134  012737  000340  000036
 1585  003142  012737  015436  000024
 1586  003150  012737  000340  000026
 1587  003156  013737  014770  014762
 1588  003164  005037  001072
 1589  003170  112737  000001  001015
 1590  003176  012737  003176  001006
 1591  003204  012737  003204  001010
 1592                              ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
 1593                              ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
 1594  003212  013746  000004
 1595  003216  012737  003252  000004

RCSR:   .WORD   176500          ;SLU2 COMMAND/STATUS REGISTER
RBUF:   .WORD   176502          ;SLU2 RECEIVER BUFFER
TCSR:   .WORD   176504          ;SLU2 TRANSMITTER COMMAND/STATUS REGISTER
TBUF:   .WORD   176506          ;SLU2 TRANSMITTER BUFFER
CRCSR:  177560                  ;SLU1 RECEIVER COMMAND/STATUS REGISTER
CRBUF:  177562                  ;SLU1 RECEIVER BUFFER
CTCSR:  177564                  ;SLU1 TRANSMITTER COMMAND/STATUS REGISTER
CTBUF:  177566                  ;SLU1 TRANSMITTER BUFFER
LKS:    .WORD   177546          ;LTC COMMAND/STATUS REGISTER

RVECT:  .WORD   300
RPSW:   .WORD   302
TVECT:  .WORD   304
TPSW:   .WORD   306
CRVECT: 60                      ;RECEIVER INTERRUPT VECTOR
CRPSW:  62
CTVECT: 64                      ;TRANSMITTER INTERRUPT VECTOR
CTPSW:  66
RTCVT:  .WORD   100
RTCPSW: .WORD   102

START:  CLR     $FATAL          ;CLEAR ERROR NO.
        CLR     $MSGTYP         ;CLEAR MESSAGE TYPE
        CLR     $TESTN          ;CLEAR TEST NO.
        CLR     $DEVM           ;CLEAR FLAGS INDICATING DEVICES UNDER TEST
1$:
        MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
        CLR     (R6)+           ;;CLEAR MEMORY LOCATION
        CMP     #SWR,R6 ;;DONE?
        BNE     .-6             ;;LOOP BACK IF NO
        MOV     #1000,SP                ;;SETUP THE STACK POINTER
        MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
        MOV     #340,@#IOTVEC+2 ;;LEVEL 7
        MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
        MOV     #340,@#EMTVEC+2 ;;LEVEL 7
        MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
        MOV     #340,@#TRAPVEC+2;LEVEL 7
        MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
        MOV     #340,@#PWRVEC+2 ;;LEVEL 7
        MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
        CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
        MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
        MOV     #..,$LPADR      ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
        MOV     #..,$LPERR      ;;SETUP THE ERROR LOOP ADDRESS
        MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
        MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
```

```
1596  003224  012737  177570  001040          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1597  003232  012737  177570  001042          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1598  003240  022777  177777  175572          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1599  003246  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1600                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
1601  003250  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
1602  003252  012716  003260          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1603  003256  000002                          RTI
1604  003260  012737  000176  001040  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1605  003266  012737  000174  001042          MOV     #D,SPREG,DISPLAY
1606  003274  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1607
1608  003300  005037  001106                  CLR     $PASS           ;;CLEAR PASS COUNT
1609  003304  132737  000200  001121          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1610  003312  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
1611  003314  012737  001122  001040          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1612  003322                          67$:
                                    .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1613
1614  003322  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1615  003326  001012                          BNE     68$             ;;BRANCH IF YES
1616  003330  123727  001120  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
1617  003336  001406                          BEQ     68$             ;;BRANCH IF YES
1618  003340  023727  001040  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1619  003346  001005                          BNE     69$             ;;BRANCH IF NO
1620  003350  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
1621  003352  000403                          BR      69$
1622  003354  112737  000001  001034  68$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1623  003362                          69$:
1624  003362  032777  000060  175450          BIT     #BIT4.BIT5,@SWR ;IS SLU1 TO BE TESTED
1625  003370  001003                          BNE     2$              ;IF EITHER BIT IS SET THEN DON'T TEST IT
1626  003372  052737  000001  001156          BIS     #BIT0,$DEVM     ;SET DEVICE FLAG TO TEST SLU1
1627  003400  032777  000050  175432  2$:     BIT     #BIT3!BIT5,@SWR ;IS SLU2 TO BE TESTED
1628  003406  001003                          BNE     3$              ;IF EITHER BIT IS SET THEN DON'T TEST IT
1629  003410  052737  000002  001156          BIS     #BIT1,$DEVM     ;SET DEVICE FLAG TO TEST SLU2
1630  003416  032777  000100  175414  3$:     BIT     #BIT6,@SWR      ;IS LTC TO BE TESTED
1631  003424  001003                          BNE     4$              ;IF BIT IS SET THEN DON'T TEST IT.
1632  003426  052737  000004  001156          BIS     #BIT2,$DEVM     ;SET DEVICE FLAG TO TEST LTC
1633  003434  032737  000001  001156  4$:     BIT     #BIT0,$DEVM     ;IS SLU1 UNDER TEST
1634  003442  001002                          BNE     TST1            ;IF YES TEST XMIT REG. BEFORE TYPING TITLE
1635  003444  000137  003700                  JMP     ID              ;IF NO SKIP TESTS AND TYPE IT NOW
```

```
1636
1637
1638                                    ;;*********************************************************************
1639                                    ;*TEST 1         TEST ABILITY TO ACCESS SLU1 TCSR
1640                                    ;;*********************************************************************
1641   003450  000004                   TST1:   SCOPE
1642   003452  012737  000001  001104           MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1643   003460  013703  000004                   MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1644   003464  012737  003500  000004           MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1645   003472  005777  177306                   TST     @C~CSR          ;REFERENCE THE XMIT COMMAND/STATUS REG.
1646   003476  000405                           BR      2$              ; GO TO END OF TEST
1647   003500  022626                   1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1648   003502  004737  016004                   JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
1649   003506  000001                           1                       ;;THE ERROR NUMBER (FROM APT LIST)
1650   003510  000000                           HALT
1651   003512  010337  000004           2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1652
1653
1654
1655                                    ;;*********************************************************************
1656                                    ;*TEST 2         TEST ABILITY TO ACCESS SLU1 TBUF
1657                                    ;;*********************************************************************
1658   003516  000004                   TST2:   SCOPE
1659   003520  012737  000002  001104           MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1660   003526  013703  000004                   MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1661   003532  012737  003546  000004           MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1662   003540  005777  177242                   TST     @CTBUF          ;REFERENCE THE XMIT BUFFER
1663   003544  000405                           BR      2$              ;GO TO END OF  TEST
1664   003546  022626                   1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1665   003550  004737  016004                   JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
1666   003554  000002                           2                       ;;THE ERROR NUMBER (FROM APT LIST)
1667   003556  000000                           HALT
1668   003560  010337  000004           2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1669
1670
1671                                    ;;*********************************************************************
1672                                    ;*TEST 3         TEST SLU1 TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
1673                                    ;;*********************************************************************
1674   003564  000004                   TST3:   SCOPE
1675   003566  012737  000003  001104           MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1676   003574  032737  000001  001120           BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
1677   003602  001405                           BEQ     70$             ;IF NO THEN DO TEST
1678   003604  005737  001106                   TST     @#$PASS         ;IS THIS FIRST PASS
1679   003610  001402                           BEQ     70$             ;IF YES THEN DO THIS SERIES OF TESTS
1680   003612  000137  004726                   JMP     SLU2RT          ;IF NO THEN SKIP THIS SERIES OF TESTS
1681   003616  005077  177164           70$:    CLR     @CTBUF          ;LOAD XBUF
1682   003622  105777  177156                   TSTB    @CTCSR          ;CHECK DONE
1683   003626  100011                           BPL     1$              ;BR IF CLEAR
1684                                                                    ;FILL SECOND BUFFER, BECAUSE REFRESH COULD CAUSE
1685                                                                    ;   FIRST TEST TO FAIL
1686   003630  005077  177152                   CLR     @CTBUF          ;FILL DOUBLE BUFFER
1687   003634  105777  177144                   TSTB    @CTCSR          ;CHECK DONE
1688   003640  100004                           BPL     1$              ;BR IF CLEAR
1689   003642  004737  016004                   JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
1690   003646  000003                           3                       ;;THE ERROR NUMBER (FROM APT LIST)
1691   003650  000000                           HALT                    ;TCSR "DONE" NOT CLEARED WITH TBUF FULL
```

```
1692  003652  005000                  1$:     CLR     R0              ;CLEAR TIMER
1693  003654  105777  177124          2$:     TSTB    @CTCSR          ;CHECK FOR XMIT DONE
1694  003660  100407                          BMI     ID              ;IF DONE SETS, BR TO END OF TEST
1695  003662  005200                          INC     R0              ;INCREMENT TIMER
1696  003664  001373                          BNE     2$              ;BR IF TIMER NOT DONE
1697  003666  004737  016004                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
1698  003672  000004                          4                               ;;THE ERROR NUMBER (FROM APT LIST)
1699  003674  000000                          HALT
1700  003676  000416                          BR      TST4            ;BR TO NEXT TEST, AND SKIP THE TYPEOUT THAT FOLLOWS
1701                                                                  ;  BECAUSE OF THIS FAILURE
1702
1703  003700  023737  000042  000046  ID:     CMP     @#42,@#46       ;UNDER ACT11?
1704  003706  001412                          BEQ     6$              ;IF YES, SKIP IDENT. TYPEOUT
1705  003710  005737  001106                  TST     $PASS           ;IS THIS THE FIRST PASS?
1706  003714  001007                          BNE     6$              ;IF NOT BR TO NEXT TEST & SKIP THE IDENTIFICATION TYPEOU
1707  003716  005737  001110                  TST     $DEVCT          ;IS THIS THE FIRST SUBPASS?
1708  003722  001004                          BNE     6$              ;IF NOT, BRTO NEXT TEST
1709  003724  104401                          TYPE                    ;TYPE PROGRAM INDENTIFICATION
1710  003726  030176                          M1
1711  003730  104401                          TYPE                    ;TYPE NUMBER OF DEVICES UNDER TEST
1712  003732  030210                          M2
1713  003734                          6$:
1714
1715                                  ;;**********************************************************************
1716                                  ;*TEST 4         TEST THAT SLU1 TCSR 'DONE'' SETS WITH RESET
1717                                  ;;**********************************************************************
1718  003734  000004                  TST4:   SCOPE
1719  003736  012737  000004  001104          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1720  003744  032737  000001  001156          BIT     #BIT0,@#$DEVM   ;DO THESE TESTS FOR THIS DEVICE?
1721  003752  001002                          BNE     99$             ;F YES CONTINUE WITH TESTS
1722  003754  000137  004726                  JMP     SLU2RT          ;IF NO GO TO START OF NEXT SET OF TESTS.
1723  003760                          99$:
1724  003760  005077  177022                  CLR     @CTBUF          ;LOAD TRANSMIT BUFFER
1725  003764  105777  177014          1$:     TSTB    @CTCSR          ;WAIT FOR DONE
1726  003770  100375                          BPL     1$
1727  003772  005077  177010                  CLR     @CTBUF          ;LOAD SECOND BUFFER
1728  003776  000240                          NOP
1729  004000  000005                          RESET                   ;CLEAR DONE WITH RESET
1730  004002  105777  176776                  TSTB    @CTCSR          ;CHECK FOR DONE SET
1731  004006  100401                          BMI     TST5            ;BR TO NEXT TEST IF DONE SET
1732
1733  004010  104005                          ERROR   5               ;TCSR 'DONE'' DOES NOT SET WITH RESET
1734
1735
1736
1737                                  ;;**********************************************************************
1738                                  ;*TEST 5         TEST ABILITY TO ACCESS SLU1 RCSR
1739                                  ;;**********************************************************************
1740  004012  000004                  TST5:   SCOPE
1741  004014  012737  000005  001104          MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1742  004022  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1743  004026  012737  004042  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1744  004034  005777  176740                  TST     @RCSR           ;ACCESS RCSR
1745  004040  000402                          BR      2$              ;BR TO END OF TEST
1746
1747  004042  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
```

```
                                        ERROR   6                       ;CAN NOT ACCESS RCSR
004046  010337  000004         2$:     MOV     R3,@#4                  ;RESTORE TIMEOUT VECTOR


                                ;;**********************************************************************
                                ;*TEST 6         TEST ABILITY TO ACCESS SLU1 RBUF
                                ;;**********************************************************************
004052  000004                  TST6:   SCOPE
004054  012737  000006  001104          MOV     #6,$TESTN               ;;SET TEST NUMBER IN APT MAIL BOX
004062  013703  000004                  MOV     @#4,R3                  ;SAVE TIMEOUT VECTOR
004066  012737  004102  000004          MOV     #1$,@#4                 ;SET UP TIMEOUT VECTOR
004074  005777  176702                  TST     @CRBUF                  ;ACCESS RBUF
004100  000402                         BR      2$                      ;BR TO END OF TEST

004102  022626                  1$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER TIMEOUT
004104  104007                          ERROR   7                       ;CAN NOT ACCESS RBUF
004106  010337  000004         2$:     MOV     R3,@#4                  ;RESTORE TIMEOUT VECTOR


                                ;;**********************************************************************
                                ;*TEST 7         TEST THAT SLU1 BIT2(MAINT. BIT) CAN BE SET & RESET
                                ;;**********************************************************************
004112  000004                  TST7:   SCOPE
004114  012737  000007  001104          MOV     #7,$TESTN               ;;SET TEST NUMBER IN APT MAIL BOX
004122  042777  000004  176654          BIC     #BIT2,@CTCSR            ;MAKE SURE BIT UNDER TEST IS INITIALIZED
004130  032777  000004  176646          BIT     #BIT2,@CTCSR            ;TEST FOR BIT2 OF TCSR CLEAR
004136  001404                         BEQ     1$                      ;BR IF CLEAR
004140  004737  016004                  JSR     PC,$ATY4                        ;;ONLY REPORT A FATAL ERROR
004144  000010                         10                                      ;;THE ERROR NUMBER (FROM APT LIST)
004146  000000                         HALT
004150  052777  000004  176626  1$:     BIS     #BIT2,@CTCSR            ;SET BIT2 OF TCSR
004156  032777  000004  176620          BIT     #BIT2,@CTCSR            ;TEST FOR BIT2 SET
004164  001001                         BNE     2$                      ;BR IF SET
004166  104011                         ERROR   11                      ;BIT2 OF TCSR WILL NOT SET
004170  042777  000004  176606  2$:     BIC     #BIT2,@CTCSR            ;CLEAR BIT2 OF TCSR
004176  032777  000004  176600          BIT     #BIT2,@CTCSR            ;TEST BIT2 CLEAR
004204  001404                         BEQ     3$                      ;BR IF CLEAR
004206  004737  016004                  JSR     PC,$ATY4                        ;;ONLY REPORT A FATAL ERROR
004212  000012                         12                                      ;;THE ERROR NUMBER (FROM APT LIST)
004214  000000                         HALT                            ;BIT0 OF TCSR WILL NOT CLEAR
004216  052777  000004  176560  3$:     BIS     #BIT2,@CTCSR            ;SET BIT2 OF TCSR
004224  000005                         RESET                           ;CLEAR BIT2 WITH RESET
004226  032777  000004  176550          BIT     #BIT2,@CTCSR            ;TEST FOR BIT2 CLEAR
004234  001404                         BEQ     TST10                   ;IF CLEAR, GO TO NEXT TEST
004236  042777  000004  176540          BIC     #BIT2,@CTCSR            ;CLEAR BIT2, TO PRINT ERROR
004244  104013                         ERROR   13
                                                                        ;RESET DID NOT CLEAR BIT2 OF TCSR


                                ;;**********************************************************************
                                ;*TEST 10        TEST THAT SLU1 BIT6(XMIT INT EN) CAN BE SET & RESET
                                ;;**********************************************************************
004246  000004                  TST10:  SCOPE
004250  012737  000010  001104          MOV     #10,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX
004256  042777  000100  176520          BIC     #BIT6,@CTCSR            ;MAKE SURE BIT UNDER TEST IS INITIALIZED
```

```
1804  004264  017703  176536              MOV     @CTVECT,R3      ;SAVE XMIT VECTOR
1805  004270  012777  004320  176530      MOV     #1$,@CTVECT     ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1806  004276  004737  015056              JSR     PC,WRPSW                ;SET PSW TO PRIORITY=7
1807  004302  000340                      .WORD   340
1808  004304  032777  000100  176472      BIT     #BIT6,@CTCSR    ;TEST BIT6 OF TCSR
1809  004312  001404                      BEQ     2$              ;BR IF ZERO
1810  004314  104014                      ERROR   14
1811                                                              ;BIT6 IN TCSR NOT CLEAR AFTER RESET
1812  004316  000402                      BR      2$
1813
1814  004320  022626          1$:         CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1815  004322  104015                      ERROR   15
1816                                                              ;XMIT INTERRUPT OCCURRED  PRIO=7
1817
1818  004324  052777  000100  176452  2$: BIS     #BIT6,@CTCSR    ;SET BIT6 OF TCSR
1819  004332  032777  000100  176444      BIT     #BIT6,@CTCSR    ;TEST BIT6 OF TCSR
1820  004340  001001                      BNE     3$              ;BR, IF SET
1821
1822  004342  104016                      ERROR   16
1823                                                              ;CANNOT SET BIT6 OF TCSR
1824
1825  004344  042777  000100  176432  3$: BIC     #BIT6,@CTCSR    ;CLEAR BIT6 OF TCSR
1826  004352  032777  000100  176424      BIT     #BIT6,@CTCSR    ;TEST BIT6 OF TCSR
1827  004360  001401                      BEQ     4$              ;BR IF CLEAR
1828  004362  104017                      ERROR   17
1829                                                              ;CANNOT CLEAR BIT6 OF TCSR
1830
1831  004364                          4$:
1832  004364  052777  000100  176412      BIS     #BIT6,@CTCSR    ;SET BIT6 OF TCSR
1833  004372  000005                      RESET                   ;CLEAR BIT6 WITH RESET
1834  004374  032777  000100  176402      BIT     #BIT6,@CTCSR    ;TEST BIT6 OF TCSR
1835  004402  001401                      BEQ     5$              ;BR IF CLEAR
1836
1837  004404  104020                      ERROR   20
1838                                                              ;CANNOT CLEAR BIT6 OF TCSR WITH RESET
1839  004406  010377  176414          5$: MOV     R3,@CTVECT      ;RESTORE XMIT VECTOR
1840
1841
1842                                  ;;*****************************************************************
1843                                  ;*TEST 11       TEST THAT SLU1 BIT6 OF RCSR CAN BE SET & RESET
1844                                  ;;*****************************************************************
1845  004412  000004          TST11:  SCOPE
1846  004414  012737  000011  001104      MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1847  004422  042777  000100  176350      BIC     #BIT6,@CRCSR    ;MAKE SURE BIT UNDER TEST IS INITIALIZED
1848  004430  017703  176366              MOV     @CRVECT,R3      ;SAVE RECEIVE VECTOR
1849  004434  012777  004464  176360      MOV     #1$,@CRVECT     ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1850  004442  004737  015056              JSR     PC,WRPSW        ;SET PSW TO PRIORITY=7
1851  004446  000340                      .WORD   340
1852  004450  032777  000100  176322      BIT     #BIT6,@CRCSR    ;TEST BIT6 OF RCSR
1853  004456  001404                      BEQ     2$
1854  004460  104021                      ERROR   21
1855                                                              ;BIT6 OF RCSR NOT CLEAR AFTER RESET
1856  004462  000402                      BR      2$
1857
1858  004464  022626          1$:         CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1859  004466  104022                      ERROR   22
```

```
                                                        ;RCVR INTERRUPT WITH PRIORITY-7

004470  052777  000100  176302  2$:     BIS     #BIT6,@CRCSR    ;SET BIT6 OF RCSR
004476  032777  000100  176274          BIT     #BIT6,@CRCSR    ;TEST BIT6 OF RCSR
004504  001001                          BNE     3$              ;BR, IF SET

004506  104023                          ERROR   23
                                                        ;CANNOT SET BIT6 OF RCSR

004510  042777  000100  176262  3$:     BIC     #BIT6,@CRCSR    ;CLEAR BIT6 OF RCSR
004516  032777  000100  176254          BIT     #BIT6,@CRCSR    ;TEST BIT6 OF RCSR
004524  001401                          BEQ     4$              ;BR, IF CLEAR

004526  104024                          ERROR   24
                                                        ;CANNOT CLEAR BIT6 OF RCSR

004530                          4$:
004530  052777  000100  176242          BIS     #BIT6,@CRCSR    ;SET BIT6 OF RCSR
004536  000005                          RESET                   ;CLEAR BIT6OF RCSR WITH RESET
004540  032777  000100  176232          BIT     #BIT6,@CRCSR    ;TEST BIT6 OF RCSR
004546  001401                          BEQ     5$              ;BR, IF CLEAR

004550  104025                          ERROR   25
                                                        ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
004552  010377  176244          5$:     MOV     R3,@CRVECT      ;RESTORE RECEIVE VECTOR


                        ;;*******************************************************************
                        ;*TEST 12       TEST THAT SLU1 RCVR DONE (7) SET & CLEAR PROPERLY
                        ;;*******************************************************************
004556  000004          TST12:  SCOPE
004560  012737  000012  001104          MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
004566  052777  000004  176210          BIS     #BIT2,@CTCSR    ;TURN ON INTERNAL MAINTENENCE WRAP AROUND
004574  005000                          CLR     R0              ;CLEAR TIMER
004576  005077  176204                  CLR     @CTBUF          ;LOAD TRANSMIT BUFFER
004602  105777  176172          CWDONE: TSTB    @CRCSR          ;CHECK FOR RECEIVER DONE
004606  100406                          BMI     1$              ;BR, IF DONE
004610  005200                          INC     R0              ;INCREMENT TIMER, IF NOT DONE
004612  001373                          BNE     CWDONE          ;CONTINUE WAIT IF TIME REMAINS
004614  042777  000004  176162          BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
004622  104026                          ERROR   26
                                                        ;RECEIVER DONE NEVER SET

004624  000005          1$:     RESET                   ;CLEAR DONE WITH RESET
004626  105777  176146                  TSTB    @CRCSR          ;CHFCK FOR DONE CLEAR
004632  001404                          BEQ     2$

004634  042777  000004  176142          BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
004642  104027                          ERROR   27
                                                        ;RESET DID NOT CLEAR RCVR DONE

004644  042777  000004  176126  2$:     BIC     #BIT2,@CRCSR    ;CLEAR MAINTENANCE BIT


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

```
1916
1917
1918  004652  000004
1919  004654  012737  000013  001104          ;*TEST 13       TEST SLU1 THAT READING RBUF CLEARS RECEIVER DONE
1920  004662  000005                          ;;**********************************************************************
1921  004664  052777  000004  176112  TST13:  SCOPE
1922  004672  005077  176110                  MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1923  004676  105777  176076                  RESET                   ;CLEAR EVERYTHING
1924  004702  100375                          BIS     #BIT2,@CTCSR    ;SET MAINTENANCE WRAP
1925  004704  017700  176072                  CLR     @CTBUF          ;LOAD TRANSMITTER
1926  004710  042777  000004  176066  1$:     TSTB    @CRCSR          ;WAIT FOR RECEIVER DONE
1927  004716  105777  176056                  BPL     1$
1928  004722  001401                          MOV     @CRBUF,R0       ;READ RECEIVE BUFFER
1929  004724  104030                          BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
1930                                          TSTB    @CRCSR          ;CHECK FOR RECEIVE DONE CLEAR
                                              BEQ     TST14           ;BR, IF CLEAR TO NEXT TEST
                                              ERROR   30
                                                                      ;READING RBUF DID NOT CLEAR RCVR DONE
```

```
 *931  004726                                SLU2RT:

1932
1933                                         ;;*****************************************************************
1934                                         ;*TEST 14       TEST ABILITY TO ACCESS SLU2 TCSR
1935                                         ;;*****************************************************************
1936  004726  000004                        TST14:  SCOPE
1937  004730  012737  000014  001104                MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1938  004736  032737  000002  001156                BIT     #BIT1,@#$DEVM   ;DO THESE TESTS FOR THIS DEVICE?
1939  004744  001002                                BNE     99$             ;F YES CONTINUE WITH TFSTS
1940  004746  000137  006152                        JMP     LTSRT           ;IF NO GO TO START OF NEXT SET OF TESTS.
1941  004752                                99$:
1942  004752  013703  000004                        MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1943  004756  012737  004772  000004                MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1944  004764  005777  176C04                        TST     @TCSR           ;REFERENCE THE XMIT COMMAND/STATUS REG.
1945  004770  000402                                BR      2$              ; GO TO END OF TEST
1946  004772  022626                        1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1947  004774  104031                                ERROR   31              ;XMIT CSR ADDRESS DOES NOT RETURN SSYNC
1948  004776  010337  000004                2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1949
1950
1951
1952                                         ;;*****************************************************************
1953                                         ;*TEST 15       TEST ABILITY TO ACCESS SLU2 TBUF
1954                                         ;;*****************************************************************
1955  005002  000004                        TST15:  SCOPE
1956  005004  012737  000015  001104                MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1957  005012  013703  000004                        MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1958  005016  012737  005032  000004                MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1959  005024  005777  175746                        TST     @TBUF           ;REFERENCE THE XMIT BUFFER
1960  005030  000402                                BR      2$              ;GO TO END OF  TEST
1961  005032  022626                        1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1962  005034  104032                                ERROR   32              ;XMIT BUFFER ADDRESS DOES NOT REUTRN SSYNC
1963  005036  010337  000004                2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1964
1965
1966                                         ;;*****************************************************************
1967                                         ;*TEST 16       TEST SLU2 TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
1968                                         ;;*****************************************************************
1969  005042  000004                        TST16:  SCOPE
1970  005044  012737  000016  001104                MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1971  005052  032737  000001  001120                BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
1972  005060  001403                                BEQ     70$             ;IF NO THEN DO TEST
1973  005062  005737  001106                        TST     @#$PASS         ;IS THIS FIRST PASS
1974  005066  001022                                BNE     TST17           ;IF NO THEN SKIP TO NEXT TEST
1975  005070                                70$:
1976  005070  005077  175702                        CLR     @TBUF           ;LOAD XBUF
1977  005074  105777  175674                        TSTB    @TCSR           ;CHECK DONE
1978  005100  100006                                BPL     1$              ;BR IF CLEAR
1979                                                                        ;FILL SECOND BUFFER, BECAUSE REFRESH COULD CAUSE
1980                                                                        ;   FIRST TEST TO FAIL
1981  005102  005077  175670                        CLR     @TBUF           ;FILL DOUBLE BUFFER
1982  005106  105777  175662                        TSTB    @TCSR           ;CHECK DONE
1983  005112  100001                                BPL     1$              ;BR IF CLEAR
1984  005114  104033                                ERROR   33              ;TCSR "DONE" NOT CLEARED WITH TBUF FULL
1985  005116  005000                        1$:     CLR     R0              ;CLEAR TIMER
1986  005120  105777  175650                2$:     TSTB    @TCSR           ;CHECK FOR XMIT DONE
```

```
1987  005124  100403                        BMI     TST17           ;IF DONE SETS, BR TO NEXT TEST
1988  005126  005200                        INC     R0              ;INCREMENT TIMER
1989  005130  001373                        BNE     2$              ;BR IF TIMER NOT DONE
1990  005132  104034                        ERROR   34              ;XMIT DONE BIT DOES NOT RESET AFTER TRANSMIT
1991
1992
1993                                  ;;*******************************************************************
1994                                  ;*TEST 17        TEST THAT SLU2 TCSR 'DONE'' SETS WITH RESET
1995                                  ;;*******************************************************************
1996  005134  000004                  TST17:  SCOPE
1997  005136  012737  000017  001104          MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1998  005144  032737  000001  001120          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
1999  005152  001403                          BEQ     70$             ;IF NO THEN DO TEST
2000  005154  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
2001  005160  001015                          BNE     TST20           ;IF NO THEN SKIP TO NEXT TEST
2002  005162                          70$:
2003  005162  005077  175610                  CLR     @TBUF           ;LOAD TRANSMIT BUFFER
2004  005166  105777  175602          1$:     TSTB    @1CSR           ;WAIT FOR DONE
2005  005172  100375                          BPL     1$
2006  005174  005077  175576                  CLR     @TBUF           ;LOAD SECOND BUFFER
2007  005200  000240                          NOP
2008  005202  000005                          RESET                   ;CLEAR DONE WITH RESET
2009  005204  105777  175564                  TSTB    @TCSR           ;CHECK FOR DONE SET
2010  005210  100401                          BMI     TST20           ;BR TO NEXT TEST IF DONE SET
2011
2012  005212  104035                          ERROR   35              ;TCSR 'DONE'' DOES NOT SET WITH RESET
2013
2014
2015
2016                                  ;;*******************************************************************
2017                                  ;*TEST 20        TEST ABILITY TO ACCESS SLU2 RCSR
2018                                  ;;*******************************************************************
2019  005214  000004                  TST20:  SCOPE
2020  005216  012737  000020  001104          MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2021  005224  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
2022  005230  012737  005244  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
2023  005236  005777  175526                  TST     @RCSR           ;ACCESS RCSR
2024  005242  000402                          BR      2$              ;BR TO END OF TEST
2025
2026  005244  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
2027  005246  104036                          ERROR   36              ;CAN NOT ACCESS RCSR
2028  005250  010337  000004          2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
2029
2030
2031
2032                                  ;;*******************************************************************
2033                                  ;*TEST 21        TEST ABILITY TO ACCESS SLU2 RBUF
2034                                  ;;*******************************************************************
2035  005254  000004                  TST21:  SCOPE
2036  005256  012737  000021  001104          MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2037  005264  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
2038  005270  012737  005304  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
2039  005276  005777  175470                  TST     @RBUF           ;ACCESS RBUF
2040  005302  000402                          BR      2$              ;BR TO END OF TEST
2041
2042  005304  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
```

```
                                        ERROR   37              ;CAN NOT ACCESS RBUF
005306  104037
005310  010337  C00004        2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
                                ;;**********************************************************************
                                ;*TEST 22       TEST SLU2 BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET
                                ;;**********************************************************************
005314  000004                TST22:  SCOPE
005316  012737  000022  001104        MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
005324  042777  000001  175442        BIC     #BIT0,@TCSR     ;MAKE SURE BIT UNDER TEST IS INITIALIZED
005332  032777  000001  175434        BIT     #BIT0,@TCSR     ;CHECK BIT0 OF TCSR CLEAR
005340  001401                        BEQ     1$              ;BR IF CLEAR
005342  104040                        ERROR   40              ;BIT0 WAS NOT CLEAR AFTER RESET
005344  052777  000001  175422  1$:   BIS     #BIT0,@TCSR     ;SET BIT0 IN TCSR
005352  032777  000001  175414        BIT     #BIT0,@TCSR     ;TEST BIT0 OF TCSR
005360  001001                        BNE     2$              ;BR IF SET
005362  104041                        ERROR   41              ;BIT0 OF TCSR WILL NOT SET
005364  042777  000001  175402  2$:   BIC     #BIT0,@TCSR     ;CLEAR BIT0 OF TCSR
005372  032777  000001  175374        BIT     #BIT0,@TCSR     ;TEST BIT0 OF TCSR
005400  001401                        BEQ     3$
005402  104042                        ERROR   42              ;BIT0 OF TCSR WILL NOT CLEAR
005404                          3$:
005404  032737  000001  001120        BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
005412  001403                        BEQ     70$             ;IF NO THEN DO TEST
005414  005737  001106                TST     @#$PASS         ;IS THIS FIRST PASS
005420  001014                        BNE     TST23           ;IF NO THEN SKIP TO NEXT TEST
005422                          70$:
005422  052777  000001  175344        BIS     #BIT0,@TCSR     ;SET BIT0 IN TCSR
005430  000005                        RESET                   ;CLEAR BIT0 WITH RESET
005432  032777  000001  175334        BIT     #BIT0,@TCSR     ;TEST BIT0 CLEAR
005440  001404                        BEQ     TST23           ;BR IF CLEAR
005442  042777  000001  175324        BIC     #BIT0,@TCSR     ;CLEAR BIT0, TO PRINT ERROR
005450  104043                        ERROR   43              ;RESET DID NOT CLEAR BIT0 OF TCSR
                                ;;**********************************************************************
                                ;*TEST 23       TEST THAT SLU2 BIT6(XMIT INT EN) CAN BE SET & RESET
                                ;;**********************************************************************
005452  000004                TST23:  SCOPE
005454  012737  000023  001104        MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
005462  042777  000100  175304        BIC     #BIT6,@TCSR     ;MAKE SURE BIT UNDER TEST IS INITIALIZED
005470  017703  175322                MOV     @TVECT,R3       ;SAVE XMIT VECTOR
005474  012777  005524  175314        MOV     #1$,@TVECT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
005502  004737  015056                JSR     PC,WRPSW                ;SET PSW TO PRIORITY-7
005506  000340                        .WORD   340
005510  032777  000100  175256        BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
005516  001404                        BEQ     2$              ;BR IF ZERO
005520  104044                        ERROR   44
                                                                ;BIT6 IN TCSR NOT CLEAR AFTER RESET
005522  000402                        BR      2$

005524  022626                1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
005526  104045                        ERROR   45
                                                                ;XMIT INTERRUPT OCCURRED  PRIO 7
```

```
2099  005530  052777  000100  175236  2$:     BIS     #BIT6,@TCSR     ;SET BIT6 OF TCSR
2100  005536  032777  000100  175230          BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
2101  005544  001001                          BNE     3$              ;BR, IF SET
2102
2103  005546  104046                          ERROR   46
2104                                                                  ;CANNOT SET BIT6 OF TCSR
2105
2106  005550  042777  000100  175216  3$:     BIC     #BIT6,@TCSR     ;CLEAR BIT6 OF TCSR
2107  005556  032777  000100  175210          BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
2108  005564  001401                          BEQ     4$              ;BR IF CLEAR
2109  005566  104047                          ERROR   47
2110                                                                  ;CANNOT CLEAR BIT6 OF TCSR
2111
2112  005570  032737  000001  001120  4$:     BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
2113  005576  001403                          BEQ     70$             ;IF NO THEN DO TEST
2114  005600  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
2115  005604  001011                          BNE     5$              ;IF NO THEN SKIP TO END OF TEST
2116  005606                          70$:
2117  005606  052777  000100  175160          BIS     #BIT6,@TCSR     ;SET BIT6 OF TCSR
2118  005614  000005                          RESET                   ;CLEAR BIT6 WITH RESET
2119  005616  032777  000100  175150          BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
2120  005624  001401                          BEQ     5$              ;BR IF CLEAR
2121
2122  005626  104050                          ERROR   50
2123                                                                  ;CANNOT CLEAR BIT6 OF TCSR WITH RESET
2124  005630  010377  175162          5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
2125
2126
2127                                  ;;*******************************************************************
2128                                  ;*TEST 24        TEST THAT SLU2 BIT6 OF RCSR CAN BE SET & RESET
2129                                  ;;*******************************************************************
2130  005634  000004                  TST24:  SCOPE
2131  005636  012737  000024  001104          MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2132  005644  042777  000100  175116          BIC     #BIT6,@RCSR     ;MAKE SURE BIT UNDER TEST IS INITIALIZED
2133  005652  017703  175134                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
2134  005656  012777  005706  175126          MOV     #1$,@RVECT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
2135  005664  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY=7
2136  005670  000340                          .WORD   340
2137  005672  032777  000100  175070          BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
2138  005700  001404                          BEQ     2$
2139  005702  104051                          ERROR   51
2140                                                                  ;BIT6 OF RCSR NOT CLEAR AFTER RESET
2141  005704  000402                          BR      2$
2142
2143  005706  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2144  005710  104052                          ERROR   52
2145                                                                  ;RCVR INTERRUPT WITH PRIORITY=7
2146
2147  005712  052777  000100  175050  2$:     BIS     #BIT6,@RCSR     ;SET BIT6 OF RCSR
2148  005720  032777  000100  175042          BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
2149  005726  001001                          BNE     3$              ;BR, IF SET
2150
2151  005730  104053                          ERROR   53
2152                                                                  ;CANNOT SET BIT6 OF RCSR
2153
2154  005732  042777  000100  175030  3$:     BIC     #BIT6,@RCSR     ;CLEAR BIT6 OF RCSR
```

```
2155  005740  032777  000100  175022          BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
2156  005746  001401                          BEQ     4$              ;BR, IF CLEAR
2157
2158  005750  104054                          ERROR   54
2159                                                                  ;CANNOT CLEAR BIT6 OF RCSR
2160
2161  005752  032737  000001  001120  4$:     BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
2162  005760  001403                          BEQ     70$             ;IF NO THEN DO TEST
2163  005762  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
2164  005766  001011                          BNE     5$              ;IF NO THEN SKIP TO END OF TEST
2165  005770                          70$:
2166  005770  052777  000100  174772          BIS     #BIT6,@RCSR     ;SET BIT6 OF RCSR
2167  005776  000005                          RESET                   ;CLEAR BIT6OF RCSR WITH RESET
2168  006000  032777  000100  174762          BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
2169  006006  001401                          BEQ     5$              ;BR, IF CLEAR
2170
2171  006010  104055                          ERROR   55
2172                                                                  ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
2173  006012  010377  174774          5$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
2174
2175
2176
2177                                  ;;*******************************************************************
2178                                  ;*TEST 25       TEST THAT SLU2 RCVR DONE (7) SET & CLEAR PROPERLY
2179                                  ;;*******************************************************************
2180  006016  000004                  TST25:  SCOPE
2181  006020  012737  000025  001104          MOV     #25,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2182  006026  005077  174740                  CLR     @RBUF           ;INITIALIZE REGISTER BEFORE TESTING
2183  006032  005000                          CLR     R0              ;CLEAR TIMER
2184  006034  005077  174736                  CLR     @TBUF           ;LOAD TRANSMIT BUFFER
2185  006040  105777  174724          WDONE:  TSTB    @RCSR           ;CHECK FOR RECEIVER DONE
2186  006044  100403                          BMI     1$              ;BR, IF DONE
2187  006046  005200                          INC     R0              ;INCREMENT TIMER, IF NOT DONE
2188  006050  001373                          BNE     WDONE           ;CONTINUE WAIT IF TIME REMAINS
2189  006052  104056                          ERROR   56
2190                                                                  ;RECEIVER DONE NEVER SET
2191
2192  006054  032737  000001  001120  1$:     BIT #1,@#$ENV   ;ARE WE RUNNING UNDER APT
2193  006062  001403                          BEQ     70$             ;IF NO THEN DO TEST
2194  006064  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
2195  006070  001005                          BNE     2$              ;IF NO THEN SKIP TO END OF TEST
2196  006072                          70$:
2197  006072  000005                          RESET                   ;CLEAR DONE WITH RESET
2198  006074  105777  174670                  TSTB    @RCSR           ;CHECK FOR DONE CLEAR
2199  006100  001401                          BEQ     2$
2200
2201  006102  104057                          ERROR   57
2202                                                                  ;RESET DID NOT CLEAR RCVR DONE
2203  006104  005777  174662          2$:     TST     @RBUF           ;CLEAR RECEIVER BUFFER
2204
2205
2206
2207                                  ;;*******************************************************************
2208                                  ;*TEST 26       TEST SLU2 THAT READING RBUF CLEARS RECEIVER DONE
2209                                  ;;*******************************************************************
2210  006110  000004                  TST26:  SCOPE
```

```
2211  006112  012737  000026  001104          MOV     #26,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2212  006120  005077  174646                  CLR     @RBUF           ;INITIALIZE REGISTER BEFORE TESTING
2213  006124  005077  174646                  CLR     @TBUF           ;LOAD TRANSMITTER
2214  006130  105777  174634          1$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
2215  006134  100375                          BPL     1$
2216  006136  017700  174630                  MOV     @RBUF,R0        ;READ RECEIVE BUFFER
2217  006142  105777  174622                  TSTB    @RCSR           ;CHECK FOR RECEIVE DONE CLEAR
2218  006146  001401                          BEQ     TST27           ;BR, IF CLEAR TO NEXT TEST
2219  006150  104060                          ERROR   60
2220                                                                  ;READING RBUF DID NOT CLEAR RCVR DONE
2221  006152                          LTCRT:
2222
2223                                  ;;*******************************************************************
2224                                  ;*TEST 27        TEST ABILITY TO ACCESS LKS
2225                                  ;;*******************************************************************
2226  006152  000004                  TST27:  SCOPE
2227  006154  012737  000027  001104          MOV     #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2228  006162  032737  000004  001156          BIT     #BIT2,@#$DEVM   ;DO THESE TESTS FOR THIS DEVICE?
2229  006170  001002                          BNE     99$             ;F YES CONTINUE WITH TESTS
2230  006172  000137  006524                  JMP     UNIQUE          ;IF NO GO TO START OF NEXT SET OF TESTS.
2231  006176                          99$:
2232  006176  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
2233  006202  012737  006216  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
2234  006210  005777  174574                  TST     @LKS            ;ACCESS LKS
2235  006214  000402                          BR      2$              ;NO TIMEOUT -  BR TO END OF TEST
2236
2237  006216  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
2238  006220  104061                          ERROR   61              ;CAN NOT ACCESS LKS
2239
2240  006222  010337  000004          2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
2241
2242                                  ;;*******************************************************************
2243                                  ;*TEST 30        TEST THAT BIT6 OF LKS CAN BE SET & RESET
2244                                  ;;*******************************************************************
2245  006226  000004                  TST30:  SCOPE
2246  006230  012737  000030  001104          MOV     #30,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2247  006236  042777  000100  174544          BIC     #BIT6,@LKS      ;MAKE SURE BIT UNDER TEST IS INITIALIZED BEFORE TESTING
2248  006244  017703  174562                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
2249  006250  012777  006300  174554          MOV     #1$,@RTCVT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
2250  006256  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2251  006262  000340                                  .WORD   340
2252  006264  032777  000100  174516          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
2253  006272  001404                          BEQ     2$
2254  006274  104062                          ERROR   62
2255                                                                  ;BIT6 OF LKS NOT CLEAR AFTER RESET
2256  006276  000402                          BR      2$
2257
2258  006300  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2259  006302  104063                          ERROR   63
2260                                                                  ;LKS INTERRUPT WITH PRIORITY 7
2261
2262  006304  052777  000100  174476  2$:     BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
2263  006312  032777  000100  174470          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
2264  006320  001001                          BNE     3$              ;BR IF SET
2265
2266  006322  104064                          ERROR   64
```

```
                                                        ;CANNOT SET BIT6 OF LKS

006324  042777  000100  174456  3$:     BIC     #BIT6,@LKS      ;CLEAR BIT6 OF LKS
006332  032777  000100  174450          BIT     #BIT6,@LKS      ;TEST BIT6 OF LK
006340  001401                          BEQ     4$
006342  104065                          ERROR   65
                                                        ;CANNOT CLEAR BIT6 OF LKS
006344  032737  000001  001120  4$:     BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
006352  001403                          BEQ     70$             ;IF NO THEN DO TEST
006354  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
006360  001011                          BNE     5$              ;IF NO THEN SKIP TO END OF TEST
006362                          70$:
006362  052777  000100  174420          BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
006370  000005                          RESET                   ;CLEAR BIT6 OF LKS WITH RESET
006372  032777  000100  174410          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
006400  001401                          BEQ     5$              ;BR IF CLEAR

006402  104066                          ERROR   66
                                                        ;CANNOT CLEAR BIT6 OF LKS WITH RESET
006404  010377  174422          5$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR



                        ;;*******************************************************************
                        ;*TEST 31        TEST THAT BIT7 OF LKS SETS & CAN BE CLEARED
                        ;;*******************************************************************
006410  000004          TST31:  SCOPE
006412  012737  000031  001104          MOV     #31,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
006420  032737  000001  001120          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
006426  001403                          BEQ     70$             ;IF NO THEN DO TEST
006430  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
006434  001033                          BNE     TST32           ;IF NO THEN SKIP TO NEXT TEST
006436                          70$:
006436  000005                          RESET                   ;CLEAR EVERYTHING & SET BIT7 OF LKS
006440  105777  174344          1$·     TSTB    @LKS            ;TEST FOR BIT7 OF LKS
006444  100401                          BMI     2$              ;BR IF SET

006446  104067                          ERROR   67              ;BIT7 OF LKS DID NOT SET WITH RESET

006450  042777  000200  174332  2$:     BIC     #BIT7,@LKS      ;CLEAR BIT7 OF LKS
006456  032777  000200  174324          BIT     #BIT7,@LKS      ;TEST BIT7 OF LKS
006464  001410                          BEQ     3$
006466  042777  000200  174314          BIC     #BIT7,@LKS      ;TRY ONE MORE TIME BECAUSE THE CLOCK
006474  032777  000200  174306          BIT     #BIT7,@LKS      ;  MAY HAVE SET IMMEDIATELY AFTER THE FIRST CLEAR
006502  001401                          BEQ     3$

006504  104070                          ERROR   70              ;CAN NOT CLEAR BIT7 OF LKS

006506  005000          3$:     CLR     R0              ;CLEAR TIMER
006510  105777  174274  CONT:   TSTB    @LKS            ;TEST FOR BIT7 OF LKS
006514  100403                          BMI     TST32           ;BR, IF SET
006516  005200                          INC     R0              ;INCREMENT TIMER
006520  001373                          BNE     CONT            ;CONTINUE UNTIL TIME EXPIRES

006522  104071                          ERROR   71              ;BIT7 OF LKS DOES NOT SET
```

```
     006524                                  UNIQUE:

                                             ;;********************************************************************
                                             ;;*TEST 32        UNIQUE INTERNAL ADDRESS TEST
                                             ;;********************************************************************
     006524  000004                          TST32:  SCOPE
     006526  012737  000032  001104                  MOV     #32,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX
     006534  032737  000001  001120                  BIT     #1, @#$ENV          ;ARE WE RUNNING UNDER APT
     006542  001403                                  BEQ     70$                 ;IF NO THEN DO TEST
     006544  005737  001106                          TST     @#$PASS             ;IS THIS FIRST PASS
     006550  001053                                  BNE     TST33               ;IF NO THEN SKIP TO NEXT TEST
     006552                                  70$:
     006552  012737  000340  177776                  MOV     #340,PS             ;WE WILL BE PLAYING WITH BIT6
                                                                                 ;SO LOCK OUT EXTRAINEOUS INTERRUPTS
     006560  012700  002770                          MOV     #RCSR,R0            ;GET LOCATION OF FIRST REGISTER ADDRESS
     006564  012703  002770                  1$:     MOV     #RCSR,R3            ;MAKE R3 POINT TO LOCATION OF FIRST
                                                                                 ;REGISTER ADDRESS
     006570  012701  000011                          MOV     #11,R1              ;SET LOOP COUNTER TO CLEAR ALL REG.
     006574  005033                          2$:     CLR     @(R3)+              ;CLEAR A REGISTER
     006576  077102                                  SOB     R1,2$               ;LOOP UNTIL ALL REGISTERS CLEARED
     006600  012770  000100  000000                  MOV     #BIT6,@(R0)         ;SET TEST BIT IN DEVICE REGISTERS
     006606  012701  002770                          MOV     #RCSR,R1            ;GET LOCATION OF FIRST REGISTER ADDRESS
     006612  012702  000011                          MOV     #11,R2              ;SET UP TEST LOOP COUNTER
     006616  032731  000100                  3$:     BIT     #BIT6,@(R1)+        ;IS TEST BIT SET IN THIS REGISTER
     006622  001006                                  BNE     5$                  ;IF YES GO SEE IF THERE IS AN ERROR
     006624  077204                          4$:     SOB     R2,3$               ;LOOP UNTIL ALL REGISTER CHECKED
     006626  020027  003010                          CMP     R0,#LKS             ;ARE WE DONE TESTING
     006632  001422                                  BEQ     7$                  ;IF YES GO TO NEXT TEST
     006634  005030                                  CLR     @(R0)+              ;CLEAR REGISTER JUST TESTED AND POINT
                                                                                 ;TO NEXT ONE
     006636  000752                                  BR      1$                  ;CONTINUE TESTING
     006640  021041                          5$:     CMP     (R0),-(R1)          ;DID WE COMPARE THE REGISTER TO ITSELF?
     006642  001413                                  BEQ     6$                  ;IF YES GET OVER ERROR CALL
     006644  011037  001020                          MOV     (R0),$GDADR         ;IF NO SET UP ERROR INFORMATION
     006650  011137  001022                          MOV     (R1),$BDADR
     006654  017037  000000  001024                  MOV     @(R0),$GDDAT
     006662  017137  000000  001026                  MOV     @(R1),$BDDAT
     006670  104072                                  ERROR   72                  ;WRITE TO 1 INTERNAL ADDRESS MODIFIED
                                                                                 ;ANOTHER SO ADDRESS NOT UNIQUE
     006672  062701  000002                  6$:     ADD     #2,R1               ;RESTORE POINTER
     006676  000752                                  BR      4$                  ;GET BACK IN TEST LOOP
     006700                                  7$:
```

```
2367  006700                                SLU1IT:
2368
2369                                        ;;*****************************************************************
2370                                        ;*TEST 33       TEST THAT SLU1 XMIT INTERRUPTS ONLY WHEN ENABLED
2371                                        ;;*****************************************************************
2372  006700  000004                       TST33:  SCOPE
2373  006702  012737  000033  001104               MOV     #33,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2374  006710  032737  000001  001120               BIT     #1,@#$ENV       ;IF NOT UNDER APT
2375  006716  001405                              BEQ     70$             ;  THEN RUN THIS SERIES OF TESTS
2376  006720  005737  001106                      TST     @#$PASS         ;  ELSE IF FIRST PASS
2377  006724  001402                              BEQ     70$             ;         THEN RUN THESE TESTS
2378  006726  000137  010600                      JMP     SLU2IT          ;         ELSE DO NOT RUN THESE TESTS
2379  006732                                70$:
2380  006732  032737  000001  001156               BIT     #BIT0,@#$DEVM   ;DO THESE TESTS FOR THIS DEVICE?
2381  006740  001002                              BNE     99$             ;F YES CONTINUE WITH TESTS
2382  006742  000137  010600                      JMP     SLU2IT          ;IF NO GO TO START OF NEXT SET OF TESTS.
2383  006746                                99$:
2384  006746  000005                              RESET                   ;CLEAR THE WORLD
2385  006750  042777  000100  174026               BIC     #BIT6,@CTCSR    ;CLEAR TRANSMIT INTERRUPT ENABLE
2386  006756  017703  174044                      MOV     @CTVECT,R3      ;SAVE XMIT VECTOR
2387  006762  012777  007006  174036               MOV     #2$,@CTVECT     ;POINT XMIT VECTOR TO ERROR REPORT
2388  006770  105777  174010                1$:   TSTB    @CTCSR          ;WAIT FOR DONE
2389  006774  100375                              BPL     1$
2390  006776  004737  015056                      JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2391  007002  000140                                      .WORD   140
2392  007004  000402                              BR      3$
2393
2394  007006  022626                        2$:   CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2395  007010  104074                              ERROR   74
2396                                                                      ;XMIT INTERRUPTS WITH INTERRUPT ENABLE CLEAR
2397  007012  012777  007032  174006        3$:   MOV     #4$,@CTVECT     ;SET XMIT VECTOR TO END OF TEST
2398  007020  052777  000100  173756               BIS     #BIT6,@CTCSR    ;ENABLE INTERRUPTS
2399  007026  000240                              NOP
2400
2401  007030  104075                              ERROR   75              ;XMIT DID NOT INTERRUPT
2402
2403  007032  042777  000100  173744        4$:   BIC     #BIT6,@CTCSR    ;DISABLE INTERRUPTS
2404  007040  022626                              CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2405  007042  010377  173760                      MOV     R3,@CTVECT      ;RESTORE XMIT VECTOR
2406
2407
2408                                        ;;*****************************************************************
2409                                        ;*TEST 34       TEST SLU1 XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
2410                                        ;;*****************************************************************
2411  007046  000004                       TST34:  SCOPE
2412  007050  012737  000034  001104               MOV     #34,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2413  007056  042777  000100  173720               BIC     #BIT6,@CTCSR    ;DISABLE INTERRUPTS
2414  007064  004737  015056                      JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2415  007070  000340                                      .WORD   340
2416  007072  017703  173730                      MOV     @CTVECT,R3      ;SAVE XMIT VECTOR
2417  007076  012777  007124  173722               MOV     #2$,@CTVECT     ;POINT XMIT VECTOR TO ERROR REPORT
2418  007104  105777  173674                1$:   TSTB    @CTCSR          ;WAIT FOR DONE
2419  007110  100375                              BPL     1$
2420  007112  052777  000100  173664               BIS     #BIT6,@CTCSR    ;ENABLE INTERRUPT
2421  007120  000240                              NOP
2422  007122  000402                              BR      3$              ;CONTINUE TEST
```

```
2423
2424  007124  022626                    2$:   CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2425  007126  104076                          ERROR   76
2426                                                                  ;XMIT INTERRUPTS AT PRIORITY=7
2427  007130  042777  000100  173646    3$:   BIC     #BIT6,@CTCSR    ;CLEAR INTERRUPT ENABLE
2428  007136  012777  007156  173662          MOV     #4$,@CTVECT     ;POINT XMIT VECTOR TO ERROR REPORT
2429  007144  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2430  007150  000140                          .WORD   140
2431  007152  000240                          NOP
2432  007154  000402                          BR      5$              ;BR TO END OF TEST-NO INTERRUPT
2433
2434  007156  022626                    4$:   CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2435  007160  104077                          ERROR   77
2436                                                                  ;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
2437  007162  010377  173640             5$:   MOV     R3,@CTVECT      ;RESTORE XMIT VECTOR
2438
2439
2440                                    ;;*******************************************************************
2441                                    ;*TEST 35        TEST SLU1 TRANSMITTER FOR DOUBLE INTERRUPTS
2442                                    ;;*******************************************************************
2443  007166  000004                    TST35:  SCOPE
2444  007170  012737  000035  001104            MOV     #35,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2445  007176  042777  000100  173600            BIC     #BIT6,@CTCSR    ;CLEAR INTERRUPT ENABLE
2446  007204  017703  173616                  MOV     @CTVECT,R3      ;SAVE XMIT VECTOR
2447  007210  017704  173614                  MOV     @CTPSW,R4       ;SAVE XMIT PSW VECTOR
2448  007214  012777  007256  173604          MOV     #2$,@CTVECT     ;SET UP XMIT VECTOR
2449  007222  012777  000340  173600          MOV     #340,@CTPSW     ;SET PIO 7 AFTER INTERRUPT
2450  007230  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2451  007234  000140                          .WORD   140
2452  007236  105777  173542             1$:   TSTB    @CTCSR          ;WAIT FOR DONE
2453  007242  100375                          BPL     1$
2454  007244  052777  000100  173532          BIS     #BIT6,@CTCSR    ;ENABLE INTERRUPTS
2455  007252  000240                          NOP
2456
2457  007254  104100                          ERROR   100
2458                                                                  ;XMIT INTERRUPT DID NOT OCCUR
2459  007256  022626                    2$:   CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2460  007260  012777  007306  173540          MOV     #4$,@CTVECT     ;POINT XMIT VECTOR TO ERROR
2461  007266  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2462  007272  000140                          .WORD   140
2463  007274  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPTS
2464  007276  042777  000100  173500          BIC     #BIT6,@CTCSR    ;DISABLE INTERRUPTS
2465  007304  000402                          BR      5$              ;BR TO END OF TEST
2466
2467  007306  022626                    4$:   CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2468  007310  104101                          ERROR   101
2469                                                                  ;XMIT RE-INTERRUPTED
2470  007312  010377  173510             5$:   MOV     R3,@CTVECT      ;RESTORE XMIT VECTOR
2471  007316  010477  173506                  MOV     R4,@CTPSW       ;RESTORE XMIT PSW VECTOR
2472
2473                                    ;;*******************************************************************
2474                                    ;*TEST 36        TEST THAT SLU1 XMIT INTERRUPT CLEARS WITH LOADING TBUF
2475                                    ;;*******************************************************************
2476  007322  000004                    TST36:  SCOPE
2477  007324  012737  000036  001104            MOV     #36,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2478  007332  042777  000100  173444            BIC     #BIT6,@CTCSR    ;DISABLE INTERRUPTS
```

```
2479  007340  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2480  007344  000340                          .WORD   340
2481  007346  017703  173454                  MOV     @CTVECT,R3      ;SAVE XMIT VECTOR
2482  007352  012777  007424  173446          MOV     #2$,@CTVECT     ;POINT XMIT VECTOR TO ERROR
2483  007360  052777  000100  173416          BIS     #BIT6,@CTCSR    ;ENABLE INTERRUPTS
2484  007366  005077  173414                  CLR     @CTBUF          ;LOAD TBUF
2485  007372  105777  173406          1$:     TSTB    @CTCSR          ;WAIT FOR DONE (INTERRUPT)
2486  007376  100375                          BPL     1$
2487  007400  005077  173402                  CLR     @CTBUF          ;FILL SECOND BUFFER TO RESET INT.
2488  007404  004737  015056                  JSR     PC,WRPSW        ;ALLOW INTERRUPTS
2489  007410  000140                          .WORD   140
2490  007412  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPTS
2491  007414  042777  000100  173362          BIC     #BIT6,@CTCSR    ;DISABLE INTERRUPTS
2492  007422  000402                          BR      3$              ;BR TO END OF TEST
2493
2494  007424  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2495  007426  104102                          ERROR   102
2496                                                                  ;LOADING TBUF DID NOT CLEAR INTERRUPT.
2497  007430  010377  173372          3$:     MOV     R3,@CTVECT      ;RESTORE XMIT VECTOR
2498
2499
2500
2501
2502
2503
2504
2505
2506                                  ;;*******************************************************************
2507                                  ;*TEST 37        TEST THAT SLU1 RCVR INTERRUPTS ONLY WHEN ENABLED
2508                                  ;;*******************************************************************
2509  007434  000004          TST37:  SCOPE
2510  007436  012737  000037  001104          MOV     #37,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2511  007444  000005                          RESET                   ;MAKE SURE NO INTERRUPTS ARE PENDING FROM
2512                                                                  ;ANOTHER TEST
2513  007446  042777  000100  173330          BIC     #BIT6,@CTCSR    ;DISABLE TRANSMIT INTERRUPTS
2514  007454  042777  000100  173316          BIC     #BIT6,@CRCSR    ;DISABLE RECEIVER INTERRUPTS
2515  007462  052777  000004  173314          BIS     #BIT2,@CTCSR    ;SET MAINTENANCE WRAP
2516  007470  017703  173326                  MOV     @CRVECT,R3      ;SAVE RECEIVE VECTOR
2517  007474  012777  007532  173320          MOV     #2$,@CRVECT     ;POINT RCV VECTOR TO ERROR REPORT
2518  007502  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2519  007506  000140                          .WORD   140
2520  007510  005077  173272                  CLR     @CTBUF          ;SEND A CHARACTER
2521  007514  105777  173260          1$:     TSTB    @CRCSR          ;WAIT FOR RECEIVER DONE
2522  007520  100375                          BPL     1$
2523  007522  042777  000004  173254          BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
2524  007530  000405                          BR      3$              ;CONTINUE TEST
2525
2526  007532  042777  000004  173244  2$:     BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
2527  007540  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2528  007542  104103                          ERROR   103
2529                                                                  ;RECEIVER INTERRPUTS WITH INT. ENABLE CLEAR
2530
2531  007544  012777  007572  173250  3$:     MOV     #4$,@CRVECT     ;POINT RCV VECTOR TO END OF TEST
2532  007552  052777  000100  173220          BIS     #BIT6,@CRCSR    ;ENABLE RCV INTERRUPTS
2533  007560  000240                          NOP                     ;GIVE ANY INTERRUPTS TIME
2534  007562  042777  000004  173214          BIC     #BIT2,@C1CSR    ;CLEAR MAINTENANCE BIT
```

```
2535  007570  104104                          ERROR   104
2536                                                    ;RCVR DID NOT INTERRUPT
2537
2538  007572  042777  000100  173200  4$:     BIC     #BIT6,@CRCSR    ;DISABLE INTERRUPTS
2539  007600  042777  000004  173176          BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
2540  007606  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2541  007610  010377  173206                  MOV     R3,@CRVECT      ;RESTORE RECEIVE VECTOR
2542
2543
2544                                  ;;*******************************************************************
2545                                  ;*TEST 40        TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
2546                                  ;;*******************************************************************
2547  007614  000004                  TST40:  SCOPE
2548  007616  012737  000040  001104          MOV     #40,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2549  007624  000005                          RESET                   ;CLEAR EVERYTHING
2550  007626  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2551  007632  000340                                  .WORD   340
2552  007634  017703  173162                  MOV     @CRVECT,R3      ;SAVE RECEIVE VECTOR
2553  007640  012777  007700  173154          MOV     #2$,@CRVECT     ;POINT RCVR VECTOR TO ERROR REPORT
2554  007646  052777  000004  173130          BIS     #BIT2,@CTCSR    ;SET MAINTENANCE WRAP
2555  007654  005077  173126                  CLR     @CTBUF          ;SEND A CHARACTER
2556  007660  105777  173114          1$:     TSTB    @CRCSR          ;WAIT FOR RECEIVER DONE
2557  007664  100375                          BPL     1$
2558  007666  052777  000100  173104          BIS     #BIT6,@CRCSR    ;ENABLE INTERRUPTS
2559  007674  000240                          NOP                     ;GIVE TIME FOR INTERRUPT
2560  007676  000405                          BR      3$              ;CONTINUE TEST
2561  007700  042777  000004  173076  2$:     BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
2562  007706  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2563  007710  104105                          ERROR   105
2564                                                    ;RCVR INTERRUPTS AT PRIORITY 7
2565
2566  007712  042777  000100  173060  3$:     BIC     #BIT6,@CRCSR    ;CLEAR INTERRUPT ENABLE
2567  007720  012777  007746  173074          MOV     #4$,@CRVECT     ;POINT RCVR VECTOR TO ERROR REPORT
2568  007726  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2569  007732  000140                                  .WORD   140
2570  007734  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPT
2571  007736  042777  000004  173040          BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
2572  007744  000405                          BR      5$              ;BR TO END OF TEST, IF NO INTERRUPT
2573
2574  007746  042777  000004  173030  4$:     BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
2575  007754  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2576  007756  104106                          ERROR   106
2577                                                    ;RCVR INTERRUPT REQUEST PASSED WITH BITE CLEAR
2578  007760  010377  173036          5$:     MOV     R3,@CRVECT      ;RESTORE RECEIVE VECTOR
2579
2580
2581                                  ;;*******************************************************************
2582                                  ;*TEST 41        TEST SLU1 RECEIVER FOR DOUBLE INTERRUPTS
2583                                  ;;*******************************************************************
2584  007764  000004                  TST41:  SCOPE
2585  007766  012737  000041  001104          MOV     #41,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2586  007774  000005                          RESET                   ;CLEAR EVERYTHING
2587  007776  017703  173020                  MOV     @CRVECT,R3      ;SAVE RECEIVE VECTOR
2588  010002  017704  173016                  MOV     @CRPSW,R4       ;SAVE RECEIVE PSW VECTOR
2589  010006  012777  010070  173006          MOV     #2$,@CRVECT     ;POINT RCV VECTOR TO CONTINUE  EST
2590  010014  012777  000340  173002          MOV     #340,@CRPSW     ;SET PRIORITY TO 7 AFTER INTERRUPT
```

```
2591  010022  004737  015056          JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2592  010026  000140                  .WORD   140
2593  010030  052777  000004  172746  BIS     #BIT2,@CTCSR    ;SET MAINTENANCE WRAP
2594  010036  005077  172744          CLR     @CTBUF          ;SEND A CHARACTER
2595  010042  105777  172732  1$:     TSTB    @CRCSR          ;WAIT FOR RCVR DONE
2596  010046  100375                  BPL     1$
2597  010050  042777  000004  172726  BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
2598  010056  052777  000100  172714  BIS     #BIT6,@CRCSR    ;ENABLE RCV INTERRUPTS
2599  010064  000240                  NOP                     ;GIVE SOME TIME
2600
2601  010066  104107                  ERROR   107
2602                                                          ;RCVR INTERRUPT DID NOT OCCUR
2603
2604  010070  022626          2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2605  010072  012777  010124  172722  MOV     #3$,@CRVECT     ;POINT RCV VECTOR TO ERROR REPORT
2606  010100  004737  015056          JSR     PC,WRPSW        ;RESET PSW TO PRIORITY 3
2607  010104  000140                  .WORD   140
2608  010106  000240                  NOP                     ;GIVE SOME TIME
2609  010110  042777  000100  172662  BIC     #BIT6,@CRCSR    ;CLEAR INTERRUPT ENABLE
2610  010116  010477  172702          MOV     R4,@CRPSW       ;RESTORE RECEIVE PSW VECTOR
2611  010122  000402                  BR      4$              ;BR TO END OF TEST
2612
2613  010124  022626          3$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2614  010126  104110                  ERROR   110
2615                                                          ;RECEIVER RE-INTERRUPTED
2616  010130  010377  172666  4$:     MOV     R3,@CRVECT      ;RESTORE RECEIVE VECTOR
2617
2618
2619                          ;;*******************************************************************
2620                          ;*TEST 42        TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
2621                          ;;*******************************************************************
2622  010134  000004  TST42:  SCOPE
2623  010136  012737  000042  001104          MOV     #42,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2624  010144  000005                          RESET                   ;CLEAR EVERYTHING
2625  010146  004737  015056                  JSR     PC,WRPSW        ;SET PSW PRIORITY TO 7
2626  010152  000340                          .WORD   340
2627  010154  017703  172642                  MOV     @CRVECT,R3      ;SAVE RECEIVE VECTOR
2628  010160  012777  010246  172634          MOV     #2$,@CRVECT     ;POINT RCV VECTOR TO ERROR REPORT
2629  010166  052777  000100  172604          BIS     #BIT6,@CRCSR    ;SET RCVR INTERRUPT ENABLE
2630  010174  052777  000004  172602          BIS     #BIT2,@CTCSR    ;SET MAINTENANCE WRAP
2631  010202  005077  172600                  CLR     @CTBUF          ;SEND A CHARACTER
2632  010206  105777  172566          1$:     TSTB    @CRCSR          ;WAIT FOR DONE (INTERRUPT)
2633  010212  100375                          BPL     1$
2634  010214  042777  000004  172562          BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
2635  010222  005777  172554                  TST     @CRBUF          ;READ RBUF TO CLEAR PENDING INTERRPUT
2636  010226  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2637  010232  000140                          .WORD   140
2638  010234  000240                          NOP                     ;ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
2639  010236  042777  000100  172534          BIC     #BIT6,@CRCSR    ;NO INTERRUPT-CLEAR INT. ENABLE
2640  010244  000402                          BR      3$
2641
2642  010246  022626          2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2643  010250  104111                  ERROR   111             ;READING RBUF DID NOT CLEAR INTERRUPT
2644
2645  010252  010377  172544          3$:     MOV     R3,@CRVECT      ;RESTORE RECEIVE VECTOR
2646
```

```
2647
2648
2649                            ;;*******************************************************************
2650                            ;*TEST 43       TEST SLU1 THAT RESET CLEARS RECEIVE INTERRUPT
2651                            ;;*******************************************************************
2652  010256  000004    TST43:  SCOPE
2653  010260  012737  000043  001104        MOV     #43,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2654  010266  000005                RESET                           ;CLEAR EVERYTHING
2655  010270  004737  015056        JSR     PC,WRPSW                ;SET PSW TO PRIORITY 7
2656  010274  000340                .WORD   340
2657  010276  017703  172520        MOV     @CRVECT,R3              ;SAVE RECEIVE VECTOR
2658  010302  012777  010362  172512        MOV     #2$,@CRVECT     ;POINT RCV VECTOR TO ERROR REPORT
2659  010310  052777  000100  172462        BIS     #BIT6,@CRCSR    ;SET RCV INTERRUPT ENABLE
2660  010316  052777  000004  172460        BIS     #BIT2,@CTCSR    ;SET MAINTENANCE WRAP
2661  010324  012777  000377  172454        MOV     #377,@CTBUF     ;SEND AN ALL 1'S CHARACTER
2662  010332  105777  172442        1$:     TSTB    @CRCSR          ;WAIT FOR RCV DONE
2663  010336  100375                BPL     1$
2664  010340  000005                RESET                           ;CLEAR RCV INTERRUPT & RBUF
2665  010342  004737  015056        JSR     PC,WRPSW                ;SET PSW TO PRIORITY 3
2666  010346  000140                .WORD   140
2667  010350  000240                NOP                             ;ALLOW TIME FOR AN ERRONEOUS INTERRUPT
2668  010352  042777  000100  172420        BIC     #BIT6,@CRCSR    ;NO INTERRUPT-CLEAR INT. ENABLE
2669  010360  000402                BR      3$                      ;CONTINUE TEST
2670
2671
2672  010362  022626        2$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
2673  010364  104112                ERROR   112                     ;RESET DID NOT CLEAR RCVR INTERRUPT
2674
2675  010366  010377  172430        3$:     MOV     R3,@CRVECT      ;RESTORE RECEIVE VECTOR
2676
2677
2678                            ;;*******************************************************************
2679                            ;*TEST 44       TEST SLU1 THAT ''OVERRUN & ERROR'' BITS CAN BE SET
2680                            ;;*******************************************************************
2681  010372  000004    TST44:  SCOPE
2682  010374  012737  000044  001104        MOV     #44,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2683  010402  032777  002000  170430        BIT     #BIT10,@SWR     ;IS THIS TEST ENABLED
2684  010410  001032                BNE     TST45                   ;IF DISABLED, BR TO NEXT TEST
2685  010412  000005                RESET                           ;CLEAR EVERYTHING
2686  010414  052777  000004  172362        BIS     #BIT2,@CTCSR    ;SET MAINTENANCE WRAP
2687  010422  012700  000003        MOV     #3,R0                   ;SET CHARACTER COUNT TO SEND 3 CHAR.
2688  010426  005077  172354        1$:     CLR     @CTBUF          ;LOAD TRANSMIT BUFFER
2689  010432  105777  172346        2$:     TSTB    @CTCSR          ;WAIT FOR TRANSMIT DONE
2690  010436  100375                BPL     2$
2691  010440  005300                DEC     R0                      ;DECREMENT CHARACTER COUNT
2692  010442  001371                BNE     1$                      ;BR IF ALL CHARACTERS NOT TRANSMITTED
2693  010444  042777  000004  172332        BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
2694  010452  032777  040000  172322        BIT     #BIT14,@CRBUF   ;TEST FOR ''OR'' ERROR FLAG
2695  010460  001001                BNE     3$                      ;BR, IF SET
2696  010462  104113                ERROR   113
2697                                                                ;''OR'' ERROR FLAG DID NOT SET
2698
2699  010464  032777  100000  172310  3$:   BIT     #BIT15,@CRBUF   ;TEST ''ERROR'' FLAG
2700  010472  001001                BNE     4$                      ;BR, IF SET
2701  010474  104114                ERROR   114
2702                                                                ;''ERROR'' FLAG DID NOT SET WITH ''OR'' FLAG
```

```
2703  010476                                  4$:
2704
2705
2706                                          ;;*****************************************************************
2707                                          ;*TEST 45       TEST SLU1 DATA PATH USING MAINTENANCE WRAP
2708                                          ;;*****************************************************************
2709  010476  000004                         TST45:  SCOPE
2710  010500  012737  000045  001104                 MOV     #45,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2711  010506  000005                                 RESET                   ;CLEAR EVERYTHING
2712  010510  005001                                 CLR     R1              ;CLEAR REGISTER FOR TEST DATA
2713  010512  052777  000004  172264                 BIS     #BIT2,@CTCSR    ;SET MAINTENANCE WRAP
2714  010520  105201                         1$:     INCB    R1              ;INCREMENT THE TEST DATA
2715  010522  010177  172260                         MOV     R1,@CTBUF       ;XMIT A CHARACTER
2716  010526  105777  172246                 2$:     TSTB    @CRCSR          ;WAIT FOR RECEIVER DONE
2717  010532  100375                                 BPL     2$
2718  010534  017702  172242                         MOV     @CRBUF,R2       ;GET RECEIVED CHARACTER
2719  010540  020102                                 CMP     R1,R2           ;COMPARE DATA
2720  010542  001003                                 BNE     3$              ;BR, IF NON-COMPARE
2721  010544  105701                                 TSTB    R1              ;TEST XMIT DATA FOR ZERO
2722  010546  001411                                 BEQ     4$              ;BR, IF FINISHED
2723  010550  000763                                 BR      1$              ;CONTINUE IF NOT
2724  010552  010137  001024                 3$:     MOV     R1,$GDDAT       ;STORE THE EXPECTED DATA
2725  010556  010237  001026                         MOV     R2,$BDDAT       ;STORE RECEIVED DATA
2726  010562  042777  000004  172214                 BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
2727  010570  104115                                 ERROR   115             ;DATA COMPARE DATA
2728
2729  010572  042777  000004  172204  4$:     BIC     #BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
2730
2731
```

```
2732  010600                                SLU2IT:
2733
2734                                        ;;*******************************************************************
2735                                        ;*TEST 46       TEST THAT SLU2 XMIT INTERRUPTS ONLY WHEN ENABLED
2736                                        ;;*******************************************************************
2737  010600  000004                TST46:  SCOPE
2738  010602  012737  000046  001104        MOV     #46,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2739  010610  032737  000002  001156        BIT     #BIT1,@#$DEVM   ;DO THESE TESTS FOR THIS DEVICE?
2740  010616  001002                        BNE     99$             ;F YES CONTINUE WITH TESTS
2741  010620  000137  012544                JMP     LTCIT           ;IF NO GO TO START OF NEXT SET OF TESTS.
2742  010624                        99$:
2743  010624  042777  000100  172142        BIC     #BIT6,@TCSR     ;CLEAR TRANSMIT INTERRUPT ENABLE
2744  010632  017703  172160                MOV     @TVECT,R3       ;SAVE XMIT VECTOR
2745  010636  012777  010662  172152        MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
2746  010644  105777  172124        1$:     TSTB    @TCSR           ;WAIT FOR DONE
2747  010650  100375                        BPL     1$
2748  010652  004737  015056                JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2749  010656  000140                                .WORD   140
2750  010660  000402                        BR      3$
2751
2752  010662  022626                2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2753  010664  104116                        ERROR   116
2754                                                                ;XMIT INTERRUPTS WITH INTERRUPT ENABLE CLEAR
2755  010666  012777  010706  172122  3$:     MOV     #4$,@TVECT      ;SET XMIT VECTOR TO END OF TEST
2756  010674  052777  000100  172072        BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
2757  010702  000240                        NOP
2758
2759  010704  104117                        ERROR   117             ;XMIT DID NOT INTERRUPT
2760
2761  010706  042777  000100  172060  4$:     BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
2762  010714  022626                        CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2763  010716  010377  172074                MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
2764
2765
2766                                        ;;*******************************************************************
2767                                        ;*TEST 47       TEST SLU2 XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
2768                                        ;;*******************************************************************
2769  010722  000004                TST47:  SCOPE
2770  010724  012737  000047  001104        MOV     #47,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2771  010732  042777  000100  172034        BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
2772  010740  004737  015056                JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2773  010744  000340                                .WORD   340
2774  010746  017703  172044                MOV     @TVECT,R3       ;SAVE XMIT VECTOR
2775  010752  012777  011000  172036        MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
2776  010760  105777  172010        1$:     TSTB    @TCSR           ;WAIT FOR DONE
2777  010764  100375                        BPL     1$
2778  010766  052777  000100  172000        BIS     #BIT6,@TCSR     ;ENABLE INTERRUPT
2779  010774  000240                        NOP
2780  010776  000402                        BR      3$              ;CONTINUE TEST
2781
2782  011000  022626                2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2783  011002  104120                        ERROR   120
2784                                                                ;XMIT INTERRUPTS AT PRIORITY-7
2785  011004  042777  000100  171762  3$:     BIC     #BIT6,@TCSR     ;CLEAR INTERRUPT ENABLE
2786  011012  012777  011032  171776        MOV     #4$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
2787  011020  004737  015056                JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
```

```
                                        .WORD   140
                                NOP
                                BR      5$              ;BR TO END OF TEST-NO INTERRUPT

                        4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
                                ERROR   121
                                                        ;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
                        5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR



                        ;;**********************************************************************
                        ;*TEST 50        TEST SLU2 TRANSMITTER FOR DOUBLE INTERRUPTS
                        ;;**********************************************************************
                        TST50:  SCOPE
                                MOV     #50,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                BIC     #BIT6,@TCSR     ;CLEAR INTERRUPT ENABLE
                                MOV     @TVECT,R3       ;SAVE XMIT VECTOR
                                MOV     @TPSW,R4        ;SAVE XMIT PSW VECTOR
                                MOV     #2$,@TVECT      ;SET UP XMIT VECTOR
                                MOV     #340,@TPSW      ;SET PIO 7 AFTER INTERRUPT
                                JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
                                .WORD   140
                        1$:     TSTB    @TCSR           ;WAIT FOR DONE
                                BPL     1$
                                BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
                                NOP

                                ERROR   122
                                                        ;XMIT INTERRUPT DID NOT OCCUR
                        2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
                                MOV     #4$,@TVECT      ;POINT XMIT VECTOR TO ERROR
                                JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
                                .WORD   140
                                NOP                     ;GIVE TIME FOR ANY INTERRUPTS
                                BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
                                BR      5$              ;BR TO END OF TEST

                        4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
                                ERROR   123
                                                        ;XMIT RE-INTERRUPTED
                        5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
                                MOV     R4,@TPSW        ;RESTORE XMIT PSW VECTOR



                        ;;**********************************************************************
                        ;*TEST 51        TEST THAT SLU2 XMIT INTERRUPT CLEARS WITH LOADING TBUF
                        ;;**********************************************************************
                        TST51:  SCOPE
                                MOV     #51,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
                                BEQ     70$             ;IF NO THEN DO TEST
                                TST     @#$PASS         ;IS THIS FIRST PASS
                                BNE     TST52           ;IF NO THEN SKIP TO NEXT TEST
                        70$:
                                BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
```

```
2844  011232  004737  015056              JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2845  011236  000340                      .WORD   340
2846  011240  017703  171552              MOV     @TVECT,R3       ;SAVE XMIT VECTOR
2847  011244  012777  011316  171544      MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR
2848  011252  052777  000100  171514      BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
2849  011260  005077  171512              CLR     @TBUF           ;LOAD TBUF
2850  011264  105777  171504      1$:     TSTB    @TCSR           ;WAIT FOR DONE (INTERRUPT)
2851  011270  100375                      BPL     1$
2852  011272  005077  171500              CLR     @TBUF           ;FILL SECOND BUFFER TO RESET INT.
2853  011276  004737  015056              JSR     PC,WRPSW        ;ALLOW INTERRUPTS
2854  011302  000140                      .WORD   140
2855  011304  000240                      NOP                     ;GIVE TIME FOR ANY INTERRUPTS
2856  011306  042777  000100  171460      BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
2857  011314  000402                      BR      3$              ;BR TO END OF TEST
2858
2859  011316  022626              2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2860  011320  104124                      ERROR   124
2861                                                              ;LOADING TBUF DID NOT CLEAR INTERRUPT.
2862  011322  005001              3$:     CLR     R1              ;INITIALIZE LOOP COUNTER
2863  011324  005201              4$:     INC     R1              ;INCREMENT LOOP COUNTER
2864  011326  001376                      BNE     4$              ;UNTIL LOOP COUNTER EQUALS 0
2865  011330  005777  171436              TST     @RBUF           ;CLEAR RECEIVER BUFFER
2866  011334  010377  171456              MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
2867
2868
2869
2870
2871                                      ;;*******************************************************************
2872                                      ;*TEST 52        TEST THAT RCVR INTERRUPTS ONLY WHEN ENABLED
2873                                      ;;*******************************************************************
2874  011340  000004              TST52:  SCOPE
2875  011342  012737  000052  001104      MOV     #52,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2876  011350  042777  000100  171416      BIC     #BIT6,@TCSR     ;DISABLE TRANSMIT INTERRUPTS
2877  011356  042777  000100  171404      BIC     #BIT6,@RCSR     ;DISABLE RECEIVER INTERRUPTS
2878  011364  017703  171422              MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
2879  011370  012777  011420  171414      MOV     #2$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
2880  011376  004737  015056              JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2881  011402  000140                      .WORD   140
2882  011404  005077  171366              CLR     @TBUF           ;SEND A CHARACTER
2883  011410  105777  171354      1$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
2884  011414  100375                      BPL     1$
2885  011416  000402                      BR      3$              ;CONTINUE TEST
2886
2887  011420  022626              2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2888  011422  104125                      ERROR   125
2889                                                              ;RECEIVER INTERRPUTS WITH INT. ENABLE CLEAR
2890
2891  011424  012777  011444  171360  3$: MOV     #4$,@RVECT      ;POINT RCV VECTOR TO END OF TEST
2892  011432  052777  000100  171330      BIS     #BIT6,@RCSR     ;ENABLE RCV INTERRUPTS
2893  011440  000240                      NOP                     ;GIVE ANY INTERRUPTS TIME
2894  011442  104126                      ERROR   126
2895                                                              ;RCVR DID NOT INTERRUPT
2896
2897  011444  042777  000100  171316  4$: BIC     #BIT6,@RCSR     ;DISABLE INTERRUPTS
2898  011452  022626                      CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2899  011454  010377  171332              MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
```

```
                                ;;*****************************************************************
                                ;*TEST 53        TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
                                ;;*****************************************************************
011460  000004          TST53:  SCOPE
011462  012737  000053  001104          MOV     #53,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
011470  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
011474  000340                          .WORD   340
011476  017703  171310                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
011502  012777  011534  171302          MOV     #2$,@RVECT      ;POINT RCVR VECTOR TO ERROR REPORT
011510  005077  171262                  CLR     @TBUF           ;SEND A CHARACTER
011514  105777  171250          1$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
011520  100375                          BPL     1$
011522  052777  000100  171240          BIS     #BIT6,@RCSR     ;ENABLE INTERRUPTS
011530  000240                          NOP                     ;GIVE TIME FOR INTERRUPT
011532  000402                          BR      3$              ;CONTINUE TEST
011534  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
011536  104127                          ERROR   127
                                                                ;RCVR INTERRUPTS AT PRIORITY 7.
011540  042777  000100  171222  3$:     BIC     #BIT6,@RCSR     ;CLEAR INTERRUPT ENABLE
011546  012777  011566  171236          MOV     #4$,@RVECT      ;POINT RCVR VECTOR TO ERROR REPORT
011554  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
011560  000140                          .WORD   140
011562  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPT
011564  000402                          BR      5$              ;BR TO END OF TEST, IF NO INTERRUPT
011566  022626                  4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
011570  104130                          ERROR   130
                                                                ;RCVR INTERRUPT REQUEST PASSED WITH BIT6 CLEAR
011572  010377  171214          5$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
                                ;;*****************************************************************
                                ;*TEST 54        TEST SLU2 RECEIVER FOR DOUBLE INTERRUPTS
                                ;;*****************************************************************
011576  000004          TST54:  SCOPE
011600  012737  000054  001104          MOV     #54,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
011606  017703  171200                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
011612  017704  171176                  MOV     @RPSW,R4        ;SAVE RECEIVE PSW VECTOR
011616  012777  011664  171166          MOV     #2$,@RVECT      ;POINT RCV VECTOR TO CONTINUE TEST
011624  012777  000340  171162          MOV     #340,@RPSW      ;SET PRIORITY TO 7 AFTER INTERRUPT
011632  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
011636  000140                          .WORD   140
011640  005077  171132                  CLR     @TBUF           ;SEND A CHARACTER
011644  105777  171120          1$:     TSTB    @RCSR           ;WAIT FOR RCVR DONE
011650  100375                          BPL     1$
011652  052777  000100  171110          BIS     #BIT6,@RCSR     ;ENABLE RCV INTERRUPTS
011660  000240                          NOP                     ;GIVE SOME TIME
011662  104131                          ERROR   131
                                                                ;RCVR INTERRUPT DID NOT OCCUR
011664  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
011666  012777  011720  171116          MOV     #3$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
```

```
2956  011674  004737  015056                  JSR     PC,WRPSW        ;RESET PSW TO PRIORITY 3
2957  011700  000140                          .WORD   140
2958  011702  000240                          NOP                     ;GIVE SOME TIME
2959  011704  042777  000100  171056          BIC     #BIT6,@RCSR     ;CLEAR INTERRUPT ENABLE
2960  011712  010477  171076                  MOV     R4,@RPSW        ;RESTORE RECEIVE PSW VECTOR
2961  011716  000402                          BR      4$              ;BR TO END OF TEST
2962
2963  011720  022626                  3$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2964  011722  104132                          ERROR   132
2965                                                                  ;RECEIVER RE-INTERRUPTED
2966  011724  010377  171062          4$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
2967
2968
2969                                  ;;*******************************************************************
2970                                  ;*TEST 55        TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
2971                                  ;;*******************************************************************
2972  011730  000004                  TST55:  SCOPE
2973  011732  012737  000055  001104          MOV     #55,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2974  011740  004737  015056                  JSR     PC,WRPSW        ;SET PSW PRIORITY TO 7
2975  011744  000340                          .WORD   340
2976  011746  017703  171040                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
2977  011752  012777  012024  171032          MOV     #2$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
2978  011760  052777  000100  171002          BIS     #BIT6,@RCSR     ;SET RCVR INTERRUPT ENABLE
2979  011766  005077  171004                  CLR     @TBUF           ;SEND A CHARACTER
2980  011772  105777  170772          1$:     TSTB    @RCSR           ;WAIT FOR DONE (INTERRUPT)
2981  011776  100375                          BPL     1$
2982  012000  005777  170766                  TST     @RBUF           ;READ RBUF TO CLEAR PENDING INTERRPUT
2983  012004  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2984  012010  000140                          .WORD   140
2985  012012  000240                          NOP                     ;ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
2986  012014  042777  000100  170746          BIC     #BIT6,@RCSR     ;NO INTERRUPT-CLEAR INT. ENABLE
2987  012022  000402                          BR      3$
2988
2989  012024  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2990  012026  104133                          ERROR   133             ;READING RBUF DID NOT CLEAR INTERRUPT
2991
2992  012030  010377  170756          3$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
2993
2994
2995
2996                                  ;;*******************************************************************
2997                                  ;*TEST 56        TEST SLU2 THAT RESET CLEARS RECEIVE INTERRUPT
2998                                  ;;*******************************************************************
2999  012034  000004                  TST56:  SCOPE
3000  012036  012737  000056  001104          MOV     #56,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3001  012044  032737  000001  001120          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
3002  012052  001403                          BEQ     70$             ;IF NO THEN DO TEST
3003  012054  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
3004  012060  001037                          BNE     TST57           ;IF NO THEN SKIP TO NEXT TEST
3005  012062                          70$:
3006  012062  000005                          RESET                   ;CLEAR EVERYTHING
3007  012064  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
3008  012070  000340                          .WORD   340
3009  012072  017703  170714                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
3010  012076  012777  012150  170706          MOV     #2$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
3011  012104  052777  000100  170656          BIS     #BIT6,@RCSR     ;SET RCV INTERRUPT ENABLE
```

```
3012  012112  012777  000377  170656          MOV     #377,@TBUF      ;SEND AN ALL 1'S CHARACTER
3013  012120  105777  170644          1$:     TSTB    @RCSR           ;WAIT FOR RCV DONE
3014  012124  100375                          BPL     1$
3015  012126  000005                          RESET                   ;CLEAR RCV INTERRUPT & RBUF
3016  012130  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
3017  012134  000140                                  .WORD   140
3018  012136  000240                          NOP                     ;ALLOW TIME FOR AN ERRONEOUS INTERRUPT
3019  012140  042777  000100  170622          BIC     #BIT6,@RCSR     ;NO INTERRUPT-CLEAR INT. ENABLE
3020  012146  000402                          BR      3$              ;CONTINUE TEST
3021
3022
3023  012150  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
3024  012152  104134                          ERROR   134             ;RESET DID NOT CLEAR RCVR INTERRUPT
3025
3026  012154  010377  170632          3$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
3027
3028
3029                                  ;;*******************************************************************
3030                                  ;*TEST 57        TEST SLU2 THAT 'OVERRUN & ERROR'' BITS CAN BE SET
3031                                  ;;*******************************************************************
3032  012160  000004                  TST57:  SCOPE
3033  012162  012737  000057  001104          MOV     #57,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3034  012170  032777  002000  166642          BIT     #BIT10,@SWR     ;IS THIS TEST DISABLED
3035  012176  001023                          BNE     TST60           ;IF NOT ENABLED, BR TO NEXT TEST
3036  012200  012700  000003                  MOV     #3,R0           ;SET CHARACTER COUNT TO SEND 3 CHAR.
3037  012204  005077  170566          1$:     CLR     @TBUF           ;LOAD TRANSMIT BUFFER
3038  012210  105777  170560          2$:     TSTB    @TCSR           ;WAIT FOR TRANSMIT DONE
3039  012214  100375                          BPL     2$
3040  012216  005300                          DEC     R0              ;DECREMENT CHARACTER COUNT
3041  012220  001371                          BNE     1$              ;BR IF ALL CHARACTERS NOT TRANSMITTED
3042  012222  032777  040000  170542          BIT     #BIT14,@RBUF    ;TEST FOR ''OR'' ERROR FLAG
3043  012230  001001                          BNE     3$              ;BR, IF SET
3044  012232  104135                          ERROR   135
3045                                                                  ;''OR'' ERROR FLAG DID NOT SET
3046
3047  012234  032777  100000  170530  3$:     BIT     #BIT15,@RBUF    ;TEST ''ERROR'' FLAG
3048  012242  001001                          BNE     4$              ;BR, IF SET
3049  012244  104136                          ERROR   136
3050                                                                  ;''ERROR'' FLAG DID NOT SET WITH ''OR'' FLAG
3051  012246                          4$:
3052
3053
3054                                  ;;*******************************************************************
3055                                  ;*TEST 60        TEST THAT BREAK TRANSMITS ALL ZEROES
3056                                  ;;*******************************************************************
3057  012246  000004                  TST60:  SCOPE
3058  012250  012737  000060  001104          MOV     #60,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3059  012256  012777  177777  170512          MOV     #-1,@TBUF       ;TRANSMIT ALL ONES TO RCVR
3060  012264  105777  170500          1$:     TSTB    @RCSR           ;WAIT FOR RCVR DONE
3061  012270  100375                          BPL     1$
3062  012272  005777  170474                  TST     @RBUF           ;CLEAR DONE (LEAVING ALL ONES IN RBUF)
3063  012276  052777  000001  170470          BIS     #BIT0,@TCSR     ;TRANSMIT BREAK
3064  012304  005000                          CLR     R0              ;CLEAR A TIMER
3065  012306  105777  170456          2$:     TSTB    @RCSR           ;WAIT FOR RCVR DONE
3066  012312  100406                          BMI     CONT41          ;BR IF DONE
3067  012314  005200                          INC     R0              ;IF NOT, INCREMENT TIMER
```

```
3068  012316  001373                          BNE     2$              ;BR IF TIME REMAINS
3069
3070  012320  042777  000001  170446          BIC     #BIT0,@TCSR       ;CLEAR BREAK BITS
3071  012326  104137                          ERROR   137             ;BREAK DID NOT TRANSMIT ANYTHING
3072
3073  012330  105777  170436          CONT41: TSTB    @RBUF           ;CHECK RECEIVE BUFFER FOR ZERO
3074  012334  001404                          BEQ     3$              ;BR, IF ZERO
3075  012336  042777  000001  170430          BIC     #BIT0,@TCSR       ;CLEAR BREAK BITS
3076
3077  012344  104137                          ERROR   137             ;BREAK DID NOT TRANSMIT ALL ZEROES
3078
3079  012346  042777  000001  170420  3$:     BIC     #BIT0,@TCSR       ;CLEAR BREAK BITS
3080
3081
3082                                  ;;*******************************************************************
3083                                  ;*TEST 61        TEST THAT "FR" ERROR CAN BE SET DURING BREAK
3084                                  ;;*******************************************************************
3085  012354  000004                  TST61:  SCOPE
3086  012356  012737  000061  001104          MOV     #61,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3087  012364  032777  002000  166446          BIT     #BIT10,@SWR     ;IS THIS TEST DISABLED
3088  012372  001025                          BNE     TST62           ;BR TO NEXT TEST, IF DISABLED
3089  012374  052777  000001  170372          BIS     #BIT0,@TCSR     ;SEND BREAK
3090  012402  005077  170370                  CLR     @TBUF           ;TRANSMIT A CHARACTER TO TIME BREAK
3091  012406  105777  170356          1$:     TSTB    @RCSR           ;WAIT FOR RCVR DONE
3092  012412  100375                          BPL     1$
3093  012414  042777  000001  170352          BIC     #BIT0,@TCSR       ;CLEAR BREAK BITS
3094  012422  032777  020000  170342          BIT     #BIT13,@RBUF    ;CHECK FOR FRAMING ERROR FLAG
3095  012430  001001                          BNE     2$              ;BR, IF SET
3096
3097  012432  104140                          ERROR   140
3098                                                                  ;BREAK DID NOT SET FRAMING ERROR
3099  012434  032777  100000  170330  2$:     BIT     #BIT15,@RBUF    ;TEST "ERROR" FLAG
3100  012442  001001                          BNE     3$              ;BR, IF SET
3101
3102  012444  104141                          ERROR   141
3103                                                                  ;"ERROR" FLAG DID NOT SET WITH "OR" FLAG
3104  012446                          3$:
3105                                  ;;*******************************************************************
3106                                  ;*TEST 62        TEST DATA PATHS USING WRAP CABLE
3107                                  ;;*******************************************************************
3108  012446  000004                  TST62:  SCOPE
3109  012450  012737  000062  001104          MOV     #62,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3110  012456  032777  000200  166354          BIT     #BIT7,@SWR      ;IS THIS TEST ENABLED
3111  012464  001027                          BNE     TST63           ;BR, IF NOT
3112  012466  005001                          CLR     R1              ;CLEAR REGISTER FOR TEST DATA
3113                                                                  ;TRANSMIT A BINARY COUNT PATTERN - UP
3114                                                                  ;TO THE BIT POSITION INDICATED BY THE
3115                                                                  ;CONTENTS OF LOCATION "$USWR"
3116  012470  105201                  1$:     INCB    R1              ;INCREMENT THE TEST DATA
3117  012472  010177  170300                  MOV     R1,@TBUF        ;XMIT A CHARACTER
3118  012476  005000                          CLR     R0              ;CLEAR A TIMER .
3119  012500  105777  170264          2$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
3120  012504  100403                          BMI     3$              ;BR IF DONE
3121  012506  005200                          INC     R0              ;INCREMENT TIMER IF NOT
3122  012510  001373                          BNE     2$              ;BR IF TIME REMAINS
3123
```

```
3124  012512  104056                    ERROR   56              ;RECEIVER DONE NOT SET
3125
3126  012514  017702  170252    3$:     MOV     @RBUF,R2        ;GET RECEIVED CHARACTER
3127  012520  020102                    CMP     R1,R2           ;COMPARE DATA
3128  012522  001003                    BNE     4$              ;BR, IF NON-COMPARE
3129  012524  105701                    TSTB    R1              ;TEST XMIT DATA FOR ZERO
3130  012526  001406                    BEQ     TST63           ;BR, IF FINISHED
3131  012530  000757                    BR      1$              ;CONTINUE IF NOT
3132  012532  010137  001024    4$:     MOV     R1,$GDDAT       ;STORE EXPECTED DATA
3133  012536  010237  001026            MOV     R2,$BDDAT       ;STORE RECEIVED DATA
3134
3135  012542  104142                    ERROR   142             ;DATA COMPARE ERROR WITH WRAP CABLE
3136
3137
```

```
3138  012544                                  LTCIT:
3139
3140                                          ;;*****************************************************************
3141                                          ;*TEST 63       TEST THAT THE REAL TIME CLOCK INTERRUPTS PROPERLY
3142                                          ;;*****************************************************************
3143  012544  000004                  TST63:  SCOPE
3144  012546  012737  000063  001104          MOV     #63,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3145  012554  032737  000004  001156          BIT     #BIT2,@#$DEVM   ;DO THESE TESTS FOR THIS DEVICE?
3146  012562  001002                          BNE     99$             ;F YES CONTINUE WITH TESTS
3147  012564  000137  013600                  JMP     BLAST           ;IF NO GO TO START OF NEXT SET OF TESTS.
3148  012570                          99$:
3149  012570  004737  015056                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
3150  012574  000340                                  .WORD   340
3151  012576  017703  170230                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
3152  012602  017704  170226                  MOV     @RTCPSW,R4      ;SAVE LINE CLOCK PSW VECTOR
3153  012606  012777  012650  170216          MOV     #2$,@RTCVT      ;SET RTC INTERRUPT VECTOR TO ERROR REPORT
3154  012614  012777  000340  170212          MOV     #340,@RTCPSW    ;KEEP PRIORITY AT 7
3155  012622  042777  000200  170160          BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
3156  012630  052777  000100  170152          BIS     #BIT6,@LKS      ;SET INTERRUPT ENABLE
3157  012636  105777  170146          1$:     TSTB    @LKS            ;WAIT FOR RTC DONE(INTERRUPT REQUEST)
3158  012642  100375                          BPL     1$
3159  012644  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPTS
3160  012646  000402                          BR      3$              ;BR, IF NO INTERRUPT OCCURS
3161
3162  012650  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
3163  012652  104143                          ERROR   143             ;RTC INTERRUPTS AT PRIORITY 7
3164
3165  012654  005077  170130          3$:     CLR     @LKS    ;DISABLE RTC INTERRUPTS & CLEAR DONE
3166  012660  012777  012706  170144          MOV     #4$,@RTCVT      ;SET RTC INTERRUPT VECTOR FOR ERROR
3167  012666  004737  015056                  JSR     PC,WRPSW        ;CHANGE PSW TO PRIORITY 5
3168  012672  000240                                  .WORD   240
3169  012674  105777  170110          20$:    TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
3170  012700  100375                          BPL     20$
3171  012702  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPT
3172  012704  000402                          BR      5$              ;IF NO INTERRUPT - BR TO CONTINUE TEST
3173
3174  012706  022626                  4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
3175  012710  104144                          ERROR   144             ;RTC INTERRUPTS WITH INTERRUPTS DISABLED
3176
3177  012712  012777  012746  170112  5$:     MOV     #7$,@RTCVT      ;POINT RTC VECTOR TO END OF TEST
3178  012720  042777  000200  170062          BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
3179  012726  052777  000100  170054          BIS     #BIT6,@LKS      ;ALLOW INTERRUPTS
3180  012734  105777  170050          6$:     TSTB    @LKS            ;WAIT FOR RTC DONE
3181  012740  100375                          BPL     6$
3182  012742  000240                          NOP                     ;GIVE TIME FOR INTERRUPT
3183
3184  012744  104145                          ERROR   145             ;RTC INTERRUPT DID NOT OCCUR
3185
3186  012746  022626                  7$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
3187  012750  042777  000100  170032          BIC     #BIT6,@LKS      ;DISABLE INTERRUPTS
3188  012756  010377  170050                  MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
3189  012762  010477  170046                  MOV     R4,@RTCPSW      ;RESTORE LINE CLOCK PSW VECTOR
3190
3191
3192
3193                                          ;;*****************************************************************
```

```
3194
3195
3196  012766  000004
3197  012770  012737  000064  001104          ;*TEST 64       TEST RTC FOR DOUBLE INTERRUPTS
3198  012776  032737  000001  001120          ;*****************************************************************
3199  013004  001403                          TST64:  SCOPE
3200  013006  005737  001106                          MOV     #64,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3201  013012  001052                          70$:    BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
3202  013014                                           BEQ     70$             ;IF NO THEN DO TEST
3203  013014  017703  170012                          TST     @#$PASS         ;IS THIS FIRST PASS
3204  013020  017704  170010                          BNE     TST65           ;IF NO THEN SKIP TO NEXT TEST
3205  013024  012777  013074  170000                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
3206  013032  012777  000340  167774                  MOV     @RTCPSW,R4      ;SAVE LINE CLOCK PSW VECTOR
3207  013040  004737  015056                          MOV     #2$,@RTCVT      ;SET UP RTC INTERRUPT VECTOR
3208  013044  000240                                  MOV     #340,@RTCPSW    ;DISALLOW INTERRUPTS AFTER THE INTERRUPT
3209  013046  042777  000200  167734                  JSR     PC,WRPSW        ;SET PRIORITY TO 5
3210  013054  052777  000100  167726                  .WORD   240
3211  013062  105777  167722                  1$:     BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
3212  013066  100375                                   BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
3213  013070  000240                                  TSTB    @LKS            ;WAIT FOR DONE
3214                                                   BPL     1$
3215  013072  104146                                  NOP                     ;GIVE TIME FOR ANY INTERRUPT
3216
3217  013074  022626                                  ERROR   146             ;RTC INTERRUPT DID NOT OCCUR
3218  013076  012777  013116  167726          2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
3219  013104  004737  015056                          MOV     #3$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
3220  013110  000240                                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 5
3221  013112  000240                                  .WORD   240
3222  013114  000402                                  NOP                     ;GIVE SOME TIME FOR AN INTERRUPT
3223                                                   BR      4$              ;NO INTERRUPT - BR TO END OF TEST
3224  013116  022626                          3$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
3225  013120  104147                                  ERROR   147             ;INTERRUPT SEQUENCE DID NOT CLEAR
3226                                                                           ;INTERRUPT REQUEST
3227
3228  013122  042777  000100  167660          4$:     BIC     #BIT6,@LKS      ;DISABLE CLOCK INTERRUPTS
3229  013130  010377  167676                          MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
3230  013134  010477  167674                          MOV     R4,@RTCPSW      ;RESTORE LINE CLOCK PSW VECTOR
3231
3232
3233                                           ;*****************************************************************
3234                                           ;*TEST 65       TEST THAT RTC INTERRUPT CLEARS WITH RESET
3235                                           ;*****************************************************************
3236  013140  000004                          TST65:  SCOPE
3237  013142  012737  000065  001104                  MOV     #65,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3238  013150  032737  000001  001120                  BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
3239  013156  001403                                  BEQ     70$             ;IF NO THEN DO TEST
3240  013160  005737  001106                          TST     @#$PASS         ;IS THIS FIRST PASS
3241  013164  001036                                  BNE     TST66           ;IF NO THEN SKIP TO NEXT TEST
3242  013166                                  70$:
3243  013166  004737  015056                          JSR     PC,WRPSW        ;SET PRIORITY TO 7
3244  013172  000340                                  .WORD   340
3245  013174  017703  167632                          MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
3246  013200  012777  013252  167624                  MOV     #2$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
3247  013206  042777  000200  167574                  BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
3248  013214  052777  000100  167566                  BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
3249  013222  105777  167562                  1$:     TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
```

```
3250  013226  100375                        BPL     1$
3251  013230  000005                        RESET                   ;CLEAR PENDING INTERRUPT WITH RESET
3252  013232  004737  015056                JSR     PC,WRPSW        ;SET PRIORITY TO 5
3253  013236  000240                        .WORD   240
3254  013240  000240                        NOP                     ;GIVE TIME FOR ANY INTERRUPT
3255  013242  042777  000100  167540        BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
3256  013250  000402                        BR      3$              ;BR TO END OF TEST
3257
3258  013252  022626                2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
3259  013254  104150                        ERROR   150             ;RESET DID NOT CLEAR INTERRUPT
3260
3261  013256  010377  167550        3$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
3262
3263
3264                                ;;*******************************************************************
3265                                ;*TEST 66        TEST THAT RTC INTERRUPT CLEARS BY CLEARING BIT7 OF LKS
3266                                ;;*******************************************************************
3267  013262  000004                TST66:  SCOPE
3268  013264  012737  000066  001104        MOV     #66,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3269  013272  032737  000001  001120        BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
3270  013300  001403                        BEQ     70$             ;IF NO THEN DO TEST
3271  013302  005737  001106                TST     @#$PASS         ;IS THIS FIRST PASS
3272  013306  001043                        BNE     TST67           ;IF NO THEN SKIP TO NEXT TEST
3273  013310                        70$:
3274  013310  004737  015056                JSR     PC,WRPSW        ;SET PRIORITY TO 7
3275  013314  000340                        .WORD   340
3276  013316  017703  167510                MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
3277  013322  012777  013400  167502        MOV     #2$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
3278  013330  042777  000200  167452        BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
3279  013336  052777  000100  167444        BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
3280  013344  105777  167440        1$:     TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
3281  013350  100375                        BPL     1$
3282  013352  042777  000200  167430        BIC     #BIT7,@LKS      ;CLEAR DONE & INTERRUPT
3283  013360  004737  015056                JSR     PC,WRPSW        ;ALLOW INTERRUPTS
3284  013364  000240                        .WORD   240
3285  013366  000240                        NOP                     ;GIVE TIME FOR ANY INTERRUPT
3286  013370  042777  000100  167412        BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
3287  013376  000402                        BR      3$              ;BR TO END OF TEST
3288
3289
3290  013400  022626                2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
3291  013402  104151                        ERROR   151             ;CLEARING BIT7 OF LKS DID NOT CLEAR INTERRUPT
3292
3293  013404  010377  167422        3$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
3294  013410  004737  015056                JSR     PC,WRPSW        ;RESTORE PRIORITY TO 7
3295  013414  000340                        .WORD   340
3296
3297
3298                                ;;*******************************************************************
3299                                ;*TEST 67        TEST CLOCK REPEATABILITY
3300                                ;;*******************************************************************
3301  013416  000004                TST67:  SCOPE
3302  013420  012737  000067  001104        MOV     #67,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3303  013426  032737  000001  001120        BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
3304  013434  001403                        BEQ     70$             ;IF NO THEN DO TEST
3305  013436  005737  001106                TST     @#$PASS         ;IS THIS FIRST PASS
```

```
3306  013442  001056                          BNE     TST70           ;IF NO THEN SKIP TO NEXT TEST
3307  013444                          70$:
3308  013444  042777  000100  167336          BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
3309  013452  005000                          CLR     R0              ;CLEAR A TIMER
3310  013454  012701  177777                  MOV     #-1,R1          ;SET A FLAG INDICATING FIRST PASS THRU THIS LOOP
3311  013460  005002                  1$:     CLR     R2              ;CLEAR CLOCK COUNTER
3312  013462  005077  167322                  CLR     @LKS            ;CLEAR DONE
3313  013466  105777  167316          2$:     TSTB    @LKS            ;SYNC ON DONE
3314  013472  100375                          BPL     2$
3315  013474  005077  167310                  CLR     @LKS            ;CLEAR DONE
3316  013500  105777  167304          3$:     TSTB    @LKS            ;IS CLOCK DONE?
3317  013504  100003                          BPL     4$              ;BR IF NOT , TO INCREMENT TIMER
3318  013506  005202                          INC     R2              ;IF DONE, INCREMENT CLOCK COUNT
3319  013510  005077  167274                  CLR     @LKS            ;CLEAR DONE
3320  013514  005200                  4$:     INC     R0              ;INCREMENT TIMER
3321  013516  001370                          BNE     3$              ;BR IF TIME REMAINS
3322  013520  005201                          INC     R1              ;INCREMENT LOOP PASS FLAG
3323  013522  001003                          BNE     CMPARE          ;BR IF TWO PASSES HAVE BEEN MADE
3324  013524  010237  013574                  MOV     R2,FIRST        ;IF NOT, STORE FIRST CLOCK COUNT
3325  013530  000753                          BR      1$              ;DO LOOP AGAIN
3326  013532  013701  013574          CMPARE: MOV     FIRST,R1        ;RECALL FIRST CLOCK COUNT
3327  013536  160201                          SUB     R2,R1           ;CALCULATE DIFFERENCE OF TWO COUNTS
3328  013540  100001                          BPL     TOLER           ;IF POSITIVE,SKIP NEGATION OF DIFFERENCE
3329  013542  005401                          NEG     R1              ;MAKE DIFFERENCE A POSITIVE NUMBER
3330  013544  020127  000002          TOLER:  CMP     R1,#2           ;COMPARE DIFFERENCE WITH DESIRED TOLFRANCE
3331  013550  003403                          BLE     5$              ;BR, IF LOWER/EQUAL TO TOLERANCE
3332
3333  013552  010237  013576                  MOV     R2,SECND        ;STORE SECOND COUNT
3334  013556  104152                          ERROR   152             ;CLOCK REPEATABILITY ERROR
3335
3336  013560  032777  000040  165252  5$:     BIT     #BIT5,@SWR      ;CLOCK TESTS ONLY?
3337  013566  001404                          BEQ     TST70           ;BR IF NOT
3338  013570  000137  014740                  JMP     $EOP            ;ELSE, JUMP TO END OF PASS ROUTINE
3339
3340  013574  000000          FIRST:  0
3341  013576  000000          SECND:  0
```

```
3342  013600                                  BLAST:
3343
3344                                          ;;*****************************************************************
3345                                          ;*TEST 70       TEST ALL INTERNAL OPTIONS SIMULTANEOUSLY
3346                                          ;;*****************************************************************
3347  013600  000004                          TST70:  SCOPE
3348  013602  012737  000070  001104                 MOV     #70,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3349  013610  032737  000001  001120                 BIT     #BIT0,@#$ENV    ;ARE WE RUNNING UNDER APT
3350  013616  001405                                 BEQ     70$             ;IF NO DO TEST
3351  013620  005737  001106                        TST     @#$PASS         ;IS THIS FIRST PASS
3352  013624  001402                                 BEQ     70$             ;IF YES DO TEST
3353  013626  000137  014740                        JMP     $EOP            ;IF NO DO NOT DO TEST
3354  013632  000005                          70$:    RESET                   ;CLEAR EVERY BODY
3355  013634  012737  000340  177776                 MOV     #340,PS         ;SET PROCESSOR PRIORITY TO 7
3356  013642  017737  167150  001060                 MOV     @TVECT,$TMP0
3357  013650  017737  167136  001062                 MOV     @RVECT,$TMP1
3358  013656  017737  167144  001064                 MOV     @CTVECT,$TMP2
3359  013664  017737  167132  001066                 MOV     @CRVECT,$TMP3
3360  013672  017737  167134  001070                 MOV     @RTCVT,$TMP4
3361  013700  005037  014726                        CLR     XMTCT1          ;INITIALIZE COUNTERS
3362  013704  005037  014730                        CLR     XMTCT2
3363  013710  005037  014732                        CLR     RECCT1
3364  013714  005037  014734                        CLR     RECCT2
3365  013720  005037  014736                        CLR     COUNT
3366  013724  012777  014154  167074                 MOV     #XMIT1,@CTVECT  ;SET UP SLU1 TRANSMIT VECTOR
3367  013732  012777  000340  167070                 MOV     #340, @CTPSW    ;AND PSW
3368  013740  012777  014210  167054                 MOV     #REC1, @CRVECT  ;SET UP SLU1 RECEIVER VECTOR
3369  013746  012777  000340  167050                 MOV     #340, @CRPSW    ;AND PSW
3370  013754  012777  014270  167034                 MOV     #XMIT2, @TVECT  ;SET UP SLU2 TRANSMIT VECTOR
3371  013762  012777  000340  167030                 MOV     #340, @TPSW     ;AND PSW
3372  013770  012777  014324  167014                 MOV     #REC2, @RVECT   ;SET UP SLU2 RECEIVER VECTOR
3373  013776  012777  000340  167010                 MOV     #340, @RPSW     ;AND PSW
3374  014004  012777  014144  167020                 MOV     #TICKER, @RTCVT ;SET UP RTC VECTOR
3375  014012  012777  000340  167014                 MOV     #340, @RTCPSW   ;AND PSW
3376  014020  032737  000001  001156                 BIT     #BIT0, $DEVM    ;IS SLU1 UNDER TEST
3377  014026  001413                                 BEQ     1$              ;IF NO DON'T TURN IT ON
3378  014030  052777  000004  166746                 BIS     #BIT2, @CTCSR   ;ENABLE SLU1 MAINTENANCE WRAP
3379  014036  052777  000100  166740                 BIS     #BIT6, @CTCSR   ;ENABLE SLU1 XMIT INTERRUPT
3380  014044  052777  000100  166726                 BIS     #BIT6, @CRCSR   ;ENABLE SLU1 RECEIVER INTERRUPT
3381  014052  012702  030450                        MOV     #BUF1, R2       ;SET UP RECEIVER BUFFER
3382  014056  032737  000002  001156  1$:     BIT     #BIT1, $DEVM    ;IS SLU2 UNDER TEST
3383  014064  001410                                 BEQ     2$              ;IF NO DON'T SET IT UP
3384  014066  052777  000100  166700                 BIS     #BIT6, @TCSR    ;ENABLE SLU2 XMIT INTERRUPT
3385  014074  052777  000100  166666                 BIS     #BIT6, @RCSR    ;ENABLE SLU2 RECEIVER INTERRUPT
3386  014102  012703  031050                        MOV     #BUF2, R3       ;SET UP RECEIVER BUFFER
3387  014106  032737  000004  001156  2$:     BIT     #BIT2, $DEVM    ;IS LTC UNDER TEST
3388  014114  001403                                 BEQ     3$              ;IF NO DON'T SET IT UP
3389  014116  052777  000100  166664                 BIS     #BIT6, @LKS     ;ENABLE RTC INTERRUPTS
3390  014124  012700  177777          3$:     MOV     #-1, R0         ;INITIALIZE DATA FOR SLU1(1ST CHR. 0)
3391  014130  012701  177777                        MOV     #-1, R1         ;INITIALIZE DATA FOR SLU2(1ST CHR. 0)
3392  014134  005037  177776                        CLR     PS              ;DROP PROCESSOR PRIORITY TO ZERO
3393  014140  000001                  WAITIO: WAIT                    ;WAIT FOR INTERRUPT
3394  014142  000776                          BR      WAITIO
3395
3396  014144  005237  014736          TICKER: INC     COUNT           ;UPDATE COUNT
3397  014150  000137  014400                  JMP     IOHAND          ;GO TO INTERRUPT HANDLER
```

```
3398  014154  005237  014726          XMIT1:  INC     XMTCT1          ;UPDATE XMIT INTERRUPT COUNT
3399  014160  005200                          INC     R0              ;UPDATE XMIT DATA
3400  014162  010077  166620                  MOV     R0, @CTBUF      ;SEND NEXT CHARACTER
3401  014166  023727  014726  000400          CMP     XMTCT1,#400     ;IF 256 CHARACTERS HAVE NOT BEEN
3402  014174  002403                          BLT     1$              ;TRANSFERRED CONTINUE
3403  014176  042777  000100  166600          BIC     #BIT6, @CTCSR   ;IF YES TURN OFF TRANSMITTER INTERRPUTS FIRST
3404  014204  000137  014400          1$:     JMP     IOHAND          ;GO TO INTERRUPT HANDLER
3405
3406  014210  005237  014732          REC1:   INC     RECCT1          ;UPDATE RECEIVER INTERUPT COUNT
3407  014214  005777  166562                  TST     @CRBUF          ;BIT 15 SET IF ANY ERRORS OCCURRED
3408  014220  100017                          BPL     3$              ;IF BIT IS CLEAR NO ERROR
3409  014222  017737  166554  001026          MOV     @CRBUF, $BDDAT  ;GET ERROR INFORMATION
3410  014230  000005                          RESET                   ;CLEAR THE WORLD STOP ALL
3411                                                                  ;INTERRUPTS
3412  014232  020227  030450                  CMP     R2, #BUF1       ;WAS MORE THAN 1 WORD TRANSFERRED
3413  014236  003004                          BGT     1$              ;IF YES GET LAST GOOD DATA
3414  014240  012737  177777  001024          MOV     #-1, $GDDAT     ;MAKE GOOD DATA = -1
3415  014246  000403                          BR      2$              ;GO TO ERROR REPORT
3416  014250  116237  177777  001024  1$:     MOVB    -1(R2), $GDDAT  ;GET LAST GOOD DATA
3417  014256  104153                  2$:     ERROR   153             ;RECEIVER STATUS ERROR
3418  014260  117722  166516          3$:     MOVB    @CRBUF, (R2)+   ;GET DATA AND STORE IT
3419  014264  000137  014400                  JMP     IOHAND          ;GO TO INTERRUPT HANDLER
3420
3421  014270  005237  014730          XMIT2:  INC     XMTCT2          ;UPDATE XMIT INTERRUPT COUNT
3422  014274  005201                          INC     R1              ;UPDATE XMIT DATA
3423  014276  010177  166474                  MOV     R1, @TBUF       ;SEND NEXT CHARACTER
3424  014302  023727  014730  000400          CMP     XMTCT2, #400    ;IF 256 CHARACTERS HAVE NOT
3425  014310  002403                          BLT     1$              ;BEEN TRANSFERRED CONTINUE
3426  014312  042777  000100  166454          BIC     #BIT6, @TCSR    ;ELSE NO MORE XMIT INTERRUPTS
3427  014320  000137  014400          1$:     JMP     IOHAND          ;GO TO INTERRUPT HANDLER
3428
3429  014324  005237  014734          REC2:   INC     RECCT2          ;UPDATE RECEIVER INTERRUPT COUNT
3430  014330  005777  166436                  TST     @RBUF           ;BIT 15 SETS IF ANY ERRORS OCCURRED
3431  014334  100017                          BPL     3$              ;IF BIT IS CLEAR NO ERRORS
3432  014336  017737  166430  001026          MOV     @RBUF, $BDDAT   ;GET ERROR INFORMATION
3433  014344  000005                          RESET                   ;CLEAR THE WORLD - STOP ALL
3434                                                                  ;INTERRUPTS
3435  014346  020327  031050                  CMP     R3, #BUF2       ;WAS MORE THAN 1 WORD TRANSFERRED
3436  014352  003004                          BGT     1$              ;IF YES GET LAST GOOD DATA
3437  014354  012737  177777  001024          MOV     #-1, $GDDAT     ;IF NO MAKE GOOD DATA -1
3438  014362  000403                          BR      2$              ;AND GET TO ERROR REPORT
3439  014364  116337  177777  001024  1$:     MOVB    -1(R3), $GDDAT  ;GET LAST GOOD DATA RECEIVED
3440  014372  104154                  2$:     ERROR   154             ;RECEIVER STATUS ERROR
3441  014374  117723  166372          3$:     MOVB    @RBUF, (R3)+    ;GET DATA AND STORE IT
3442
3443  014400  032737  000004  001156  IOHAND: BIT     #BIT2, $DEVM    ;IS RTC UNDER TEST
3444  014406  001416                          BEQ     1$              ;IF NO CHECK OTHER DEVICES
3445  014410  023727  014736  000074          CMP     COUNT, #74      ;HAS 1 SEC ELAPSED
3446  014416  002427                          BLT     3$              ;IF NO CONTINUE TEST
3447  014420  042777  000100  166356          BIC     #BIT6, @CTCSR   ;IF YES STOP TRANSMISSIONS
3448  014426  042777  000100  166340          BIC     #BIT6, @TCSR    ;
3449  014434  042777  000100  166346          BIC     #BIT6,  @LKS    ;TURN OFF LINE CLOCK
3450  014442  000416                          BR      WAITER
3451  014444  032737  000001  001156  1$:     BIT     #BIT0, $DEVM    ;IS SLU1 UNDER TEST
3452  014452  001405                          BEQ     2$              ;IF NO CHECK FOR SLU2
3453  014454  032777  000100  166322          BIT     #BIT6, @CTCSR   ;IS TRANSMITTER SHUTDOWN
```

```
3454  014462  001405                          BEQ     3$              ;IF NO CONTINUE TEST
3455  014464  000405                          BR      WAITER          ;IF YES GO WAIT FOR POSSIBLE
3456
3457  014466  032777  000100  166300  2$:     BIT     #BIT6, @TCSR    ;IS TRANSMITTER SHUT DOWN
3458  014474  001001                          BNE     WAITER          ;IF YES GO WAIT FOR POSSIBLE
3459                                                                  ;LAST CHARACTER
3460  014476  000002                  3$:     RTI                     ;RETURN FROM INTERRUPT TO AWAIT NEXT
3461
3462  014500  005037  177776          WAITER: CLR     PS              ;MAKE PROCESSOR PRIORITY 0
3463  014504  012705  140000                  MOV     #-40000, R5     ;SET UP LOOP COUNTER
3464  014510  062705  000001          1$:     ADD     #1, R5          ;DO LOOP UNTIL R5 = 0
3465  014514  001375                          BNE     1$
3466  014516  000005                          RESET                   ;STOP EVERYONE SHOULD BE DONE
3467  014520  012706  001000                  MOV     #1000,SP        ;RESET STACK AFTER LAST INTERRUPT
3468
3469  014524  032737  000001  001156  CHECK1: BIT     #BIT0, $DEVM    ;SLU1 UNDER TEST
3470  014532  001424                          BEQ     CHECK2          ;IF NO GO CHECK SLU2 DATA
3471  014534  023737  014726  014732          CMP     XMTCT1,RECCT1   ;# OF XMIT INTERRUPTS = REC INTERRUPTS
3472  014542  001401                          BEQ     1$              ;IF YES GET OVER ERROR AND CHECK DATA
3473  014544  104155                          ERROR   155             ;INTERRUPT COMPARISON ERROR
3474  014546  012702  030450          1$:     MOV     #BUF1, R2       ;POINT TO FIRST DATA
3475  014552  005000                          CLR     R0              ;INITIALIZE TO FIRST DATA XMIT
3476  014554  013704  014726                  MOV     XMTCT1, R4      ;GET # OF BYTES TRANSFERRED
3477  014560  122200                  2$:     CMPB    (R2)+, R0       ;IS RECEIVED DATA = EXPECTED
3478  014562  001406                          BEQ     3$              ;IF YES CONTINUE
3479  014564  114237  001026                  MOVB    -(R2), $BDDAT   ;IF NO GET ERROR INFORMATION
3480  014570  010037  001024                  MOV     R0, $GDDAT
3481  014574  104156                          ERROR   156             ;SLU1 DATA COMPARISON ERROR
3482  014576  005202                          INC     R2              ;IF CONTINUE ON ERROR RESET POINTER
3483  014600  005200                  3$:     INC     R0              ;UPDATE TO NEXT GOOD DATA
3484  014602  077412                          SOB     R4,2$           ;LOOP UNTIL ALL DATA CHECKED
3485  014604  032737  000002  001156  CHECK2: BIT     #BIT1, $DEVM    ;SLU2 UNDER TEST
3486  014612  001424                          BEQ     FINIE           ;IF NO WE'RE DONE
3487  014614  023737  014730  014734          CMP     XMTCT2, RECCT2  ;#OF XMIT INTERRUPTS = REC INTERRUPTS
3488  014622  001401                          BEQ     1$              ;IF YES CHECK DATA
3489  014624  104157                          ERROR   157             ;INTERRUPT COMPARISON ERROR
3490  014626  012703  031050          1$:     MOV     #BUF2, R3       ;INITIALIZE TO FIRST RECEIVED DATA
3491  014632  005001                          CLR     R1              ;INITIALIZE TO FIRST XMIT DATA
3492  014634  013704  014730                  MOV     XMTCT2, R4      ;GET # OF BYTES TRANSFERRED
3493  014640  122301                  2$:     CMPB    (R3)+, R1       ;IS RECEIVED DATA = EXPECTED DATA
3494  014642  001406                          BEQ     3$              ;IF YES CONTINUE TESTING
3495  014644  114337  001026                  MOVB    -(R3), $BDDAT   ;IF NO GET ERROR INFORMATION
3496  014650  010137  001024                  MOV     R1, $GDDAT
3497  014654  104160                          ERROR   160             ;SLU2 DATA COMPARISON ERROR
3498  014656  005203                          INC     R3              ;IF COONTINUE ON ERROR RESET POINTER
3499  014660  005201                  3$:     INC     R1              ;UPDATE TO NEXT GOOD DATA
3500  014662  077412                          SOB     R4,2$           ;LOOP UNTIL ALL DATA CHECKED
3501  014664  013777  001060  166124  FINIE:  MOV     $TMP0, @TVECT   ;RESTORE VECTORS
3502  014672  013777  001062  166112          MOV     $TMP1, @RVECT
3503  014700  013777  001064  166120          MOV     $TMP2, @CTVECT
3504  014706  013777  001066  166106          MOV     $TMP3, @CRVECT
3505  014714  013777  001070  166110          MOV     $TMP4, @RTCVT
3506  014722  000137  014740                  JMP     $EOP            ;FINISHED TESTING GO TO END OF PASS
3507
3508  014726  000000                  XMTCT1: .WORD   0
3509  014730  000000                  XMTCT2: .WORD   0
```

```
3510  014732  000000          RECCT1: .WORD   0
3511  014734  000000          RECCT2: .WORD   0
3512  014736  000000          COUNT:  .WORD   0
3513
```

```
                                .SBTTL  END OF PASS ROUTINE

                                ;;*****************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*TYPE 'END PASS #XXXXX' (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO TST1

014740                          $EOP:
014740  000004                          SCOPE
014742  005037  001002                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
014746  005237  001106                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
014752  042737  100000  001106          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
014760  005327                          DEC     (PC)+           ;;LOOP?
014762  000001          $EOPCT: .WORD   1
014764  003022                  BGT     $DOAGN          ;;YES
014766  012737                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
014770  000001          $ENDCT: .WORD   1
014772  014762                  $EOPCT
014774  104401  015041          TYPE    ,$ENDMG         ;;TYPE 'END PASS #'
015000  013746  001106          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
015004  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
015006  104401  015036          TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
015012  013700  000042  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
015016  001405                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
015020  000005                  RESET                   ;;CLEAR THE WORLD
015022  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
015024  000240                  NOP                     ;;SAVE ROOM
015026  000240                  NOP                     ;;FOR
015030  000240                  NOP                     ;;ACT11
015032                          $DOAGN:
015032  000137                  JMP     @(PC)+          ;;RETURN
015034  003450          $RTNAD: .WORD.  TST1
015036     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
015041     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
015046  050040  051501  020123
015054  000043

015056  011646          WRPSW:  MOV(SP),-(SP)           ;COPY RETUN PC
015060  013616                  MOV     @(SP)+,(SP)     ;MOVE NEW PSW TO STACK
015062  062746  000002          ADD     #2,-(SP)        ;ADJUST JSR RETURN
015066  000002                  RTI                     ;POP RETURN PC & NEW PSW

                                ;SUBROUTINE TO REPORT UNEXPECTED OR ERRONEOUS TRAPS OR INTERRUPTS

015070  012600          CATCH:  MOV     (SP)+,R0        ;GET ADDRESS OF TRAP VECTOR + 4
015072  162700  000004          SUB     #4,R0           ;ADJUST TO POINT TO TRAP ADDRESS
015076  010037  015116          MOV     R0,BDVECT       ;STORE TRAP OR INTERRUPT ADDRESS
015102  016637  000002  015114  MOV     2(SP),OLDPC     ;GET PC WHERE TRAP OR INTERRUPT OCCURRED
015110  104161                  ERROR   161             ;REPORT ERROR

015112  000000                  HALT                    ;PROGRAM MUST BE RESTARTED AT THIS POINT
015114  000000          OLDPC:  .WORD   0
015116  000000          BDVECT: .WORD   0
```

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

015120                                  $ERROR:
015120  105237  001003                  7$:     INCB    $ERFLG          ;SET THE ERROR FLAG
015124  001775                                  BEQ     7$              ;DON'T LET FLAG GO TO ZERO
015126  013777  001002  163706                  MOV     $TSTNM,@DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
015134  005237  001012                          INC     $ERTTL          ;INCREMENT ERROR COUNT
015140  011637  001016                          MOV     (SP),$ERRPC     ;GET ADDRESS OF ERROR INSTRUCTION
015144  162737  000002  001016                  SUB     #2,$ERRPC
015152  117737  163640  001014                  MOVB    @$ERRPC,$ITEMB  ;STRIP AND SAVE THE ERROR ITEM CODE
015160  032777  020000  163652                  BIT     #BIT13,@SWR     ;SKIP TYPEOUT IF SET
015166  001004                                  BNE     20$             ;SKIP TYPEOUTS
015170  004737  015302                          JSR     PC,$ERRTYP      ;GO TO USER ERROR ROUTINE
015174  104401  001075                          TYPE    ,$CRLF
015200                                  20$:
015200  122737  000001  001120                  CMPB    #APTENV,$ENV    ;RUNNING IN APT MODE
015206  001007                                  BNE     2$              ;NO, SKIP APT ERROR REPORT
015210  113737  001014  015222                  MOVB    $ITEMB,21$      ;SET ITEM NUMBER AS ERROR NUMBER
015216  004737  016004                          JSR     PC,$ATY4        ;REPORT FATAL ERROR TO APT
015222     000                          21$:    .BYTE   0
015223     000                                  .BYTE   0
015224  000777                          22$:    BR      22$             ;APT ERROR LOOP
015226  005777  163606                  2$:     TST     @SWR            ;HALT ON ERROR
015232  100001                                  BPL     3$              ;SKIP IF CONTINUE
015234  000000                                  HALT                    ;HALT ON ERROR!
015236  104407                          3$:     CKSWR                   ;TEST FOR CHANGE IN SOFT-SWR
015240  032777  001000  163572                  BIT     #BIT09,@SWR     ;LOOP ON ERROR SWITCH SET?
015246  001402                                  BEQ     4$              ;BR IF NO
015250  013716  001010                          MOV     $LPERR,(SP)     ;FUDGE RETURN FOR LOOPING
015254  005737  001072                  4$:     TST     $ESCAPE         ;CHECK FOR AN ESCAPE ADDRESS
015260  001402                                  BEQ     5$              ;BR IF NONE
015262  013716  001072                          MOV     $ESCAPE,(SP)    ;FUDGE RETURN ADDRESS FOR ESCAPE
015266                                  5$:
015266  022737  015022  000042                  CMP     #$ENDAD,@#42    ;ACT-11 AUTO-ACCEPT?
015274  001001                                  BNE     6$              ;BR IF NO
015276  000000                                  HALT                    ;YES
015300                                  6$:
015300  000002                                  RTI                     ;RETURN
```

```
3620
3621
3622
3623
3624
3625
3626
3627
3628  015302
3629  015302  104401  001075
3630  015306  010046
3631  015310  005000
3632  015312  153700  001014
3633  015316  001004
3634
3635  015320  013746  001016
3636
3637  015324  104402
3638  015326  000426
3639  015330  005300
3640  015332  006300
3641  015334  006300
3642  015336  006300
3643  015340  062700  001160
3644  015344  012037  015354
3645  015350  001404
3646  015352  104401
3647  015354  000000
3648  015356  104401  001075
3649  015362  012037  015372
3650  015366  001404
3651  015370  104401
3652  015372  000000
3653  015374  104401  001075
3654  015400  011000
3655  015402  001004
3656  015404  012600
3657  015406  104401  001075
3658  015412  000207
3659  015414
3660  015414  013046
3661  015416  104402
3662  015420  005710
3663  015422  001770
3664  015424  104401  015432
3665  015430  000771
3666  015432  020040      000
3667        015436
3668

.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.

$ERRTYP:
        TYPE    ,$CRLF          ;;''CARRIAGE RETURN'' & 'LINE FEED''
        MOV     R0,-(SP)        ;;SAVE R0
        CLR     R0              ;;PICKUP THE ITEM INDEX
        BISB    @#$ITEMB,R0
        BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                ;;TYPE THE PC OF THE ERROR
        MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                ;;ERROR ADDRESS
        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
        BR      6$              ;;GET OUT
1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
        ASL     R0              ;;      WORK FOR THE ERROR TABLE
        ASL     R0
        ASL     R0
        ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
        MOV     (R0)+,2$        ;;PICKUP 'ERROR MESSAGE'' POINTER
        BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
        TYPE                    ;;TYPE THE 'ERROR MESSAGE''
2$:     .WORD   0               ;;'ERROR MESSAGE'' POINTER GOES HERE
        TYPE    ,$CRLF          ;;''CARRIAGE RETURN'' & 'LINE FEED''
3$:     MOV     (R0)+,4$        ;;PICKUP 'DATA HEADER'' POINTER
        BEQ     5$              ;;SKIP TYPEOUT IF 0
        TYPE                    ;;TYPE THE 'DATA HEADER''
4$:     .WORD   0               ;;'DATA HEADER'' POINTER GOES HERE
        TYPE    ,$CRLF          ;;''CARRIAGE RETURN'' & 'LINE FEED''
5$:     MOV     (R0),R0         ;;PICKUP 'DATA TABLE'' POINTER
        BNE     7$              ;;GO TYPE THE DATA
6$:     MOV     (SP)+,R0        ;;RESTORE R0
        TYPE    ,$CRLF          ;;''CARRIAGE RETURN'' & 'LINE FEED''
        RTS     PC              ;;RETURN
7$:
        MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
        TST     (R0)            ;;IS THERE ANOTHER NUMBER?
        BEQ     6$              ;;BR IF NO
        TYPE    ,8$             ;;TYPE TWO(2) SPACES
        BR      7$              ;;LOOP
8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
        .EVEN
```

```
3669
3670                                    .SBTTL  POWER DOWN AND UP ROUTINES
3671                                    ;;*****************************************************************
3672                                    ;*POWER DOWN ROUTINE
3673                                    ;;*****************************************************************
3674  015436  012737  015576  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;SET FOR FAST UP
3675  015444  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;PRIO:7
3676  015452  010046                         MOV     R0,-(SP)        ;PUSH R0 ON STACK
3677  015454  010146                         MOV     R1,-(SP)        ;PUSH R1 ON STACK
3678  015456  010246                         MOV     R2,-(SP)        ;PUSH R2 ON STACK
3679  015460  010346                         MOV     R3,-(SP)        ;PUSH R3 ON STACK
3680  015462  010446                         MOV     R4,-(SP)        ;PUSH R4 ON STACK
3681  015464  010546                         MOV     R5,-(SP)        ;PUSH R5 ON STACK
3682  015466  017746  163346                 MOV     @SWR,-(SP)      ;PUSH @SWR ON STACK
3683  015472  010637  015602                 MOV     SP,$SAVR6       ;SAVE SP
3684  015476  012737  015510  000024         MOV     #$PWRUP,@#PWRVEC ;SET UP VECTOR
3685  015504  000000                         HALT
3686  015506  000776                         BR      .-2             ;HANG UP
3687
3688
3689                                    ;;*****************************************************************
3690                                    ;*POWER UP ROUTINE
3691                                    ;;*****************************************************************
3692  015510  012737  015576  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;SET FOR FAST DOWN
3693  015516  013706  015602                 MOV     $SAVR6,SP       ;GET SP
3694  015522  012677  163312                 MOV     (SP)+,@SWR      ;POP STACK INTO @SWR
3695  015526  012605                         MOV     (SP)+,R5
3696  015530  012604                         MOV     (SP)+,R4        ;POP STACK INTO R4
3697  015532  012603                         MOV     (SP)+,R3        ;POP STACK INTO R3
3698  015534  012602                         MOV     (SP)+,R2        ;POP STACK INTO R2
3699  015536  012601                         MOV     (SP)+,R1        ;POP STACK INTO R1
3700  015540  012600                         MOV     (SP)+,R0        ;POP STACK INTO R0
3701  015542  012737  015436  000024         MOV     #$PWRDN,@#PWRVEC ;SET UP THE POWER DOWN VECTOR
3702  015550  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;PRIO:7
3703  015556  005037  015602                 CLR     $SAVR6          ;WAIT LOOP FOR THE TTY
3704  015562  005237  015602          1$:    INC     $SAVR6          ;WAIT FOR THE INC
3705  015566  001375                         BNE     1$              ;OF WORD
3706  015570  104401                         TYPE                    ;REPORT THE POWER FAILURE
3707  015572  015604          $PWRMG: .WORD   $POWER                  ;POWER FAIL MESSAGE POINTER
3708  015574  000002                         RTI
3709  015576  000000          $ILLUP: HALT                            ;THE POWER UP SEQUENCE WAS STARTED
3710  015600  000776                         BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
3711  015602  000000          $SAVR6: 0                               ;PUT THE SP HERE
3712  015604  005015  047520  042527 $POWER: .ASCIZ  <15><12>'POWER'
3713  015612  000122
3714
```

```
                                        .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

015614                                  $SCOPE:
015614  104407                                 CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
015616  032777  040000  163214          1$:    BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
015624  001052                                 BNE     $OVER           ;;YES IF SW14=1
                                        ;#####START OF CODE FOR THE XOR TESTER#####
015626  000416                          $XTSTR: BR      6$              ;;IF RUNNING ON THE ''XOR'' TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A ''NOP'' (NOP-240)
015630  013746  000004                         MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
015634  012737  015654  000004               MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
015642  005737  177060                       TST     @#177060        ;;TIME OUT ON XOR?
015646  012637  000004                       MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
015652  000421                             BR      $SVLAD          ;;GO TO THE NEXT TEST
015654  022626                          5$:    CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
015656  012637  000004                       MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
015662  000407                             BR      7$              ;;LOOP ON THE PRESENT TEST
015664                                  6$:;#####END OF CODE FOR THE XOR TESTER#####
015664  105737  001003                  2$:    TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
015670  001412                             BEQ     $SVLAD          ;;BR IF NO
015672  032777  001000  163140               BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
015700  001404                             BEQ     4$              ;;BR IF NO
015702  013737  001010  001006          7$:    MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
015710  000420                             BR      $OVER
015712  105037  001003                  4$:    CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
015716  105237  001002                  $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
015722  113737  001002  001104               MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
015730  011637  001006                       MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
015734  011637  001010                       MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
015740  005037  001072                       CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
015744  112737  000001  001015               MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
015752  013777  001002  163062          $OVER: MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
015760  013716  001006                       MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
015764  000002                             RTI                     ;;FIXES PS
```

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

015766  112737  000001  016232  $ATY1:  MOVB    #1,$FFLG        ;TO REPORT FATAL ERROR
015774  112737  000001  016230  $ATY3:  MOVB    #1,$MFLG        ;TO TYPE A MESSAGE
016002  000403                         BR      $ATYC
016004  112737  000001  016232  $ATY4:  MOVB    #1,$FFLG        ;TO ONLY REPORT FATAL FRROR
016012                         $ATYC:
016012  010046                         MOV     R0,-(SP)        ;PUSH R0 ON STACK
016014  010146                         MOV     R1,-(SP)        ;PUSH R1 ON STACK
016016  105737  016230                 TSTB    $MFLG           ;SHOULD TYPE A MESSAGE?
016022  001450                         BEQ     5$              ;IF NOT: BR
016024  122737  000001  001120         CMPB    #APTENV,$ENV    ;OPERATING UNDER APT?
016032  001031                         BNE     3$              ;IF NOT: BR
016034  132737  000100  001121         BITB    #APTSPOOL,$ENVM ;SHOULD SPOOL MESSAGE?
016042  001425                         BEQ     3$              ;IF NOT: BR
016044  017600  000004                 MOV     @4(SP),R0       ;GET MESSAGE ADDRESS
016050  062766  000002  000004         ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
016056  005737  001100          1$:    TST     $MSGTYPE                ;SEE IF DONE W/ LAST XMISSION?
016062  001375                         BNE     1$              ;IF NOT: WAIT
016064  010037  001114                 MOV     R0,$MSGAD       ;PUT ADDRESS IN MAILBOX
016070  105720                 2$:     TSTB    (R0)+           ;FIND END OF MESSAGE
016072  001376                         BNE     2$
016074  163700  001114                 SUB     $MSGAD,R0       ;SUB START OF MESSAGE
016100  006200                         ASR     R0              ;GET MESSAGE LENGTH IN WORDS
016102  010037  001116                 MOV     R0,$MSGLGT      ;PUT LENGTH IN MAILBOX
016106  012737  000004  001100         MOV     #4,$MSGTYPE     ;TELL APT TO TAKE MESSAGE
016114  000413                         BR      5$
016116  017637  000004  016142  3$:     MOV     @4(SP),4$       ;PUT MSG ADDR IN JSR LINKAGE
016124  062766  000002  000004         ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
016132  013746  177776                 MOV     177776,-(SP)    ;PUSH 177776 ON STACK
016136  004737  016234                 JSR     PC,$TYPE        ;CALL TYPE MACRO
016142  000000                 4$:     .WORD   0
016144                         5$:
016144  105737  016232          10$:   TSTB    $FFLG           ;SHOULD REPORT FATAL ERROR?
016150  001413                         BEQ     12$             ;IF NOT: BR
016152  005737  001120                 TST     $ENV            ;RUNNING UNDER APT?
016156  001410                         BEQ     12$             ;IF NOT: BR
016160  005737  001100          11$:   TST     $MSGTYPE        ;FINISHED LAST MESSAGE?
016164  001375                         BNE     11$             ;IF NOT: WAIT
016166  017637  000004  001102         MOV     @4(SP),$FATAL   ;GET ERROR #
016174  005237  001100                 INC     $MSGTYPE        ;TELL APT TO TAKE ERROR
016200  062766  000002  000004  12$:    ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
016206  105037  016232                 CLRB    $FFLG           ;CLEAR FATAL FLAG
016212  105037  016231                 CLRB    $LFLG           ;CLEAR LOG FLAG
016216  105037  016230                 CLRB    $MFLG           ;CLEAR MESSAGE FLAG
016222  012601                         MOV     (SP)+,R1        ;POP STACK INTO R1
016224  012600                         MOV     (SP)+,R0        ;POP STACK INTO R1
016226  000207                         RTS     PC              ;RETURN
016230     000                 $MFLG:  .BYTE   0
016231     000                 $LFLG:  .BYTE   0               ;LOG FLAG
016232     000                 $FFLG:  .BYTE   0               ;FATAL FLAG

        016234                           .EVEN
```

```
3817            000200          APTSIZE=200
3818            000001          APTENV=001
3819            000100          APTSPOOL=100
3820            000040          APTCSUP=040
3821
```

```
                                .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
                                ;*

016234  105737  001057          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
016240  100002                          BPL     1$              ;;BR IF YES
016242  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
016244  000430                          BR      3$              ;;LEAVE
016246  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
016250  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
016254  122737  000001  001120          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
016262  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
016264  132737  000100  001121          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
016272  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
016274  010037  016304                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
016300  004737  015774                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
016304  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
016306  132737  000040  001121  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
016314  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
016316  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
016320  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
016322  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
016324  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
016326  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
016332  000002                          RTI                     ;;RETURN
016334  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
016340  001430                          BEQ     8$
016342  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
016346  001006                          BNE     5$
016350  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
016352  104401                          TYPE                    ;;TYPE A CR AND LF
016354  001075                          $CRLF
016356  105037  016564                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
016362  000755                          BR      2$              ;;GET NEXT CHARACTER
016364  004737  016446          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
016370  123726  001056          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
016374  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
016376  013746  001054                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
016402  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
016406  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
016410  004737  016446                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
```

```
3878  016414  105337  016564                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
3879  016420  000770                          BR      7$              ;;LOOP
3880
3881                                          ;HORIZONTAL TAB PROCESSOR
3882
3883  016422  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
3884  016426  004737  016446          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
3885  016432  132737  000007  016564          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
3886  016440  001372                          BNE     9$              ;;TAB STOP
3887  016442  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
3888  016444  000724                          BR      2$              ;;GET NEXT CHARACTER
3889  016446                        $TYPEC:
3890  016446  105777  162372                  TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?          ;MJD001
3891  016452  100022                          BPL     10$             ;;BR IF NOT                     ;MJD001
3892  016454  017746  162366                  MOV     @$TKB,-(SP)     ;;GET CHAR                      ;MJD001
3893  016460  042716  177600                  BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS         ;MJD001
3894  016464  122716  000023                  CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF                 ;MJD001
3895  016470  001012                          BNE     102$            ;;BR IF NOT                     ;MJD001
3896  016472                                                                                          ;MJD001
3897  016472  105777  162346          101$:   TSTB    @$TKS           ;;WAIT FOR CHAR                 ;MJD001
3898  016476  100375                          BPL     101$                                            ;MJD001
3899  016500  117716  162342                  MOVB    @$TKB,(SP)      ;;GET CHAR                      ;MJD001
3900  016504  042716  177600                  BIC     #177600,(SP)    ;;STRIP IT                      ;MJD001
3901  016510  122716  000021                  CMPB    #$XON,(SP)      ;;WAS IT XON?                   ;MJD001
3902  016514  001366                          BNE     101$            ;;BR IF NOT                     ;MJD001
3903  016516                                                                                          ;MJD001
3904  016516  005726                  102$:   TST     (SP)+           ;;FIX STACK                     ;MJD001
3905  016520                                                                                          ;MJD001
3906  016520  105777  162324          10$:    TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY   ;MJD001
3907  016524  100375                          BPL     10$                                             ;MJD001
3908  016526  116677  000002  162316          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
3909  016534  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
3910  016542  001003                          BNE     1$              ;;BRANCH IF NO
3911  016544  105037  016564                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
3912  016550  000406                          BR      $TYPEX          ;;EXIT
3913  016552  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
3914  016560  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
3915  016562  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
3916  016564  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
3917  016566  000207          $TYPEX: RTS     PC
3918
3919                                          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3920
3921                                          ;;**************************************************************************
3922                                          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3923                                          ;*OCTAL (ASCII) NUMBER AND TYPE IT.
3924                                          ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3925                                          ;*CALL:
3926                                          ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3927                                          ;*      TYPOS                   ;;CALL FOR TYPEOUT
3928                                          ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3929                                          ;*      .BYTE   M               ;;M=1 OR 0
3930                                          ;*                                      ;;1-TYPE LEADING ZEROS
3931                                          ;*                                      ;;0=SUPPRESS LEADING ZEROS
3932                                          ;*
3933                                          ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
```

```
3934                                    ;*$TYPOS OR $TYPOC
3935                                    ;*CALL:
3936                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3937                                    ;*      TYPON                   ;;CALL FOR TYPEOUT
3938                                    ;*
3939                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3940                                    ;*CALL:
3941                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3942                                    ;*      TYPOC                   ;;CALL FOR TYPEOUT
3943
3944  016570  017646  000000            $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
3945  016574  116637  000001  017013            MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
3946  016602  112637  017015                    MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
3947  016606  062716  000002                    ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
3948  016612  000406                            BR      $TYPON
3949  016614  112737  000001  017013  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
3950  016622  112737  000006  017015            MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
3951  016630  112737  000005  017012  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
3952  016636  010346                            MOV     R3,-(SP)        ;;SAVE R3
3953  016640  010446                            MOV     R4,-(SP)        ;;SAVE R4
3954  016642  010546                            MOV     R5,-(SP)        ;;SAVE R5
3955  016644  113704  017015                    MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
3956  016650  005404                            NEG     R4
3957  016652  062704  000006                    ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
3958  016656  110437  017014                    MOVB    R4,$OMODE       ;;SAVE IT FOR USE
3959  016662  113704  017013                    MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
3960  016666  016605  000012                    MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
3961  016672  005003                            CLR     R3              ;;CLEAR THE OUTPUT WORD
3962  016674  006105                    1$:     ROL     R5              ;;ROTATE MSB INTO "C"
3963  016676  000404                            BR      3$              ;;GO DO MSB
3964  016700  006105                    2$:     ROL     R5              ;;FORM THIS DIGIT
3965  016702  006105                            ROL     R5
3966  016704  006105                            ROL     R5
3967  016706  010503                            MOV     R5,R3
3968  016710  006103                    3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
3969  016712  105337  017014                    DECB    $OMODE          ;;TYPE THIS DIGIT?
3970  016716  100016                            BPL     7$              ;;BR IF NO
3971  016720  042703  177770                    BIC     #177770,R3      ;;GET RID OF JUNK
3972  016724  001002                            BNE     4$              ;;TEST FOR 0
3973  016726  005704                            TST     R4              ;;SUPPRESS THIS 0?
3974  016730  001403                            BEQ     5$              ;;BR IF YES
3975  016732  005204                    4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
3976  016734  052703  000060                    BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
3977  016740  052703  000040            5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
3978  016744  110337  017010                    MOVB    R3,8$           ;;SAVE FOR TYPING
3979  016750  104401  017010                    TYPE    ,8$             ;;GO TYPE THIS DIGIT
3980  016754  105337  017012            7$:     DECB    $OCNT           ;;COUNT BY 1
3981  016760  003347                            BGT     2$              ;;BR IF MORE TO DO
3982  016762  002402                            BLT     6$              ;;BR IF DONE
3983  016764  005204                            INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
3984  016766  000744                            BR      2$              ;;GO DO THE LAST DIGIT
3985  016770  012605                    6$:     MOV     (SP)+,R5        ;;RESTORE R5
3986  016772  012604                            MOV     (SP)+,R4        ;;RESTORE R4
3987  016774  012603                            MOV     (SP)+,R3        ;;RESTORE R3
3988  016776  016666  000002  000004            MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
3989  017004  012616                            MOV     (SP)+,(SP)
```

```
                        RTI                             ;;RETURN
8$:                     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
                        .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
$OCNT:                  .BYTE   0                       ;;OCTAL DIGIT COUNTER
$OFILL:                 .BYTE   0                       ;;ZERO FILL SWITCH
$OMODE:                 .WORD   0                       ;;NUMBER OF DIGITS TO TYPE
                        .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

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

$TYPDS:
                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
                        MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
                        MOV     20(SP),R5       ;;GET THE INPUT NUMBER
                        BPL     1$              ;;BR IF INPUT IS POS.
                        NEG     R5              ;;MAKE THE BINARY NUMBER POS.
                        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
1$:                     CLR     R0              ;;ZERO THE CONSTANTS INDEX
                        MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
                        MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
2$:                     CLR     R2              ;;CLEAR THE BCD NUMBER
                        MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
3$:                     SUB     R1,R5           ;;FORM THIS BCD DIGIT
                        BLT     4$              ;;BR IF DONE
                        INC     R2              ;;INCREASE THE BCD DIGIT BY 1
                        BR      3$
4$:                     ADD     R1,R5           ;;ADD BACK THE CONSTANT
                        TST     R2              ;;CHECK IF BCD DIGIT=0
                        BNE     5$              ;;FALL THROUGH IF 0
                        TSTB    (SP)            ;;STILL DOING LEADING 0'S?
                        BMI     7$              ;;BR IF YES
5$:                     ASLB    (SP)            ;;MSD?
                        BCC     6$              ;;BR IF NO
                        MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
6$:                     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
7$:                     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
                        MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
                        TST     (R0)+           ;;JUST INCREMENTING
                        CMP     R0,#10          ;;CHECK THE TABLE INDEX
                        BLT     2$              ;;GO DO THE NEXT DIGIT
                        BGT     8$              ;;GO TO EXIT
                        MOV     R5,R2           ;;GET THE LSD
                        BR      6$              ;;GO CHANGE TO ASCII
8$:                     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
```

```
 4046  017160  100003                    BPL     9$                  ;;BR IF NO
 4047  017162  116663  177777  177776    MOVB    -1(SP),-2(R3)       ;;YES--SET THE SIGN FOR TYPING
 4048  017170  105013                9$: CLRB    (R3)                ;;SET THE TERMINATOR
 4049  017172  012605                    MOV     (SP)+,R5            ;;POP STACK INTO R5
 4050  017174  012603                    MOV     (SP)+,R3            ;;POP STACK INTO R3
 4051  017176  012602                    MOV     (SP)+,R2            ;;POP STACK INTO R2
 4052  017200  012601                    MOV     (SP)+,R1            ;;POP STACK INTO R1
 4053  017202  012600                    MOV     (SP)+,R0            ;;POP STACK INTO R0
 4054  017204  104401  017232            TYPE    ,$DBLK              ;;NOW TYPE THE NUMBER
 4055  017210  016666  000002  000004    MOV     2(SP),4(SP)         ;;ADJUST THE STACK
 4056  017216  012616                    MOV     (SP)+,(SP)
 4057  017220  000002                    RTI                         ;;RETURN TO USER
 4058  017222  023420                $DTBL:  10000.
 4059  017224  001750                        1000.
 4060  017226  000144                        100.
 4061  017230  000012                        10.
 4062  017232  000004                $DBLK:  .BLKW   4
 4063
```

```
                                        .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.
017242  022737  000176  001040  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
017250  001074                        BNE     15$             ;;BRANCH IF NO
017252  105777  161566                TSTB    @$TKS           ;;CHAR THERE?
017256  100071                        BPL     15$             ;;IF NO, DON'T WAIT AROUND
017260  117746  161562                MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
017264  042716  177600                BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
017270  022726  000007                CMP     #7,(SP)+        ;;IS IT A CONTROL G?
017274  001062                        BNE     15$             ;;NO, RETURN TO USER
017276  123727  001034  000001        CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
017304  001456                        BEQ     15$             ;;BRANCH IF YES

017306  104401  017777                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
017312  104401  020004        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
017316  013746  000176                MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
017322  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
017324  104401  020015                TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
017330  005046                19$:    CLR     -(SP)           ;;CLEAR COUNTER
017332  005046                        CLR     -(SP)           ;;THE NEW SWR
017334  105777  161504        7$:     TSTB    @$TKS           ;;CHAR THERE?
017340  100375                        BPL     7$              ;;IF NOT TRY AGAIN

017342  117746  161500                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
017346  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII


017352  021627  000025        9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
017356  001005                        BNE     10$             ;;BRANCH IF NOT
017360  104401  017772                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
017364  062706  000006        20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
017370  000757                        BR      19$             ;;LET'S TRY IT AGAIN


017372  021627  000015        10$:    CMP     (SP),#15        ;;IS IT A <CR>?
017376  001022                        BNE     16$             ;;BRANCH IF NO
017400  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
017404  001403                        BEQ     11$             ;;BRANCH IF YES
017406  016677  000002  161424        MOV     2(SP),@SWR      ;;SAVE NEW SWR
017414  062706  000006        11$:    ADD     #6,SP           ;;CLEAR UP STACK
017420  104401  001075        14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
017424  123727  001035  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
017432  001003                        BNE     15$             ;;BRANCH IF NOT
017434  012777  000100  161402        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
017442  000002                15$:    RTI                     ;;RETURN
017444  004737  016446        16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
```

```
4120  017450  021627  000060            CMP     (SP),#60        ;;CHAR < 0?
4121  017454  002420                    BLT     18$             ;;BRANCH IF YES
4122  017456  021627  000067            CMP     (SP),#67        ;;CHAR > 7?
4123  017462  003015                    BGT     18$             ;;BRANCH IF YES
4124  017464  042726  000060            BIC     #60,(SP)+       ;;STRIP-OFF ASCII
4125  017470  005766  000002            TST     2(SP)           ;;IS THIS THE FIRST CHAR
4126  017474  001403                    BEQ     17$             ;;BRANCH IF YES
4127  017476  006316                    ASL     (SP)            ;;NO, SHIFT PRESENT
4128  017500  006316                    ASL     (SP)            ;;   CHAR OVER TO MAKE
4129  017502  006316                    ASL     (SP)            ;;   ROOM FOR NEW ONE.
4130  017504  005266  000002    17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
4131  017510  056616  177776            BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
4132  017514  000707                    BR      7$              ;;GET THE NEXT ONE
4133  017516  104401  001074    18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
4134  017522  000720                    BR      20$             ;;SIMULATE CONTROL-U
4135                            .DSABL  LSB
4136
4137
4138                            ;;*****************************************************************
4139                            ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
4140                            ;*CALL:
4141                            ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
4142                            ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
4143                            ;*                              ;;WITH PARITY BIT STRIPPED OFF
4144                            ;
4145
4146  017524  011646    $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
4147  017526  016666  000004  000002    MOV     4(SP),2(SP)     ;;SAVE THE PS
4148  017534  105777  161304    1$:     TSTB    @$TKS           ;;WAIT FOR
4149  017540  100375                    BPL     1$              ;;A CHARACTER
4150  017542  117766  161300  000004    MOVB    @$TKB,4(SP)     ;;READ THE TTY
4151  017550  042766  177600  000004    BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
4152  017556  026627  000004  000023    CMP     4(SP),#23       ;;IS IT A CONTROL-S?
4153  017564  001013                    BNE     3$              ;;BRANCH IF NO
4154  017566  105777  161252    2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
4155  017572  100375                    BPL     2$              ;;LOOP UNTIL ITS THERE
4156  017574  117746  161246            MOVB    @$TKB,-(SP)     ;;GET CHARACTER
4157  017600  042716  177600            BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
4158  017604  022627  000021            CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
4159  017610  001366                    BNE     2$              ;;IF NOT DISCARD IT
4160  017612  000750                    BR      1$              ;;YES, RESUME
4161  017614  026627  000004  000021  3$:     CMP     4(SP),#$XON     ;;IS IT A RANDOM XON?           ;RAN001
4162  017622  001744                    BEQ     1$              ;;BRANCH IF YES                 ;RAN001
4163  017624  026627  000004  000140        CMP     4(SP),#140      ;;IS IT UPPER CASE?
4164  017632  002407                    BLT     4$              ;;BRANCH IF YES
4165  017634  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
4166  017642  003003                    BGT     4$              ;;BRANCH IF YES
4167  017644  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
4168  017652  000002              4$:     RTI                     ;;GO BACK TO USER
4169                            ;;*****************************************************************
4170                            ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
4171                            ;*CALL:
4172                            ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
4173                            ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
4174                            ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
4175
```

```
 4176  017654  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
 4177  017656  012703  017762          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
 4178  017662  022703  017772          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
 4179  017666  101405                          BLOS    4$              ;;BR IF YES
 4180  017670  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 4181  017672  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 4182  017674  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
 4183  017700  001003                          BNE     3$              ;;SKIP IF NOT
 4184  017702  104401  001074          4$:     TYPE    ,$QUES          ;;TYPE A '?'
 4185  017706  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 4186  017710  111337  017760          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 4187  017714  104401  017760                  TYPE    ,9$
 4188  017720  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 4189  017724  001356                          BNE     2$              ;;LOOP IF NOT RETURN
 4190  017726  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 4191  017732  104401  001076                  TYPE    ,$LF            ;;TYPE A LINE FEED
 4192  017736  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 4193  017740  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 4194  017742  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 4195  017750  012766  017762  000004          MOV     #$TTYIN,4(SP)
 4196  017756  000002                          RTI                     ;;RETURN
 4197  017760     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 4198  017761     000                          .BYTE   0               ;;TERMINATOR
 4199  017762  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
 4200  017772  052536  005015     000 $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
 4201  017777     136  006507  000012 $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
 4202  020004  005015  053523  020122 $MSWR:  .ASCIZ  <15><12>/SWR = /
 4203  020012  020075     000
 4204  020015     040  047040  053505 $MNEW:  .ASCIZ  /  NEW = /
 4205  020022  036440  000040
 4206
```

```
                                        .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.

020026  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
020030  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
020034  005740                          TST     -(R0)           ;;BACKUP BY 2
020036  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
020040  006300                          ASL     R0              ;;POSITION FOR INDEXING
020042  016000  020062                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
020046  000200                          RTS     R0              ;;GO TO ROUTINE


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

020050  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
020052  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
020060  000002                          RTI                     ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

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

                                        ;       ROUTINE
                                        ;       -------
020062  020050                  $TRPAD: .WORD   $TRAP2
020064  016234                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
020066  016614                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
020070  016570                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
020072  016630                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
020074  017016                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

020076  017312                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

020100  017242                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
020102  017524                          $RDCHR  ;;CALL-RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
020104  017654                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
```

```
                                        .SBTTL  ECHO TEST
                                ;;**************************************************************
                                ;*THIS ROUTINE WILL ECHO ANY CHARACTER TYPED
                                ;*ON EITHER SLU1 OR SLU2. DEFAULT IS TO THE CONSOLE DEVICE SLU1.
                                ;*THE TEST IS HALTED BY TYPING A CONTROL-C
                                ;*TEST CAN BE RESTAR' D AFTER HALTING BY JUST CONTINUING
                                ;;**************************************************************
020106  012737  000176  001040  ECHO:   MOV     #SWREG,SWR              ;SET UP FOR SOFTWARE SWITCH REGISTER
020114  012737  000174  001042          MOV     #DISPREG,DISPLAY        ;AND DISPLAY REGISTER
020122  032777  000020  160710          BIT     #BIT4, @SWR             ;CHECK IF BIT4 SET IN SWITCH REG
020130  001403                        BEQ     1$                      ;IF NOT THEN SELECT SLU1
020132  012703  002770                MOV     #RCSR, R3               ;IF BIT4 SET THEN SELECT SLU2
020136  000402                        BR      2$
020140  012703  003000          1$:   MOV     #CRCSR, R3              ;SELECT SLU1 (THIS IS DEFAULT DEVICE)
020144  000005                  2$:   RESET                   ;CLEAR EVERYTHING
020146  112773  000052  000006        MOVB    #'*,@6(R3)      ;TRANSMIT PROMPT ''*''
020154  105773  000000          3$:   TSTB    @(R3)           ;WAIT FOR INPUT
020160  100375                        BPL     3$
020162  117373  000002  000006        MOVB    @2(R3),@6(R3)   ;ECHO INPUT
020170  017300  000002                MOV     @2(R3),R0       ;STORE INPUT
020174  100023                        BPL     6$              ;BR IF 'ERROR'NOT SET
020176  052701  010000                BIS     #BIT12,R1       ;SET PARITY ERROR TEST MASK
020202  030100                        BIT     R1,R0           ;CHECK  FOR PARITY ERROR FLAG
020204  001403                        BEQ     4$              ;BR IF NOT SET
020206  004737  020270                JSR     PC,MSG          ;REPORT PARITY ERROR
020212  020316                        MPAR
020214  006301                  4$:   ASL     R1              ;SHIFT MASK TO TEST ''FR'' FLAG
020216  030100                        BIT     R1,R0           ;TEST FOR FRAMING ERROR FLAG
020220  001403                        BEQ     5$              ;BR IF NOT SET
020222  004737  020270                JSR     PC,MSG          ;REPSORT FRAMING ERROR
020226  020327                        MFR
020230  006301                  5$:   ASL     R1              ;SHIFT MASK TO TEST ''OR'' FLAG
020232  030100                        BIT     R1,R0           ;TEST FOR OVERFLOW ERROR
020234  001403                        BEQ     6$              ;BR IF NOT SET
020236  004737  020270                JSR     PC,MSG          ;REPORT OVERFLOW ERROR
020242  020341                        MOR
020244  042700  000200          6$:   BIC     #BIT7,R0        ;CLEAR ANY PARITY BIT
020250  022700  000003                CMP     #3,R0           ;WAS INPUT CONTROL-C
020254  001337                        BNE     3$              ;BR IF IS NOT
020256  004737  020270                JSR     PC,MSG          ;REPORT PROGRAM STOP
020262  020354                        MSTOP
020264  000000                        HALT                    ;END OF TEST HALT
020266  000707                        BR      ECHO            ;AFTER END OF TEST HALT
                                                            ;   PRESS CONTINUE TO RESTART ECHO TEST

020270  013600                  MSG:  MOV     @(SP)+,R0       ;PICK UP MESSAGE POINTER
020272  062746  000002                ADD     #2,-(SP)        ;ADJUST RETURN PC
020276  105773  000004          WAIT: TSTB    @4(R3)          ;WAIT FOR XMIT DONE
020302  100375                        BPL     WAIT
020304  112073  000006                MOVB    (R0)+,@6(R3)    ;SEND CHARACTER
020310  105710                        TSTB    (R0)            ;IS THIS END OF MESSAGE?
020312  001371                        BNE     WAIT            ;BR IF NOT
020314  000207                        RTS     PC              ;RETURN

020316  005015  040520  044522  MPAR: .ASCIZ  <CR><LF>/PARITY/
020324  054524     000
```

```
 4308  020327     015  043012  040522  MFR:    .ASCIZ  <CR><LF>/FRAMING/
 4309  020334  044515  043516     000
 4310  020341     015  047412  042526  MOR:    .ASCIZ  <CR><LF>/OVERFLOW/
 4311  020346  043122  047514  000127
 4312  020354  005015  052123  050117  MSTOP:  .ASCIZ  <CR><LF>/STOP/
 4313  020362     000
```

```


        020364                                .EVEN

                                     .SBTTL  TERMINAL OUTPUT TEST

                                     ;;*********************************************************************
                                              ;*THIS ROUTINE WILL OUTPUT ALL WRITABLE CHARACTERS FOR THE
                                              ;*THE OCTAL CODE 040 --> 377
                                              ;*32 CHARACTERS ARE PRINTED ON EACH LINE
                                              ;*THE PATTERN IS REPEATED EVERY THREE LINES
                                              ;*
                                     ;;*********************************************************************

020364  012737  000176  001040       OUTTST: MOV     #SWREG,SWR               ;SET UP FOR SOFTWARE SWITCH REGISTER
020372  012737  000174  001042               MOV     #DISPREG,DISPLAY         ;AND DISPLAY REGISTER
020400  032777  000020  160432               BIT     #BIT4, @SWR              ;CHECK IF BIT4 SET IN SWITCH REG
020406  001403                               BEQ     1$                       ;IF NOT THEN SELECT SLU1
020410  012703  002770                       MOV     #RCSR, R3                ;IF BIT4 SET THEN SELECT SLU2
020414  000402                               BR      2$
020416  012703  003000               1$:     MOV     #CRCSR, R3               ;SELECT SLU1 (THIS IS DEFAULT DEVICE)
020422  000005                       2$:     RESET                    ;CLEAR THE WORLD
020424  012701  000040               3$:     MOV     #40,R1           ;LOAD FIRST WRITABLE CHARACTER
020430  012700  000040               4$:     MOV     #40,R0           ;LOAD CHAR COUNT PER LINE
020434  105773  000004               5$:     TSTB    @4(R3)           ;WAIT FOR DONE
020440  100375                               BPL     5$
020442  010173  000006                       MOV     R1,@6(R3)        ;TRANSMIT A CHARACTER
020446  105201                               INCB    R1               ;INCREMENT CHARACTER CODE
020450  005300                               DEC     R0               ;DECREMENT CHAR COUNT
020452  001370                               BNE     5$               ;BR IF LINE NOT COMPLETE
020454  004737  020270                       JSR     PC,MSG           ;SSUE CR,LINE FEED

020460  001075                               $CRLF
020462  105773  000000                       TSTB    @(R3)            ;ANY CHARACTER RECEIVED?
020466  100404                               BMI     6$               ;BR IF YES
020470  032701  000200                       BIT     #BIT7,R1         ;FINISHED ONE PASS OF WRITABLE CHARACTERS?
020474  001353                               BNE     3$               ;BR IF YES
020476  000754                               BR      4$               ;IF NOT WRITE NEXT LINE

020500  005073  000002               6$:     CLR     @2(R3)           ;CLEAR RECEIVER
020504  000000                               HALT                     ;STOP TEST
020506  000726                               BR      OUTTST           ;RESTART TEST IF CONTINUED
```

```
4356
4357
4358  020510  046123  030525  052040  EM5:    .ASCIZ  /SLU1 TCSR DONE NOT SET WITH RESET/
4359  020516  051503  020122  047504
4360  020524  042516  047040  052117
4361  020532  051440  052105  053440
4362  020540  052111  020110  042522
4363  020546  042523  000124
4364  020552  046123  030525  051040  EM6:    .ASCIZ  /SLU1 RCSR DID NOT RETURN SSYNC/
4365  020560  051503  020122  044504
4366  020566  020104  047516  020124
4367  020574  042522  052524  047122
4368  020602  051440  054523  041516
4369  020610     000
4370  020611     123  052514  020061  EM7:    .ASCIZ  /SLU1 RBUF DID NOT RETURN SSYNC/
4371  020616  041122  043125  042040
4372  020624  042111  047040  052117
4373  020632  051040  052105  051125
4374  020640  020116  051523  047131
4375  020646  000103
4376  020650  040503  020116  047516  EM11:   .ASCIZ  /CAN NOT SET BIT2 OF SLU1 TCSR/
4377  020656  020124  042523  020124
4378  020664  044502  031124  047440
4379  020672  020106  046123  030525
4380  020700  052040  051503  000122
4381  020706  042522  042523  020124  EM13:   .ASCIZ  /RESET DID NOT CLEAR BIT2 OF SLU1 TCSR/
4382  020714  044504  020104  047516
4383  020722  020124  046103  040505
4384  020730  020122  044502  031124
4385  020736  047440  020106  046123
4386  020744  030525  052040  051503
4387  020752  000122
4388  020754  044502  033124  047440  EM14:   .ASCIZ  /BIT6 OF SLU1 TCSR NOT CLEAR AFTER RESET/
4389  020762  020106  046123  030525
4390  020770  052040  051503  020122
4391  020776  047516  020124  046103
4392  021004  040505  020122  043101
4393  021012  042524  020122  042522
4394  021020  042523  000124
4395  021024  046123  030525  054040  EM15:   .ASCIZ  /SLU1 XMIT INTERRUPTED WITH PRIORITY 7/
4396  021032  044515  020124  047111
4397  021040  042524  051122  050125
4398  021046  042524  020104  044527
4399  021054  044124  050040  044522
4400  021062  051117  052111  020131
4401  021070  000067
4402  021072  040503  020116  047516  EM16:   .ASCIZ  /CAN NOT SET BIT6 OF SLU1 TCSR/
4403  021100  020124  042523  020124
4404  021106  044502  033124  047440
4405  021114  020106  046123  030525
4406  021122  052040  051503  000122
4407  021130  040503  020116  047516  EM17:   .ASCIZ  /CAN NOT CLEAR BIT6 OF SLU1 TCSR/
4408  021136  020124  046103  040505
4409  021144  020122  044502  033124
4410  021152  047440  020106  046123
4411  021160  030525  052040  051503
```

```
4412  021166  000122
4413  021170  042522  042523  020124  EM20:   .ASCIZ  /RESET DID NOT CLEAR BIT6 OF SLU1 TCSR/
4414  021176  044504  020104  047516
4415  021204  020124  046103  040505
4416  021212  020122  044502  033124
4417  021220  047440  020106  046123
4418  021226  030525  052040  051503
4419  021234  000122
4420  021236  044502  033124  047440  EM21:   .ASCIZ  /BIT6 OF SLU1 RCSR NOT CLEAR AFTER RESET/
4421  021244  020106  046123  030525
4422  021252  051040  051503  020122
4423  021260  047516  020124  046103
4424  021266  040505  020122  043101
4425  021274  042524  020122  042522
4426  021302  042523  000124
4427  021306  046123  030525  051040  EM22:   .ASCIZ  /SLU1 RCVR INTERRUPT WITH PRIORITY 7/
4428  021314  053103  020122  047111
4429  021322  042524  051122  050125
4430  021330  020124  044527  044124
4431  021336  050040  044522  051117
4432  021344  052111  020131  000067
4433  021352  040503  020116  047516  EM23:   .ASCIZ  /CAN NOT SET BIT6 OF SLU1 RCSR/
4434  021360  020124  042523  020124
4435  021366  044502  033124  047440
4436  021374  020106  046123  030525
4437  021402  051040  051503  000122
4438  021410  040503  020116  047516  EM24:   .ASCIZ  /CAN NOT CLEAR BIT6 OF SLU1 RCSR/
4439  021416  020124  046103  040505
4440  021424  020122  044502  033124
4441  021432  047440  020106  046123
4442  021440  030525  051040  051503
4443  021446  000122
4444  021450  040503  020116  047516  EM25:   .ASCIZ  /CAN NOT CLEAR BIT6 OF SLU1 RCSR WITH RESET/
4445  021456  020124  046103  040505
4446  021464  020122  044502  033124
4447  021472  047440  020106  046123
4448  021500  030525  051040  051503
4449  021506  020122  044527  044124
4450  021514  051040  051505  052105
4451  021522     000
4452  021523     123  052514  020061  EM26:   .ASCIZ  /SLU1 RECEIVER DONE NEVER SET/
4453  021530  042522  042503  053111
4454  021536  051105  042040  047117
4455  021544  020105  042516  042526
4456  021552  020122  042523  000124
4457  021560  042522  042523  020124  EM27:   .ASCIZ  /RESET DID NOT CLEAR SLU1 RCVR DONE/
4458  021566  044504  020104  047516
4459  021574  020124  046103  040505
4460  021602  020122  046123  030525
4461  021610  051040  053103  020122
4462  021616  047504  042516     000
4463  021623     122  040505  044504  EM30:   .ASCIZ  /READING SLU1 RBUF DID NOT CLEAR RCVR DONE/
4464  021630  043516  051440  052514
4465  021636  020061  041122  043125
4466  021644  042040  042111  047040
4467  021652  052117  041440  042514
```

```
                                                                    B 8
 4468  021660  051101  051040  053103
 4469  021666  020122  047504  042516
 4470  021674     000
 4471  021675     123  052514  020062  EM31:   .ASCIZ  /SLU2 TCSR DID NOT RETURN SSYNC/
 4472  021702  041524  051123  042040
 4473  021710  042111  047040  052117
 4474  021716  051040  052105  051125
 4475  021724  020116  051523  047131
 4476  021732  000103
 4477  021734  046123  031125  052040  EM32:   .ASCIZ  /SLU2 TBUF DID NOT RETURN SSYNC/
 4478  021742  052502  020106  044504
 4479  021750  020104  047516  020124
 4480  021756  042522  052524  047122
 4481  021764  051440  054523  041516
 4482  021772     000
 4483  021773     123  052514  020062  EM33:   .ASCIZ  /SLU2 TCSR DONE NOT CLEARED WITH TBUF FULL/
 4484  022000  041524  051123  042040
 4485  022006  047117  020105  047516
 4486  022014  020124  046103  040505
 4487  022022  042522  020104  044527
 4488  022030  044124  052040  052502
 4489  022036  020106  052506  046114
 4490  022044     000
 4491  022045     123  052514  020062  EM34:   .ASCIZ  /SLU2 TCSR DONE NOT SET AFTER TRANSMIT/
 4492  022052  041524  051123  042040
 4493  022060  047117  020105  047516
 4494  022066  020124  042523  020124
 4495  022074  043101  042524  020122
 4496  022102  051124  047101  046523
 4497  022110  052111     000
 4498  022113     123  052514  020062  EM35:   .ASCIZ  /SLU2 TCSR DONE NOT SET WITH RESET/
 4499  022120  041524  051123  042040
 4500  022126  047117  020105  047516
 4501  022134  020124  042523  020124
 4502  022142  044527  044124  051040
 4503  022150  051505  052105     000
 4504  022155     123  052514  020062  EM36:   .ASCIZ  /SLU2 RCSR DID NOT RETURN SSYNC/
 4505  022162  041522  051123  042040
 4506  022170  042111  047040  052117
 4507  022176  051040  052105  051125
 4508  022204  020116  051523  047131
 4509  022212  000103
 4510  022214  046123  031125  051040  EM37:   .ASCIZ  /SLU2 RBUF DID NOT RETURN SSYNC/
 4511  022222  052502  020106  044504
 4512  022230  020104  047516  020124
 4513  022236  042522  052524  047122
 4514  022244  051440  054523  041516
 4515  022252     000
 4516  022253     102  052111  020060  EM40:   .ASCIZ  /BIT0 OF SLU2 TCSR NOT CLEAR AFTER RESET/
 4517  022260  043117  051440  052514
 4518  022266  020062  041524  051123
 4519  022274  047040  052117  041440
 4520  022302  042514  051101  040440
 4521  022310  052106  051105  051040
 4522  022316  051505  052105     000
 4523  022323     103  047101  047040  EM41:   .ASCIZ  /CAN NOT SET BIT0 OF SLU2 TCSR/
```

```
 4524  022330  052117  051440  052105
 4525  022336  041040  052111  020060
 4526  022344  043117  051440  052514
 4527  022352  020062  041524  051123
 4528  022360     000
 4529  022361     103  047101  047040  EM42:   .ASCIZ  /CAN NOT CLEAR BIT0 OF SLU2 TCSR/
 4530  022366  052117  041440  042514
 4531  022374  051101  041040  052111
 4532  022402  020060  043117  051440
 4533  022410  052514  020062  041524
 4534  022416  051123     000
 4535  022421     122  051505  052105  EM43:   .ASCIZ  /RESET DID NOT CLEAR BIT0 OF SLU2 TCSR/
 4536  022426  042040  042111  047040
 4537  022434  052117  041440  042514
 4538  022442  051101  041040  052111
 4539  022450  020060  043117  051440
 4540  022456  052514  020062  041524
 4541  022464  051123     000
 4542  022467     102  052111  020066  EM44:   .ASCIZ  /BIT6 OF SLU2 TCSR NOT CLEAR AFTER RESET/
 4543  022474  043117  051440  052514
 4544  022502  020062  041524  051123
 4545  022510  047040  052117  041440
 4546  022516  042514  051101  040440
 4547  022524  052106  051105  051040
 4548  022532  051505  052105     000
 4549  022537     123  052514  020062  EM45:   .ASCIZ  /SLU2 XMIT INTERRUPTED WITH PRIORITY 7/
 4550  022544  046530  052111  044440
 4551  022552  052116  051105  052522
 4552  022560  052120  042105  053440
 4553  022566  052111  020110  051120
 4554  022574  047511  044522  054524
 4555  022602  033440     000
 4556  022605     103  047101  047040  EM46:   .ASCIZ  /CAN NOT SET BIT6 OF SLU2 TCSR/
 4557  022612  052117  051440  052105
 4558  022620  041040  052111  020066
 4559  022626  043117  051440  052514
 4560  022634  020062  041524  051123
 4561  022642     000
 4562  022643     103  047101  047040  EM47:   .ASCIZ  /CAN NOT CLEAR BIT6 OF SLU2 TCSR/
 4563  022650  052117  041440  042514
 4564  022656  051101  041040  052111
 4565  022664  020066  043117  051440
 4566  022672  052514  020062  041524
 4567  022700  051123     000
 4568  022703     122  051505  052105  EM50:   .ASCIZ  /RESET DID NOT CLEAR BIT6 OF SLU2 TCSR/
 4569  022710  042040  042111  047040
 4570  022716  052117  041440  042514
 4571  022724  051101  041040  052111
 4572  022732  020066  043117  051440
 4573  022740  052514  020062  041524
 4574  022746  051123     000
 4575  022751     102  052111  020066  EM51:   .ASCIZ  /BIT6 OF SLU2 RCSR NOT CLEAR AFTER RESET/
 4576  022756  043117  051440  052514
 4577  022764  020062  041522  051123
 4578  022772  047040  052117  041440
 4579  023000  042514  051101  040440
```

```
4580  023006  052106  051105  051040
4581  023014  051505  052105     000
4582  023021     123  052514  020062  EM52:   .ASCIZ  /SLU2 RCVR INTERRUPT WITH PRIORITY 7/
4583  023026  041522  051126  044440
4584  023034  052116  051105  052522
4585  023042  052120  053440  052111
4586  023050  020110  051120  047511
4587  023056  044522  054524  033440
4588  023064     000
4589  023065     103  047101  047040  EM53:   .ASCIZ  /CAN NOT SET BIT6 OF SLU2 RCSR/
4590  023072  052117  051440  052105
4591  023100  041040  052111  020066
4592  023106  043117  051440  052514
4593  023114  020062  041522  051123
4594  023122     000
4595  023123     103  047101  047040  EM54:   .ASCIZ  /CAN NOT CLEAR BIT6 OF SLU2 RCSR/
4596  023130  052117  041440  042514
4597  023136  051101  041040  052111
4598  023144  020066  043117  051440
4599  023152  052514  020062  041522
4600  023160  051123     000
4601  023163     103  047101  047040  EM55:   .ASCIZ  /CAN NOT CLEAR BIT6 OF SLU2 RCSR WITH RESET/
4602  023170  052117  041440  042514
4603  023176  051101  041040  052111
4604  023204  020066  043117  051440
4605  023212  052514  020062  041522
4606  023220  051123  053440  052111
4607  023226  020110  042522  042523
4608  023234  000124
4609  023236  046123  031125  051040  EM56:   .ASCIZ  /SLU2 RECEIVER DONE NEVER SET/
4610  023244  041505  044505  042526
4611  023252  020122  047504  042516
4612  023260  047040  053105  051105
4613  023266  051440  052105     000
4614  023273     122  051505  052105  EM57:   .ASCIZ  /RESET DID NOT CLEAR SLU2 RCVR DONE/
4615  023300  042040  042111  047040
4616  023306  052117  041440  042514
4617  023314  051101  051440  052514
4618  023322  020062  041522  051126
4619  023330  042040  047117  000105
4620  023336  042522  042101  047111  EM60:   .ASCIZ  /READING SLU2 RBUF DID NOT CLEAR RCVR DONE/
4621  023344  020107  046123  031125
4622  023352  051040  052502  020106
4623  023360  044504  020104  047516
4624  023366  020124  046103  040505
4625  023374  020122  041522  051126
4626  023402  042040  047117  000105
4627  023410  045514  020123  044504  EM61:   .ASCIZ  /LKS DID NOT RETURN SSYNC/
4628  023416  020104  047516  020124
4629  023424  042522  052524  047122
4630  023432  051440  054523  041516
4631  023440     000
4632  023441     102  052111  020066  EM62:   .ASCIZ  /BIT6 OF LKS NOT CLEAR AFTER RESET/
4633  023446  043117  046040  051513
4634  023454  047040  052117  041440
4635  023462  042514  051101  040440
```

```
4636  023470  052106  051105  051040
4637  023476  051505  052105     000
4638  023503     114  051513  044440  EM63:   .ASCIZ  /LKS INTERRUPT WITH PRIORITY 7/
4639  023510  052116  051105  052522
4640  023516  052120  053440  052111
4641  023524  020110  051120  047511
4642  023532  044522  054524  033440
4643  023540     000
4644  023541     103  047101  047040  EM64:   .ASCIZ  /CAN NOT SET BIT6 OF LKS/
4645  023546  052117  051440  052105
4646  023554  041040  052111  020066
4647  023562  043117  046040  051513
4648  023570     000
4649  023571     103  047101  047040  EM65:   .ASCIZ  /CAN NOT CLEAR BIT6 OF LKS/
4650  023576  052117  041440  042514
4651  023604  051101  041040  052111
4652  023612  020066  043117  046040
4653  023620  051513     000
4654  023623     122  051505  052105  EM66:   .ASCIZ  /RESET DID NOT CLEAR BIT6 OF LKS/
4655  023630  042040  042111  047040
4656  023636  052117  041440  042514
4657  023644  051101  041040  052111
4658  023652  020066  043117  046040
4659  023660  051513     000
4660  023663     102  052111  020067  EM67:   .ASCIZ  /BIT7 OF LKS NOT SET AFTER RESET/
4661  023670  043117  046040  051513
4662  023676  047040  052117  051440
4663  023704  052105  040440  052106
4664  023712  051105  051040  051505
4665  023720  052105     000
4666  023723     103  047101  047040  EM70:   .ASCIZ  /CAN NOT CLEAR BIT7 OF LKS/
4667  023730  052117  041440  042514
4668  023736  051101  041040  052111
4669  023744  020067  043117  046040
4670  023752  051513     000
4671  023755     102  052111  020067  EM71:   .ASCIZ  /BIT7 OF LKS DOES NOT SET/
4672  023762  043117  046040  051513
4673  023770  042040  042517  020123
4674  023776  047516  020124  042523
4675  024004  000124
4676  024006  051127  052111  047111  EM72:   .ASCIZ  /WRITING TO ONE INTERNAL ADDRESS MODIFIED ANOTHER/
4677  024014  020107  047524  047440
4678  024022  042516  044440  052116
4679  024030  051105  040516  020114
4680  024036  042101  051104  051505
4681  024044  020123  047515  044504
4682  024052  044506  042105  040440
4683  024060  047516  044124  051105
4684  024066     000
4685  024067     123  052514  020061  EM74:   .ASCIZ  /SLU1 XMIT INTERRUPTS WHEN DISABLED/
4686  024074  046530  052111  044440
4687  024102  052116  051105  052522
4688  024110  052120  020123  044127
4689  024116  047105  042040  051511
4690  024124  041101  042514  000104
4691  024132  046123  030525  054040  EM75:   .ASCIZ  /SLU1 XMIT DID NOT INTERRUPT/
```

```
4692  024140  044515  020124  044504
4693  024146  020104  047516  020124
4694  024154  047111  042524  051122
4695  024162  050125  000124
4696  024166  046123  030525  054040  EM76:   .ASCIZ  /SLU1 XMIT INTERRUPT AT PRIORITY 7/
4697  024174  044515  020124  047111
4698  024202  042524  051122  050125
4699  024210  020124  052101  050040
4700  024216  044522  051117  052111
4701  024224  020131  000067
4702  024230  046123  030525  054040  EM77:   .ASCIZ  /SLU1 XMIT INTERRUPTS WITH ENABLE CLEAR/
4703  024236  044515  020124  047111
4704  024244  042524  051122  050125
4705  024252  051524  053440  052111
4706  024260  020110  047105  041101
4707  024266  042514  041440  042514
4708  024274  051101     000
4709  024277     123  052514  020061  EM100:  .ASCIZ  /SLU1 XMIT DID NOT INTERRUPT/
4710  024304  046530  052111  042040
4711  024312  042111  047040  052117
4712  024320  044440  052116  051105
4713  024326  052522  052120     000
4714  024333     123  052514  020061  EM101:  .ASCIZ  /SLU1 XMIT RE-INTERRUPTED/
4715  024340  046530  052111  051040
4716  024346  026505  047111  042524
4717  024354  051122  050125  042524
4718  024362  000104
4719  024364  047514  042101  047111  EM102:  .ASCIZ  /LOADING SLU1 TBUF DID NOT CLEAR INTERRUPT/
4720  024372  020107  046123  030525
4721  024400  052040  052502  020106
4722  024406  044504  020104  047516
4723  024414  020124  046103  040505
4724  024422  020122  047111  042524
4725  024430  051122  050125  000124
4726  024436  046123  030525  051040  EM103:  .ASCIZ  /SLU1 RCVR INTERRUPTS WITH ENABLE CLEAR/
4727  024444  053103  020122  047111
4728  024452  042524  051122  050125
4729  024460  051524  053440  052111
4730  024466  020110  047105  041101
4731  024474  042514  041440  042514
4732  024502  051101     000
4733  024505     123  052514  020061  EM104:  .ASCIZ  /SLU1 RCVR DID NOT INTERRUPT/
4734  024512  041522  051126  042040
4735  024520  042111  047040  052117
4736  024526  044440  052116  051105
4737  024534  052522  052120     000
4738  024541     123  052514  020061  EM105:  .ASCIZ  /SLU1 RCVR INTERRUPTS AT PRIORITY 7/
4739  024546  041522  051126  044440
4740  024554  052116  051105  052522
4741  024562  052120  020123  052101
4742  024570  050040  044522  051117
4743  024576  052111  020131  000067
4744  024604  046123  030525  051040  EM106:  .ASCIZ  /SLU1 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR/
4745  024612  053103  020122  047111
4746  024620  042524  051122  050125
4747  024626  051524  053440  052111
```

```
 4748  024634  020110  047111  042524
 4749  024642  051122  050125  020124
 4750  024650  047105  041101  042514
 4751  024656  041440  042514  051101
 4752  024664     000
 4753  024665     123  052514  020061  EM107:  .ASCIZ  /SLU1 RCVR DID NOT INTERRUPT/
 4754  024672  041522  051126  042040
 4755  024700  042111  047040  052117
 4756  024706  044440  052116  051105
 4757  024714  052522  052120     000
 4758  024721     123  052514  020061  EM110:  .ASCIZ  /SLU1 RECEIVER RE-INTERRUPTED/
 4759  024726  042522  042503  053111
 4760  024734  051105  051040  026505
 4761  024742  047111  042524  051122
 4762  024750  050125  042524  000104
 4763  024756  046123  030525  051040  EM111:  .ASCIZ  /SLU1 READING RBUF DID NOT CLEAR INTERRUPT/
 4764  024764  040505  044504  043516
 4765  024772  051040  052502  020106
 4766  025000  044504  020104  047516
 4767  025006  020124  046103  040505
 4768  025014  020122  047111  042524
 4769  025022  051122  050125  000124
 4770  025030  042522  042523  020124  EM112:  .ASCIZ  /RESET DID NOT CLEAR SLU1 RCVR INTERRUPT/
 4771  025036  044504  020104  047516
 4772  025044  020124  046103  040505
 4773  025052  020122  046123  030525
 4774  025060  051040  053103  020122
 4775  025066  047111  042524  051122
 4776  025074  050125  000124
 4777  025100  046123  030525  023440  EM113:  .ASCIZ  /SLU1 'OR' FLAG DID NOT SET/
 4778  025106  051117  020047  046106
 4779  025114  043501  042040  042111
 4780  025122  047040  052117  051440
 4781  025130  052105     000
 4782  025133     123  052514  020061  EM114:  .ASCIZ  /SLU1 'ERROR' NOT SET WITH 'OR' FLAG/
 4783  025140  042447  051122  051117
 4784  025146  020047  047516  020124
 4785  025154  042523  020124  044527
 4786  025162  044124  023440  051117
 4787  025170  020047  046106  043501
 4788  025176     000
 4789  025177     104  052101  020101  EM115:  .ASCIZ  /DATA COMPARE ERROR/
 4790  025204  047503  050115  051101
 4791  025212  020105  051105  047522
 4792  025220  000122
 4793  025222  046123  031125  054040  EM116:  .ASCIZ  /SLU2 XMIT INTERRUPTS WHEN DISABLED/
 4794  025230  044515  020124  047111
 4795  025236  042524  051122  050125
 4796  025244  051524  053440  042510
 4797  025252  020116  044504  040523
 4798  025260  046102  042105     000
 4799  025265     123  052514  020062  EM117:  .ASCIZ  /SLU2 XMIT DID NOT INTERRUPT/
 4800  025272  046530  052111  042040
 4801  025300  042111  047040  052117
 4802  025306  044440  052116  051105
 4803  025314  052522  052120     000
```

```
4804  025321     123  052514  020062  EM120:  .ASCIZ  /SLU2 XMIT INTERRUPT AT PRIORITY 7/
4805  025326  046530  052111  044440
4806  025334  052116  051105  052522
4807  025342  052120  040440  020124
4808  025350  051120  047511  044522
4809  025356  054524  033440     000
4810  025363     123  052514  020062  EM121:  .ASCIZ  /SLU2 XMIT INTERRUPTS WITH ENABLE CLEAR/
4811  025370  046530  052111  044440
4812  025376  052116  051105  052522
4813  025404  052120  020123  044527
4814  025412  044124  042440  040516
4815  025420  046102  020105  046103
4816  025426  040505  000122
4817  025432  046123  031125  054040  EM122:  .ASCIZ  /SLU2 XMIT DID NOT INTERRUPT/
4818  025440  044515  020124  044504
4819  025446  020104  047516  020124
4820  025454  047111  042524  051122
4821  025462  050125  000124
4822  025466  046123  031125  054040  EM123:  .ASCIZ  /SLU2 XMIT RE-INTERRUPTED/
4823  025474  044515  020124  042522
4824  025502  044455  052116  051105
4825  025510  052522  052120  042105
4826  025516     000
4827  025517     114  040517  044504  EM124:  .ASCIZ  /LOADING SLU2 TBUF DID NOT CLEAR INTERRUPT/
4828  025524  043516  051440  052514
4829  025532  020062  041124  043125
4830  025540  042040  042111  047040
4831  025546  052117  041440  042514
4832  025554  051101  044440  052116
4833  025562  051105  052522  052120
4834  025570     000
4835  025571     123  052514  020062  EM125:  .ASCIZ  /SLU2 RCVR INTERRUPTS WITH ENABLE CLEAR/
4836  025576  041522  051126  044440
4837  025604  052116  051105  052522
4838  025612  052120  020123  044527
4839  025620  044124  042440  040516
4840  025626  046102  020105  046103
4841  025634  040505  000122
4842  025640  046123  031125  051040  EM126:  .ASCIZ  /SLU2 RCVR DID NOT INTERRUPT/
4843  025646  053103  020122  044504
4844  025654  020104  047516  020124
4845  025662  047111  042524  051122
4846  025670  050125  000124
4847  025674  046123  031125  051040  EM127:  .ASCIZ  /SLU2 RCVR INTERRUPTS AT PRIORITY 7/
4848  025702  053103  020122  047111
4849  025710  042524  051122  050125
4850  025716  051524  040440  020124
4851  025724  051120  047511  044522
4852  025732  054524  033440     000
4853  025737     123  052514  020062  EM130:  .ASCIZ  /SLU2 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR/
4854  025744  041522  051126  044440
4855  025752  052116  051105  052522
4856  025760  052120  020123  044527
4857  025766  044124  044440  052116
4858  025774  051105  052522  052120
4859  026002  042440  040516  046102
```

```
4860  026010  020105  046103  040505
4861  026016  000122
4862  026020  046123  031125  051040  EM131:  .ASCIZ  /SLU2 RCVR DID NOT INTERRUPT/
4863  026026  053103  020122  044504
4864  026034  020104  047516  020124
4865  026042  047111  042524  051122
4866  026050  050125  000124
4867  026054  046123  031125  051040  EM132:  .ASCIZ  /SLU2 RECEIVER RE-INTERRUPTED/
4868  026062  041505  044505  042526
4869  026070  020122  042522  044455
4870  026076  052116  051105  052522
4871  026104  052120  042105     000
4872  026111     123  052514  020062  EM133:  .ASCIZ  /SLU2 READING RBUF DID NOT CLEAR INTERRUPT/
4873  026116  042522  042101  047111
4874  026124  020107  041122  043125
4875  026132  042040  042111  047040
4876  026140  052117  041440  042514
4877  026146  051101  044440  052116
4878  026154  051105  052522  052120
4879  026162     000
4880  026163     122  051505  052105  EM134:  .ASCIZ  /RESET DID NOT CLEAR SLU2 RCVR INTERRUPT/
4881  026170  042040  042111  047040
4882  026176  052117  041440  042514
4883  026204  051101  051440  052514
4884  026212  020062  041522  051126
4885  026220  044440  052116  051105
4886  026226  052522  052120     000
4887  026233     123  052514  020062  EM135:  .ASCIZ  /SLU2 'OR' FLAG DID NOT SET/
4888  026240  047447  023522  043040
4889  026246  040514  020107  044504
4890  026254  020104  047516  020124
4891  026262  042523  000124
4892  026266  046123  031125  023440  EM136:  .ASCIZ  /SLU2 'ERROR' NOT SET WITH 'OR' FLAG/
4893  026274  051105  047522  023522
4894  026302  047040  052117  051440
4895  026310  052105  053440  052111
4896  026316  020110  047447  023522
4897  026324  043040  040514  000107
4898  026332  046123  031125  041040  EM137:  .ASCIZ  /SLU2 BREAK DID NOT TRANSMIT ALL ZEROES/
4899  026340  042522  045501  042040
4900  026346  042111  047040  052117
4901  026354  052040  040522  051516
4902  026362  044515  020124  046101
4903  026370  020114  042532  047522
4904  026376  051505     000
4905  026401     102  042522  045501  EM140:  .ASCIZ  /BREAK DID NOT SET FRAMING ERROR/
4906  026406  042040  042111  047040
4907  026414  052117  051440  052105
4908  026422  043040  040522  044515
4909  026430  043516  042440  051122
4910  026436  051117     000
4911  026441     123  052514  020062  EM141:  .ASCIZ  /SLU2 'ERROR' NOT SET WITH 'FR' FLAG/
4912  026446  042447  051122  051117
4913  026454  020047  047516  020124
4914  026462  042523  020124  044527
4915  026470  044124  023440  051106
```

```
4916  026476  020047  046106  043501
4917  026504     000
4918  026505     104  052101  020101  EM142:  .ASCIZ  /DATA COMPARE ERROR WITH CABLE/
4919  026512  047503  050115  051101
4920  026520  020105  051105  047522
4921  026526  020122  044527  044124
4922  026534  041440  041101  042514
4923  026542     000
4924  026543     122  041524  044440  EM143:  .ASCIZ  /RTC INTERRUPT AT PRIORITY 7/
4925  026550  052116  051105  052522
4926  026556  052120  040440  020124
4927  026564  051120  047511  044522
4928  026572  054524  033440     000
4929  026577     122  041524  044440  EM144:  .ASCIZ  /RTC INTERRUPTS WHEN DISABLED/
4930  026604  052116  051105  052522
4931  026612  052120  020123  044127
4932  026620  047105  042040  051511
4933  026626  041101  042514  000104
4934  026634  052122  020103  047111  EM145:  .ASCIZ  /RTC INTERRUPT DID NOT OCCUR/
4935  026642  042524  051122  050125
4936  026650  020124  044504  020104
4937  026656  047516  020124  041517
4938  026664  052503  000122
4939  026670  052122  020103  047111  EM146:  .ASCIZ  /RTC INTERRUPT DID NOT OCCUR/
4940  026676  042524  051122  050125
4941  026704  020124  044504  020104
4942  026712  047516  020124  041517
4943  026720  052503  000122
4944  026724  052122  020103  047504  EM147:  .ASCIZ  /RTC DOUBLE INTERRUPT/
4945  026732  041125  042514  044440
4946  026740  052116  051105  052522
4947  026746  052120     000
4948  026751     122  051505  052105  EM150:  .ASCIZ  /RESET DID NOT CLEAR RTC INTERRUPT/
4949  026756  042040  042111  047040
4950  026764  052117  041440  042514
4951  026772  051101  051040  041524
4952  027000  044440  052116  051105
4953  027006  052522  052120     000
4954  027013     122  041524  044440  EM151:  .ASCIZ  /RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS/
4955  027020  052116  051105  052522
4956  027026  052120  042040  042111
4957  027034  047040  052117  041440
4958  027042  042514  051101  053440
4959  027050  052111  020110  044502
4960  027056  033524  047440  020106
4961  027064  045514  000123
4962  027070  046103  041517  020113  EM152:  .ASCIZ  /CLOCK REPEATABILITY ERROR/
4963  027076  042522  042520  052101
4964  027104  041101  046111  052111
4965  027112  020131  051105  047522
4966  027120  000122
4967  027122  046123  030525  051040  EM153:  .ASCIZ  /SLU1 RECEIVER STATUS ERROR/
4968  027130  041505  044505  042526
4969  027136  020122  052123  052101
4970  027144  051525  042440  051122
4971  027152  051117     000
```

```
 4972  027155     123  052514  020062  EM154:  .ASCIZ  /SLU2 RECEIVER STATUS ERROR/
 4973  027162  042522  042503  053111
 4974  027170  051105  051440  040524
 4975  027176  052524  020123  051105
 4976  027204  047522  000122
 4977  027210  047111  047503  051122  EM155:  .ASCIZ  /INCORRECT RECEIVE COUNT SLU1/
 4978  027216  041505  020124  042522
 4979  027224  042503  053111  020105
 4980  027232  047503  047125  020124
 4981  027240  046123  030525     000
 4982  027245     123  052514  020061  EM156:  .ASCIZ  /SLU1 DATA COMPARE ERROR IN EXERCISER/
 4983  027252  040504  040524  041440
 4984  027260  046517  040520  042522
 4985  027266  042440  051122  051117     -
 4986  027274  044440  020116  054105
 4987  027302  051105  044503  042523
 4988  027310  000122
 4989  027312  047111  047503  051122  EM157:  .ASCIZ  /INCORRECT RECEIVE COUNT SLU2/
 4990  027320  041505  020124  042522
 4991  027326  042503  053111  020105
 4992  027334  047503  047125  020124
 4993  027342  046123  031125     000
 4994  027347     123  052514  020062  EM160:  .ASCIZ  /SLU2 DATA COMPARE ERROR IN EXERCISER/
 4995  027354  040504  040524  041440
 4996  027362  046517  040520  042522
 4997  027370  042440  051122  051117
 4998  027376  044440  020116  054105
 4999  027404  051105  044503  042523
 5000  027412  000122
 5001  027414  051124  050101  041440  EM161:  .ASCIZ  /TRAP CATCHER/
 5002  027422  052101  044103  051105
 5003  027430     000
 5004
 5005  027431     124  051505  021524  DH5:    .ASCIZ  /TEST#   ERR PC  TCSR/
 5006  027436  020040  042440  051122
 5007  027444  050040  020103  052040
 5008  027452  051503  000122
 5009  027456  042524  052123  020043  DH6:    .ASCIZ  /TEST#   ERR PC  RCSR/
 5010  027464  020040  051105  020122
 5011  027472  041520  020040  041522
 5012  027500  051123     000
 5013  027503     124  051505  021524  DH7:    .ASCIZ  /TEST#   ERR PC  RBUF/
 5014  027510  020040  042440  051122
 5015  027516  050040  020103  051040
 5016  027524  052502  000106
 5017  027530  042524  052123  020043  DH32:   .ASCIZ  /TEST#   ERR PC  TBUF/
 5018  027536  020040  051105  020122
 5019  027544  041520  020040  041124
 5020  027552  043125     000
 5021  027555     124  051505  021524  DH61:   .ASCIZ  /TEST#   ERR PC  LKS/
 5022  027562  020040  042440  051122
 5023  027570  050040  020103  046040
 5024  027576  051513     000
 5025  027601     124  051505  021524  DH72:   .ASCIZ  /TEST#   ERR PC  GDADR   BDADR   GDDAT   BDDAT/
 5026  027606  020040  042440  051122
 5027  027614  050040  020103  043440
```

```
                                                                L 8
 5028  027622  040504  051104  020040
 5029  027630  041040  040504  051104
 5030  027636  020040  043440  042104
 5031  027644  052101  020040  041040
 5032  027652  042104  052101     000
 5033  027657     124  051505  021524  DH115:  .ASCIZ  /TEST#   ERR PC  RCSR    GOOD    BAD/
 5034  027664  020040  042440  051122
 5035  027672  050040  020103  051040
 5036  027700  051503  020122  020040
 5037  027706  043440  047517  020104
 5038  027714  020040  041040  042101
 5039  027722     000
 5040  027723     124  051505  021524  DH137:  .ASCIZ  /TEST#   ERR PC  RCSR    DATA/
 5041  027730  020040  042440  051122
 5042  027736  050040  020103  051040
 5043  027744  051503  020122  020040
 5044  027752  042040  052101  000101
 5045  027760  042524  052123  020043  DH142:  .ASCIZ  /TEST#   ERR PC  GOOD    BAD/
 5046  027766  020040  051105  020122
 5047  C27774  041520  020040  047507
 5048  030002  042117  020040  020040
 5049  030010  040502  000104
 5050  030014  042524  052123  020043  DH152:  .ASCIZ  /TEST#   ERR PC  LKS     CNT1    CNT2/
 5051  030022  020040  051105  020122
 5052  030030  041520  020040  045514
 5053  030036  020123  020040  020040
 5054  030044  047103  030524  020040
 5055  030052  020040  047103  031124
 5056  030060     000
 5057  030061     124  051505  021524  DH155:  .ASCIZ  /TEST#   ERR PC  RCSR    TRANS   RCV/
 5058  030066  020040  042440  051122
 5059  030074  050040  020103  051040
 5060  030102  051503  020122  020040
 5061  030110  052040  040522  051516
 5062  030116  020040  051040  053103
 5063  030124     000
 5064  030125     124  051505  021524  DH161:  .ASCIZ  /TEST#   ERR PC  RCSR    OLDPC   TRAP ADR/
 5065  030132  020040  042440  051122
 5066  030140  050040  020103  051040
 5067  030146  051503  020122  020040
 5068  030154  047440  042114  041520
 5069  030162  020040  052040  040522
 5070  030170  020120  042101  000122
 5071  030176  005015  045103  042113  M1:     .ASCIZ  <CR><LF>/CJKDF-A/
 5072  030204  026506  000101
 5073  030210     200                  M2:     .ASCII  <CRLF>
 5074
 5075          030212                          .EVEN
 5076  030212  001104  001016  003004  DT5:    .WORD   $TESTN,$ERRPC,CTCSR,0
 5077  030220  000000
 5078  030222  001104  001016  003000  DT6:    .WORD   $TESTN,$ERRPC,CRCSR,0
 5079  030230  000000
 5080  030232  001104  001016  003002  DT7:    .WORD   $TESTN,$ERRPC,CRBUF,0
 5081  030240  000000
 5082  030242  001104  001016  002774  DT31:   .WORD   $TESTN,$ERRPC,TCSR,0
 5083  030250  000000
```

```
5084  030252  001104  001016  002776  DT32:   .WORD   $TESTN,$ERRPC,TBUF,0
5085  030260  000000
5086  030262  001104  001016  002770  DT36:   .WORD   $TESTN,$ERRPC,RCSR,0
5087  030270  000000
5088  030272  001104  001016  002772  DT37:   .WORD   $TESTN,$ERRPC,RBUF,0
5089  030300  000000
5090  030302  001104  001016  003010  DT61:   .WORD   $TESTN,$ERRPC,LKS,0
5091  030310  000000
5092  030312  001104  001016  001020  DT72:   .WORD   $TESTN,$ERRPC,$GDADR,$BDADR,$GDDAT,$BDDAT,0
5093  030320  001022  001024  001026
5094  030326  000000
5095  030330  001104  001016  003000  DT115:  .WORD   $TESTN,$ERRPC,CRCSR,$GDDAT,$BDDAT,0
5096  030336  001024  001026  000000
5097  030344  001104  001016  002770  DT137:  .WORD   $TESTN,$ERRPC,RCSR,$BDDAT,0
5098  030352  001026  000000
5099  030356  001104  001016  001024  DT142:  .WORD   $TESTN,$ERRPC,$GDDAT,$BDDAT,0
5100  030364  001026  000000
5101  030370  001104  001016  003010  DT152:  .WORD   $TESTN,$ERRPC,LKS,FIRST,SECND,0
5102  030376  013574  013576  000000
5103  030404  001104  001016  003000  DT155:  .WORD   $TESTN,$ERRPC,CRCSR,XMTCT1,RECCT1,0
5104  030412  014726  014732  000000
5105  030420  001104  001016  002770  DT157:  .WORD   $TESTN,$ERRPC,RCSR,XMTCT2,RECCT2,0
5106  030426  014730  014734  000000
5107  030434  001104  001016  002770  DT161:  .WORD   $TESTN,$ERRPC,RCSR,OLDPC,BDVECT,0
5108  030442  015114  015116  000000
5109  030450  000200                  BUF1:   .BLKW   200             ;SLU1 INPUT BUFFER FOR BLAST TEST
5110  031050  000200                  BUF2:   .BLKW   200             ;SLU2 INPIT BUFFER FOR BLAST TEST
5111          000001                          .END
```

```
ABASE = 176500          817#    935     976
ACDW1 = 000000          935
ACDW2 = 000000          935
ACPUOP= 000000          935     950
ADDW0 = 000000          935
ADDW1 = 000000          935
ADDW10= 000000          935
ADDW11= 000000          935
ADDW12= 000000          935
ADDW13= 000000          935
ADDW14= 000000          935
ADDW15= 000000          935
ADDW2 = 000000          935
ADDW3 = 000000          935
ADDW4 = 000000          935
ADDW5 = 000000          935
ADDW6 = 000000          935
ADDW7 = 000000          935
ADDW8 = 000000          935
ADDW9 = C00000          935
ADEVCT= 000000          935     941
ADEVM = 000000          935     977
AENV  = 000000          935     946
AENVM = 000000          935     947
AFATAL= 000000          935     938
AMADR1= 000000          935     963
AMADR2= 000000          935     967
AMADR3= 000000          935     970
AMADR4= 000000          935     973
AMAMS1= 000000          935     957
AMAMS2= 000000          935     965
AMAMS3= 000000          935     968
AMAMS4= 000000          935     971
AMSGAD= 000000          935     943
AMSGLG= 000000          935     944
AMSGTY= 000000          935     937
AMTYP1= 000000          935     958
AMTYP2= 000000          935     966
AMTYP3= 000000          935     969
AMTYP4= 000000          935     972
APASS = 000000          935     940
APRIOR= 000000          935
APTCSU= 000040          3820#   3853
APTENV= 000001          3596    3775    3818#   3846
APTSIZ= 000200          1609    3817#
APTSPO= 000100          3777    3819#   3848
ASWREG= 000000          935     948
ATESTN= 000000          935     939
AUNIT = 000000          935     942
AUSWR = 000400          819#    935     949
AVECT1= 000300          818#    935     974
AVECT2= 000000          935     975
BDVECT  015116          3562*   3568#   5107
BIT0  = 000001          802#    1626    1633    1720    2053    2054    2057    2058    2061    2062    2071    2073    2075
                        2380    3063    3070    3075    3079    3089    3093    3349    3376    3451    3469
BIT00 = 000001          792#    802
```

```
BIT01 = 000002          791#    801
BIT02 = 000004          790#    800
BIT03 = 000010          789#    799
BIT04 = 000020          788#    798
BIT05 = 000040          787#    797
BIT06 = 000100          786#    796
BIT07 = 000200          785#    795
BIT08 = 000400          784#    794
BIT09 = 001000          783#    793   3607   3746
BIT1  = 000002          801#   1629   1938   2739   3382   3485
BIT10 = 002000          782#   2683   3034   3087
BIT11 = 004000          781#
BIT12 = 010000          780#   4273
BIT13 = 020000          779#   3094   3591
BIT14 = 040000          778#   2694   3042   3730
BIT15 = 100000          777#   2699   3047   3099
BIT2  = 000004          800#   1632   1773   1774   1779   1780   1783   1784   1789   1791   1793   1893   1900
                       1908    1912   1921   1926   2228   2515   2523   2526   2534   2539   2554   2561   2571
                       2574    2593   2597   2630   2634   2660   2686   2693   2713   2726   2729   3145   3378
                       3387    3443
BIT3  = 000010          799#   1627
BIT4  = 000020          798#   1624   4261   4330
BIT5  = 000040          797#   1624   1627   3336
BIT6  = 000100          796#   1630   1803   1808   1818   1819   1825   1826   1832   1834   1847   1852   1862
                       1863    1869   1870   1877   1879   2084   2089   2099   2100   2106   2107   2117   2119
                       2132    2137   2147   2148   2154   2155   2166   2168   2247   2252   2262   2263   2269
                       2270    2279   2281   2344   2347   2385   2398   2403   2413   2420   2427   2445   2454
                       2464    2478   2483   2491   2513   2514   2532   2538   2558   2566   2598   2609   2629
                       2639    2659   2668   2743   2756   2761   2771   2778   2785   2805   2814   2824   2843
                       2848    2856   2876   2877   2892   2897   2914   2921   2948   2959   2978   2986   3011
                       3019    3156   3179   3187   3210   3228   3248   3255   3279   3286   3308   3379   3380
                       3384    3385   3389   3403   3426   3447   3448   3449   3453   3457
BIT7  = 000200          795#   2307   2308   2310   2311   3110   3155   3178   3209   3247   3278   3282   4288
                       4349
BIT8  = 000400          794#
BIT9  = 001000          793#
BLAST   013600         3147    3342#
BPT   = 000003          822#
BPTVEC= 000014          809#
BUF1    030450         3381    3412   3474   5109#
BUF2    031050         3386    3435   3490   5110#
CATCH   015070          833    3560#
CHECK1  014524         3469#
CHECK2  014604         3470    3485#
CKSWR = 104407         3606    3729   4248#
CMPARE  013532         3323    3326#
CONT    006510         2317#   2320
CONT41  012330         3066    3073#
COUNT   014736         3365*   3396*  3445   3512#
CR    = 000015          717#   3909   3919   4306   4308   4310   4312   5071
CRBUF   003002         1547#   1759   1925   2635   2694   2699   2718   3407   3409   3418   5080
CRCSR   003000         1546#   1744   1847*  1852   1862*  1863   1869*  1870   1877*  1879   1896   1905   1912*
                       1923    1927   2514*  2521   2532*  2538*  2556   2558*  2566*  2595   2598*  2609*  2629*
                       2632    2639*  2659*  2662   2668*  2716   3380*  4265   4334   5078   5095   5103
CRLF  = 000200          718#   3863   3919   5073
CRPSW   003024         1560#   2588   2590*  2610*  3369*
```

```
CRVECT  003022  1559#  1848   1849*  1884*  2516   2517*  2531*  2541*  2552   2553*  2567*  2578*  2587
                2589*  2605*  2616*  2627   2628*  2645*  2657   2658*  2675*  3359   3368*  3504*
CTBUF   003006  1549#  1662   1681*  1686*  1724*  1727*  1895*  1922*  2484*  2487*  2520*  2555*  2594*
                2631*  2661*  2688*  2715*  3400*
CTCSR   003004  1548#  1645   1682   1687   1693   1725   1730   1773*  1774   1779*  1780   1783*  1784
                1789*  1791   1793*  1803*  1808   1818*  1819   1825*  1826   1832*  1834   1893*  1900*
                1908*  1921*  1926*  2385*  2388   2398*  2403*  2413*  2418   2420*  2427*  2445*  2452
                2454*  2464*  2478*  2483*  2485   2491*  2513*  2515*  2523*  2526*  2534*  2539*  2554*
                2561*  2571*  2574*  2593*  2597*  2630*  2634*  2660*  2686*  2689   2693*  2713*  2726*
                2729*  3378*  3379*  3403*  3447*  3453   5076
CTPSW   003030  1562#  2447   2449*  2471*  3367*
CTVECT  003026  1561#  1804   1805*  1839*  2386   2387*  2397*  2405*  2416   2417*  2428*  2437*  2446
                2448*  2460*  2470*  2481   2482*  2497*  3358   3366*  3503*
CWDONE  004602  1896#  1899
DDISP = 177570   724#   913   1597
DH115   027657  1356   1506   1521   5033#
DH137   027723  1446   5040#
DH142   027760  1461   1511   1531   5045#
DH152   030014  1501   5050#
DH155   030061  1516   1526   5057#
DH161   030125  1536   5064#
DH32    027530  1103   5017#
DH5     027431  1002   1020   1028   1033   1038   1043   1048   1053   1098   1108   1113   1118   1133
                1138   1143   1148   1153   1158   1163   1168   1173   1271   1276   1281   1286   1291
                1296   1301   1361   1366   1371   1376   1381   1386   1391   5005#
DH6     027456  1007   1058   1063   1068   1073   1078   1083   1088   1093   1123   1178   1183   1188
                1193   1198   1203   1208   1213   1306   1311   1316   1321   1326   1331   1336   1341
                1346   1351   1396   1401   1406   1411   1416   1421   1426   1431   1436   1441   1451
                1456   5009#
DH61    027555  1218   1223   1228   1233   1238   1243   1248   1253   1258   1466   1471   1476   1481
                1486   1491   1496   5021#
DH7     027503  1012   1128   5013#
DH72    027601  1263   5025#
DISPLA  001042   913#  1597*  1605*  3586*  3757*  4260*  4329*
DISPRE  000174   843#  1605   4260   4329
DSWR  = 177570   723#   912   1596
DT115   030330  1357   1507   1522   5095#
DT137   030344  1447   5097#
DT142   030356  1462   1512   1532   5099#
DT152   030370  1502   5101#
DT155   030404  1517   5103#
DT157   030420  1527   5105#
DT161   030434  1537   5107#
DT31    030242  1099   1109   1114   1119   1134   1139   1144   1149   1154   1159   1164   1169   1174
                1362   1367   1372   1377   1382   1387   1392   5082#
DT32    030252  1104   5084#
DT36    030262  1124   1179   1184   1189   1194   1199   1204   1209   1214   1397   1402   1407   1412
                1417   1422   1427   1432   1437   1442   1457   5086#
DT37    030272  1129   5088#
DT5     030212  1003   1021   1029   1034   1039   1044   1049   1054   1272   1277   1282   1287   1292
                1297   1302   5076#
DT6     030222  1008   1059   1064   1069   1074   1079   1084   1089   1094   1307   1312   1317   1322
                1327   1332   1337   1342   1347   1352   1452   5078#
DT61    030302  1219   1224   1229   1234   1239   1244   1249   1254   1259   1467   1472   1477   1482
                1487   1492   1497   5090#
DT7     030232  1013   5080#
```

```
DT72    030312          1264    5092#
ECHO    020106           848    4259#   4294
EMTVEC= 000030           812#   1581*   1582*
EM100   024277          1290    4709#
EM101   024333          1295    4714#
EM102   024364          1300    4719#
EM103   024436          1305    4726#
EM104   024505          1310    4733#
EM105   024541          1315    4738#
EM106   024604          1320    4744#
EM107   024665          1325    4753#
EM11    020650          1019    4376#
EM110   024721          1330    4758#
EM111   024756          1335    4763#
EM112   025030          1340    4770#
EM113   025100          1345    4777#
EM114   025133          1350    4782#
EM115   025177          1355    4789#
EM116   025222          1360    4793#
EM117   025265          1365    4799#
EM120   025321          1370    4804#
EM121   025363          1375    4810#
EM122   025432          1380    4817#
EM123   025466          1385    4822#
EM124   025517          1390    4827#
EM125   025571          1395    4835#
EM126   025640          1400    4842#
EM127   025674          1405    4847#
EM13    020706          1027    4381#
EM130   025737          1410    4853#
EM131   026020          1415    4862#
EM132   026054          1420    4867#
EM133   026111          1425    4872#
EM134   026163          1430    4880#
EM135   026233          1435    4887#
EM136   026266          1440    4892#
EM137   026332          1445    4898#
EM14    020754          1032    4388#
EM140   026401          1450    4905#
EM141   026441          1455    4911#
EM142   026505          1460    4918#
EM143   026543          1465    4924#
EM144   026577          1470    4929#
EM145   026634          1475    4934#
EM146   026670          1480    4939#
EM147   026724          1485    4944#
EM15    021024          1037    4395#
EM150   026751          1490    4948#
EM151   027013          1495    4954#
EM152   027070          1500    4962#
EM153   027122          1505    4967#
EM154   027155          1510    4972#
EM155   027210          1515    4977#
EM156   027245          1520    4982#
EM157   027312          1525    4989#
EM16    021072          1042    4402#
```

```
EM160   027347          1530    4994#
EM161   027414          1535    5001#
EM17    021130          1047    4407#
EM20    021170          1052    4413#
EM21    021236          1057    4420#
EM22    021306          1062    4427#
EM23    021352          1067    4433#
EM24    021410          1072    4438#
EM25    021450          1077    4444#
EM26    021523          1082    4452#
EM27    021560          1087    4457#
EM30    021623          1092    4463#
EM31    021675          1097    4471#
EM32    021734          1102    4477#
EM33    021773          1107    4483#
EM34    022045          1112    4491#
EM35    022113          1117    4498#
EM36    022155          1122    4504#
EM37    022214          1127    4510#
EM40    022253          1132    4516#
EM41    022323          1137    4523#
EM42    022361          1142    4529#
EM43    022421          1147    4535#
EM44    022467          1152    4542#
EM45    022537          1157    4549#
EM46    022605          1162    4556#
EM47    022643          1167    4562#
EM5     020510          1001    4358#
EM50    022703          1172    4568#
EM51    022751          1177    4575#
EM52    023021          1182    4582#
EM53    023065          1187    4589#
EM54    023123          1192    4595#
EM55    023163          1197    4601#
EM56    023236          1202    4609#
EM57    023273          1207    4614#
EM6     020552          1006    4364#
EM60    023336          1212    4620#
EM61    023410          1217    4627#
EM62    023441          1222    4632#
EM63    023503          1227    4638#
EM64    023541          1232    4644#
EM65    023571          1237    4649#
EM66    023623          1242    4654#
EM67    023663          1247    4660#
EM7     020611          1011    4370#
EM70    023723          1252    4666#
EM71    023755          1257    4671#
EM72    024006          1262    4676#
EM74    024067          1270    4685#
EM75    024132          1275    4691#
EM76    024166          1280    4696#
EM77    024230          1285    4702#
ERRVEC= 000004           805#   1594    1595*   1606*   3735    3736*   3738*   3741*
FINIE   014664          3486    3501#
FIRST   013574          3324*   3326    3340#   5101
```

```
GNS    = ****** U    4240  4241  4242  4243  4244  4246  4248  4249  4250
GTSWR  = 104406      1620  4246#
HT     = 000011       715#  3861  3919
ID       003700      1635  1694  1703#
IOHAND   014400      3397  3404  3419  3427  3443#
IOTVEC = 000020       810#  1579* 1580*
LF     = 000012       716#  3913  3919  4306  4308  4310  4312  5071
LKS      003010     1550#  2234  2247* 2252  2262* 2263  2269* 2270  2279* 2281  2302  2307* 2308
                    2310*  2311  2317  2350  3155* 3156* 3157  3165* 3169  3178* 3179* 3180  3187*
                    3209*  3210* 3211  3228* 3247* 3248* 3249  3255* 3278* 3279* 3280  3282* 3286*
                    3308*  3312* 3313  3315* 3316  3319* 3389* 3449* 5090  5101
LTCIT    012544     2741  3138#
LTCRT    006152     1940  2221#
MFR      020327     4282  4308#
MOR      020341     4287  4310#
MPAR     020316     4277  4306#
MSG      020270     4276  4281  4286  4291  4297#  4344
MSTOP    020354     4292  4312#
M1       030176     1710  5071#
M2       030210     1712  5073#
OLDPC    015114     3563* 3567#  5107
OUTTST   020364      849  4328#  4355
PIRQ   = 177772      722#
PIRQVE = 000240      816#
PR0    = 000000      739#
PR1    = 000040      740#
PR2    = 000100      741#
PR3    = 000140      742#
PR4    = 000200      743#
PR5    - 000240      744#
PR6    = 000300      745#
PR7    = 000340      746#
PS     - 177776      719#   720  2336* 3355* 3392* 3462*
PSW    = 177776      720#
PWRVEC = 000024       811#  1585* 1586* 3674* 3675* 3684* 3692* 3701* 3702*
RBUF     002772     1543#  2039  2182* 2203  2212* 2216  2865  2982  3042  3047  3062  3073  3094
                    3099   3126  3430  3432  3441  5088
RCSR     002770     1542#  2023  2132* 2137  2147* 2148  2154* 2155  2166* 2168  2185  2198  2214
                    2217   2338  2339  2345  2877* 2883  2892* 2897* 2912  2914* 2921* 2946  2948*
                    2959*  2978* 2980  2986* 3011* 3013  3019* 3060  3065  3091  3119  3385* 4263
                    4332   5086  5097  5105  5107
RDCHR  - 104410     4180  4249#
RDLIN  - 104411     4250#
RECCT1   014732     3363*  3406* 3471  3510#  5103
RECCT2   014734     3364*  3429* 3487  3511#  5105
REC1     014210     3368   3406#
REC2     014324     3372   3429#
RESVEC = 000010      806#
RPSW     003014     1556#  2940  2942* 2960* 3373*
RTCPSW   003034     1564#  3152  3154* 3189* 3204  3206* 3230* 3375*
RTCVT    003032     1563#  2248  2249* 2286* 3151  3153* 3166* 3177* 3188* 3203  3205* 3218* 3229*
                    3245   3246* 3261* 3276  3277* 3293* 3360  3374* 3505*
RVECT    003012     1555#  2133  2134* 2173* 2878  2879* 2891* 2899* 2909  2910* 2922* 2931* 2939
                    2941*  2955* 2966* 2976  2977* 2992* 3009  3010* 3026* 3357  3372* 3502*
SECND    013576     3333*  3341# 5101
SLU1IT   006700     2367#
```

```
SLU2IT  010600          2378    2382    2732#
SLU2RT  004726          1680    1722    1931#
STACK = 001100           710#
START   003036           847    1566#
STKLMT= 177774           721#
SWR     001040           912#   1575    1596*   1598    1604*   1611*   1618    1624    1627    1630    2683    3034    3087
                        3110    3336    3591    3603    3607    3682    3694*   3730    3746    4075    4112*   4259*   4261
                        4328*   4330
SWREG   000176           844#   1604    1618    4075    4088    4259    4328
SW0   = 000001           774#
SW00  = 000001           764#    774
SW01  = 000002           763#    773
SW02  = 000004           762#    772
SW03  = 000010           761#    771
SW04  = 000020           760#    770
SW05  = 000040           759#    769
SW06  = 000100           758#    768
SW07  = 000200           757#    767
SW08  = 000400           756#    766
SW09  = 001000           755#    765
SW1   = 000002           773#
SW10  = 002000           754#
SW11  = 004000           753#
SW12  = 010000           752#
SW13  = 020000           751#
SW14  = 040000           750#
SW15  = 100000           749#
SW2   = 000004           772#
SW3   = 000010           771#
SW4   = 000020           770#
SW5   = 000040           769#
SW6   = 000100           768#
SW7   = 000200           767#
SW8   = 000400           766#
SW9   = 001000           765#
TBITVE= 000014           807#
TBUF    002776          1545#   1959    1976*   1981*   2003*   2006*   2184*   2213*   2849*   2852*   2882*   2911*   2945*
                        2979*   3012*   3037*   3059*   3090*   3117*   3423*   5084
TCSR    002774          1544#   1944    1977    1982    1986    2004    2009    2053*   2054    2057*   2058    2061*   2062
                        2071*   2073    2075*   2084*   2089    2099*   2100    2106*   2107    2117*   2119    2743*   2746
                        2756*   2761*   2771*   2776    2778*   2785*   2805*   2812    2814*   2824*   2843*   2848*   2850
                        2856*   2876*   3038    3063*   3070*   3075*   3079*   3089*   3093*   3384*   3426*   3448*   3457
                        5082
TICKER  014144          3374    3396#
TKVEC = 000060           814#
TOLER   013544          3328    3330#
TPSW    003020          1558#   2807    2809*   2831*   3371*
TPVEC = 000064           815#
TRAPVE= 000034           813#   1583*   1584*
TRTVEC= 000014           808#
TST1    003450          1634    1641#   3547
TST10   004246          1792    1801#
TST11   004412          1845#
TST12   004556          1891#
TST13   004652          1918#
TST14   004726          1928    1936#
```

```
TST15   005002          1955#
TST16   005042          1969#
TST17   005134          1974    1987    1996#
TST2    003516          1658#
TST20   005214          2001    2010    2019#
TST21   005254          2035#
TST22   005314          2051#
TST23   005452          2069    2074    2082#
TST24   005634          2130#
TST25   006016          2180#
TST26   006110          2210#
TST27   006152          2218    2226#
TST3    003564          1674#
TST30   006226          2245#
TST31   006410          2294#
TST32   006524          2299    2318    2329#
TST33   006700          2334    2372#
TST34   007046          2411#
TST35   007166          2443#
TST36   007322          2476#
TST37   007434          2509#
TST4    003734          1700    1718#
TST40   007614          2547#
TST41   007764          2584#
TST42   010134          2622#
TST43   010256          2652#
TST44   010372          2681#
TST45   010476          2684    2709#
TST46   010600          2737#
TST47   010722          2769#
TST5    004012          1731    1740#
TST50   011042          2803#
TST51   011176          2836#
TST52   011340          2841    2874#
TST53   011460          2905#
TST54   011576          2937#
TST55   011730          2972#
TST56   012034          2999#
TST57   012160          3004    3032#
TST6    004052          1755#
TST60   012246          3035    3057#
TST61   012354          3085#
TST62   012446          3088    3108#
TST63   012544          3111    3130    3143#
TST64   012766          3196#
TST65   013140          3201    3236#
TST66   013262          3241    3267#
TST67   013416          3272    3301#
TST7    004112          1771#
TST70   013600          3306    3337    3347#
TVECT   003016          1557#   2085    2086*   2124*   2744    2745*   2755*   2763*   2774    2775*   2786*   2795*   2806
                        2808*   2820*   2830*   2846    2847*   2866*   3356    3370*   3501*
TYPDS = 104405          3536    4244#
TYPE  = 104401          1709    1711    3534    3537    3594    3629    3646    3648    3651    3653    3657    3664    3706
                        3866    3979    4054    4086    4087    4090    4103    4114    4133    4184    4187    4191    4240#
TYPOC = 104402          3637    3661    4089    4241#
```

```
TYPON = 104404          4243#
TYPOS = 104403          4242#
UNIQUE  006524          2230    2324#
WAIT    020276          4299#   4300    4303
WAITER  014500          3450    3455    3458    3462#
WAITIO  014140          3393#   3394
WDONE   006040          2185#   2188
WRPSW   015056          1806    1850    2087    2135    2250    2390    2414    2429    2450    2461    2479    2488    2518
                        2550    2568    2591    2606    2625    2636    2655    2665    2748    2772    2787    2810    2821
                        2844    2853    2880    2907    2923    2943    2956    2974    2983    3007    3016    3149    3167
                        3207    3219    3243    3252    3274    3283    3294    3553#
XMIT1   014154          3366    3398#
XMIT2   014270          3370    3421#
XMITCT1 014726          3361*   3398*   3401    3471    3476    3508#   5103
XMITCT2 014730          3362*   3421*   3424    3487    3492    3509#   5105
$APTHD  000500           871     877#
$ASTAT= ****** U        1648    1665    1689    1697    1776    1786
$ATYC   016012          3768    3770#
$ATY1   015766          3766#
$ATY3   015774          3767#   3851
$ATY4   016004          1648    1665    1689    1697    1776    1786    3599    3769#
$AUTOB  001034           909#   1622*   4083    4206
$BASE   001154           976#
$BDADR  001022           904#   2358*   5092
$BDDAT  001026           906#   2360*   2725*   3133*   3409*   3432*   3479*   3495*   5092    5095    5097    5099
$CHARC  016564          3868*   3878*   3885    3911*   3916#
$CKSWR  017242          4075#   4248
$CMTAG  001000           892#   1572    1573    1581    1587    1588
$CM3  - 000000           922#
$CM4  = 000005           922#    923#    924#    925#    926#    927#
$CNTLG  017777          4086    4201#
$CNTLU  017772          4103    4200#
$CPUOP  001126           950#
$CRLF   001075           929#   3594    3629    3648    3653    3657    3867    3919    4114    4200    4346
$DBLK   017232          4020    4054    4062#
$DEVCT  001110           941#   1707
$DEVM   001156           977#   1569*   1626*   1629*   1632*   1633    1720    1938    2228    2380    2739    3145    3376
                        3382    3387    3443    3451    3469    3485
$DOAGN  015032          3530    3539    3545#
$DTBL   017222          4023    4058#
$ENDAD  015022           858    3541#   3614
$ENDCT  014770          1587    3532#
$ENDMG  015041          3534    3549#
$ENULL  015036          3537    3548#
$ENV    001120           946#   1616    1676    1971    1998    2066    2112    2161    2192    2274    2296    2331    2374
                        2838    3001    3198    3238    3269    3303    3349    3596    3775    3799    3846
$ENVM   001121           947#   1609    3777    3848    3853
$EOP    014740          3338    3353    3506    3523#
$EOPCT  014762          1587*   3529#   3533
$ERFLG  001003           895#   3584*   3721    3744    3750*   3760
$ERMAX  001015           901#   1589*   3756*   3760
$ERROR  015120          1581    3583#
$ERRPC  001016           902#   3588*   3589*   3590    3635    5076    5078    5080    5082    5084    5086    5088    5090
                        5092    5095    5097    5099    5101    5103    5105    5107
$ERRTB  001160           994#   3643
$ERRTY  015302          3593    3628#
```

```
$ERTTL  001012          899#  3587*
$ESCAP  001072          927#  1588*  3610  3612  3755*
$ETABL  001120          945#
$ETEND  001160          883    978#
$FATAL  001102          938#  1566*  3803*
$FFLG   016232         3766*  3769*  3797  3806*  3814#
$FILLC  001056          920#  3871   3919
$FILLS  001055          919#  3919
$GDADR  001020          903#  2357*  5092
$GDDAT  001024          905#  2359*  2724*  3132*  3414*  3416*  3437*  3439*  3480*  3496*  5092  5095  5099
$GET42  015012         3538#
$GTSWR  017312         4087#  4246
$HD   = 000003          705    706
$HIBTS  000500          878#
$ICNT   001004          896#
$ILLUP  015576         3674   3692  3709#
$INTAG  001035          910#  4115  4206
$ITEMB  001014          900#  3590*  3598  3632
$LF     001076          930#  3919  4191  4200
$LFLG   016231         3807*  3813#
$LPADR  001006          897#  1590*  3748*  3753*  3758  3760
$LPERR  001010          898#  1591*  3609  3748  3754*  3760
$MADR1  001132          963#
$MADR2  001136          967#
$MADR3  001142          970#
$MADR4  001146          973#
$MAIL   001100          879    883   936#  1608  1616  1642  1659  1675  1719  1741  1756  1772  1802
                       1846   1892  1919  1937  1956  1970  1997  2020  2036  2052  2083  2131  2181
                       2211   2227  2246  2295  2330  2373  2412  2444  2477  2510  2548  2585  2623
                       2653   2682  2710  2738  2770  2804  2837  2875  2906  2938  2973  3000  3033
                       3058   3086  3109  3144  3197  3237  3268  3302  3348  3752  3846
$MAMS1  001130          957#
$MAMS2  001134          965#
$MAMS3  001140          968#
$MAMS4  001144          971#
$MBADR  000502          879#
$MFLG   016230         3767*  3773  3808*  3812#
$MNEW   020015         4090   4204#
$MSGAD  001114          943#  3783*  3786
$MSGLG  001116          944#  3788*
$MSGTY  001100          937#  1567*  3781  3789*  3801  3804*
$MSWR   020004         4087   4202#
$MTYP1  001131          958#
$MTYP2  001135          966#
$MTYP3  001141          969#
$MTYP4  001145          972#
$NULL   001054          918#  3873  3919
$NWTST= 000001         1638#  1655#  1671#  1715#  1737#  1752#  1768#  1798#  1842#  1888#  1915#  1933#  1952#
                       1966#  1993#  2016#  2032#  2048#  2079#  2127#  2177#  2207#  2223#  2242#  2291#  2326#
                       2369#  2408#  2440#  2473#  2506#  2544#  2581#  2619#  2649#  2678#  2706#  2734#  2766#
                       2800#  2833#  2871#  2902#  2934#  2969#  2996#  3029#  3054#  3082#  3105#  3140#  3193#
                       3233#  3264#  3298#  3344#
$OCNT   017012         3951*  3980*  3993#
$OMODF  017014         3946*  3950*  3955  3958*  3969*  3995#
$OVER   015752         3731   3749   3757#
$PASS   001106          940#  1608*  1678  1705  1973  2000  2068  2114  2163  2194  2276  2298  2333
```

```
                        2376    2840    3003    3200    3240    3271    3305    3351    3526*   3527*   3535    3548
$PASTM  000506           881#
$POWER  015604          3707    3712#
$PWRDN  015436          1585    3674#   3701
$PWRMG  015572          3707#
$PWRUP  015510          3684    3692#
$QUES   001074           928#   3919    4133    4184    4200
$RDCHR  017524          4146#   4249
$RDDEC= ****** U        4251
$RDLIN  017654          4176#   4250
$RDOCT= ****** U        4251
$RDSZ = 000010          4169#
$RTNAD  015034          3547#
$R2A  = ****** U        4251
$SAVRE= ****** U        4251
$SAVR6  015602          3683*   3693    3703*   3704*   3711#
$SCOPE  015614          1579    3728#
$SETUP= 000137           840#   1578    1579    1581    1583    1585    1587    1588    1590    1613    3525    3729    4070
                        4206
$STUP = 177777           840#
$SVLAD  015716          3739    3745    3751#
$SVPC = 000500           856#    861
$SWR  = 161000           705     706#    821#    927     928    1588    1590    1591    1642    1659    1675    1719    1741
                        1756    1772    1802    1846    1892    1919    1937    1956    1970    1997    2020    2036    2052
                        2083    2131    2181    2211    2227    2246    2295    2330    2373    2412    2444    2477    2510
                        2548    2585    2623    2653    2682    2710    2738    2770    2804    2837    2875    2906    2938
                        2973    3000    3033    3058    3086    3109    3144    3197    3237    3268    3302    3348    3520
                        3526    3540    3546    3548    3722    3723    3724    3725    3730    3742    3744    3745    3746
                        3751    3754    3757    3760
$SWREG  001122           948#   1611
$SWRMK= 000000          3725
$TESTN  001104           939#   1568*   1642*   1659*   1675*   1719*   1741*   1756*   1772*   1802*   1846*   1892*   1919*
                        1937*   1956*   1970*   1997*   2020*   2036*   2052*   2083*   2131*   2181*   2211*   2227*   2246*
                        2295*   2330*   2373*   2412*   2444*   2477*   2510*   2548*   2585*   2623*   2653*   2682*   2710*
                        2738*   2770*   2804*   2837*   2875*   2906*   2938*   2973*   3000*   3033*   3058*   3086*   3109*
                        3144*   3197*   3237*   3268*   3302*   3348*   3752*   5076    5078    5080    5082    5084    5086
                        5088    5090    5092    5095    5097    5099    5101    5103    5105    5107
$TKB    001046           915#   3892    3899    3919    4068    4079    4096    4150    4156
$TKS    001044           914#   3890    3897    3919    4068    4077    4093    4117*   4148    4154
$TMP0   001060           922#   3356*   3501
$TMP1   001062           923#   3357*   3502
$TMP2   001064           924#   3358*   3503
$TMP3   001066           925#   3359*   3504
$TMP4   001070           926#   3360*   3505
$TN   = 000071           705#    820#   1638    1642#   1655    1659#   1671    1675#   1715    1719#   1737    1741#   1752
                        1756#   1768    1772#   1798    1802#   1842    1846#   1888    1892#   1915    1919#   1933    1937#
                        1952    1956#   1966    1970#   1974    1993    1997#   2001    2016    2020#   2032    2036#   2048
                        2052#   2069    2079    2083#   2127    2131#   2177    2181#   2207    2211#   2223    2227#   2242
                        2246#   2291    2295#   2299    2326    2330#   2334    2369    2373#   2408    2412#   2440    2444#
                        2473    2477#   2506    2510#   2544    2548#   2581    2585#   2619    2623#   2649    2653#   2678
                        2682#   2706    2710#   2734    2738#   2766    2770#   2800    2804#   2833    2837#   2841    2871
                        2875#   2902    2906#   2934    2938#   2969    2973#   2996    3000#   3004    3029    3033#   3054
                        3058#   3082    3086#   3105    3109#   3140    3144#   3193    3197#   3201    3233    3237#   3241
                        3264    3268#   3272    3298    3302#   3306    3344    3348#
$TPB    001052           917#   3908*   3919
$TPFLG  001057           921#   3840    3919
```

```
$TPS     001050            916#   3906   3919
$TRAP    020026           1583    4217#
$TRAP2   020050           4228#   4239
$TRP   = 000012           4232#   4241#  4242#  4243#  4244#  4245#  4246   4247#  4248   4249#  4250#  4251#
$TRPAD   020062           4222    4239#
$TSTM    000504            880#
$TSTNM   001002            894#   3525*  3586   3721   3751*  3752   3757   3760
$TTYIN   017762           4177    4178   4195   4199#
$TYPBN=  ****** U         4245
$TYPDS   017016           4008#   4244
$TYPE    016234           3794    3840#  4232   4240
$TYPEC   016446           3870    3877   3884.  3889#  4119
$TYPEX   016566           3912    3914   3917#
$TYPOC   016614           3949#   4241
$TYPON   016630           3948    3951#  4243
$TYPOS   016570           3944#   4242
$UNIT    001112            942#
$UNITM   000510            882#
$USWR    001124            949#
$VECT1   C01150            974#
$VECT2   001152            975#
$XOFF  = 000023           3894    3919
$XON   = 000021           3901    3919   4161
$XTSTR   015626           3733#
$$GET4=  000000           3540#
$OFILL   017013           3945*   3949*  3959   3994#
$40CAT=  ****** U         3730
.      - 031450            825#    831    832#   836#   842#   846#   851#   856    857#   859#   861#   867    868#
                           870#    872#   891#   931    997#   1016#  1024#  1267#  1576   1590   1591   3548   3552
                          3667#   3686    3710   3760   3816#  3919   4062#  4068   4199#  4200   4206   4316#  5075#
                          5109#   5110#
.$ERRT   001160            996#
.$X      000500            867#    872
```

```
APTSKP   691#  197¹  1998  2065  2296  2331  2838  3001  3198  3238  3269  3303
COMMEN   817#
DOTHIS   693#  1720  1938  2228  2379  2739  3145
ENDCOM   817#
ERROR    711#  1733  1748  1763  1782  1794  1810  1815  1822  1828  1837  1854  1859  1866  1873
         1882  1901  1909  1929  1947  1962  1984  1990  2012  2027  2043  2056  2060  2064  2076
         2091  2096  2103  2109  2122  2139  2144  2151  2158  2171  2189  2201  2219  2238  2254
         2259  2266  2272  2284  2305  2314  2322  2361  2395  2401  2425  2435  2457  2468  2495
         2528  2535  2563  2576  2601  2614  2643  2673  2696  2701  2727  2753  2759  2783  2793
         2817  2828  2860  2888  2894  2918  2929  2951  2964  2990  3024  3044  3049  3071  3077
         3097  3102  3124  3135  3163  3175  3184  3215  3225  3259  3291  3334  3417  3440  3473
         3481  3489  3497  3564
ESCAPE   817#
GETPRI   817#
GETSWR   817#  1613
MULT     817#
NEWTST   817#  1638  1655  1671  1715  1737  1752  1768  1798  1842  1888  1915  1933  1952  1966
         1993  2016  2032  2048  2079  2127  2177  2207  2223  2242  2291  2326  2369  2408  2440
         2473  2506  2544  2581  2619  2649  2678  2706  2734  2766  2800  2833  2871  2902  2934
         2969  2996  3029  3054  3082  3105  3140  3193  3233  3264  3298  3344
POP      696#   817#  4049
PUSH     696#   817#  4008
REPORT   817#  1648  1665  1689  1697  1776  1786
SCOPE    712#  1641  1658  1674  1718  1740  1755  1771  1801  1845  1891  1918  1936  1955  1969
         1996  2019  2035  2051  2082  2130  2180  2210  2226  2245  2294  2329  2372  2411  2443
         2476  2509  2547  2584  2622  2652  2681  2709  2737  2769  2803  2836  2874  2905  2937
         2972  2999  3032  3057  3085  3108  3143  3196  3236  3267  3301  3347  3524
SETPRI   817#
SETRA    4232#  4241  4242  4243  4244  4246  4248  4249  4250
SETUP    696#   817#  1570
SKIP     817#
SLASH    817#
SPACE    817#
STARS    696#   817#  826   854   864   866   873   887   931   934   1638  1640  1655  1657  1671
         1673  1715  1717  1737  1739  1752  1754  1768  1770  1798  1800  1842  1844  1888  1890
         1915  1917  1933  1935  1952  1954  1966  1968  1993  1995  2016  2018  2032  2034  2048
         2050  2079  2081  2127  2129  2177  2179  2207  2209  2223  2225  2242  2244  2291  2293
         2326  2328  2369  2371  2408  2410  2440  2442  2473  2475  2506  2508  2544  2546  2581
         2583  2619  2621  2649  2651  2678  2680  2706  2708  2734  2736  2766  2768  2800  2802
         2833  2835  2871  2873  2902  2904  2934  2936  2969  2971  2996  2998  3029  3031  3054
         3056  3082  3084  3105  3107  3140  3142  3193  3195  3233  3235  3264  3266  3298  3300
         3344  3346  3517  3571  3581  3623  3671  3673  3689  3691  3718  3762  3764  3825  3921
         3998  4067  4070  4138  4169  4211  4253  4258  4320  4326
SWRSU    817#  1592#
TRMTRP   4232#
TYPBIN   817#
TYPDEC   817#  3535
TYPNAM   817#
TYPNUM   817#
TYPOCS   817#
TYPOCT   817#  3635  3659  4088
TYPTXT   817#
$SKIP    692#  1974  2001  2069  2299  2334  2841  3004  3201  3241  3272  3306
$$CMRE   885#
$$CMTM   885#   922   923   924   925   926
$$ESCA   817#
```

```
$$NEWT    817#   1638   1655   1671   1715   1737   1752   1768   1798   1842   1888   1915   1933   1952   1966
          1993   2016   2032   2048   2079   2127   2177   2207   2223   2242   2291   2326   2369   2408   2440
          2473   2506   2544   2581   2619   2649   2678   2706   2734   2766   2800   2833   2871   2902   2934
          2969   2996   3029   3054   3082   3105   3140   3193   3233   3264   3298   3344
$$SET     4232#  4241   4242   4243   4244   4246   4248   4249   4250
$$SETM    1608#
$$SKIP     817#
.EQUAT     696#   707
.HEADE     696#
.SETUP     696#   840
.$ACT1     696#   852
.$APTB     696#   932#
.$APTH     696#   862
.$CMTA     696#   885
.$EOP      696#  3515
.$ERRT     696#  3621
.$READ     696#  4065
.$SCOP     696#  3716
.$TRAP     696#  4209
.$TYPD     696#  3996
.$TYPE     696#  3823
.$TYPO     696#  3919


. ABS.  031450     000


 ERRORS DETECTED:  0

 DSKM:CJKDFA,DSKZ:CJKDFA.LST/SOL/NL:TOC/CRF=CJKDFA.P11
 RUN-TIME: 75 47 4 SECONDS
 RUN-TIME RATIO: 184/127=1.4
 CORE USED:  29K  (57 PAGES)
```