Micro Fiche Scan

Name of device(s) tested:
    KDF11-B
Test description:
    11/23B SLU/RTC REPAIR
MAINDEC Number or Package Identifier (after SEP 1977):
    CJKDIB0
Fiche Document Part Number:
    AH-T054B-MC
Fiche preparation date unknown, using copyright year:
    1986
Image resolution:
    1-bit black&white, compressed for minimal file size

```
.REM ¢

        IDENTIFICATION
      ----------------- --

    PRODUCT CODE:  AC-T052B-MC

    PRODUCT NAME:  CJKDIB0 11/23B SLU LTC REPAIR

    PRODUCT DATE:  MARCH-86

    MAINTAINER:    MSD DIAGNOSTIC ENGINEERING


THE INFORMATION IS THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO
RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS
DOCUMENT.



NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF
SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS
AFFILIATED COMPANIES.

COPYRIGHT (C):  1982, 1986 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL         PDP     UNIBUS          MASSBUS
DEC             DECUS   DECTAPE         DEC/X11
```

C1


```
                                HISTORY
                                ---------

REVISION        A       FIRST RELEASE OF DIAGNOSTIC
REVISION        B       DIAGNOSTIC SOURCE WAS RECREATED
```

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 M8189 MODULE. ITS MAIN PURPOSE IS TO PROVIDE SCOPE LOOPING FOR REPAIR PERSONNEL.
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

- 11/23B CPU MODULE
- MINIMUM 8K OF MEMORY
- TURN AROUND JUMPER INSTALLED ON SLU2

B. SOFTWARE ENVIRONMENTS

- XXDP STAND-ALONE
- XXDP CHAIN MODE
- APT

1.3 RELATED DOCUMENTS AND STANDARDS

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

- 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

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):

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 (BOT ' _US)
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 . 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 INPUTING 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, TBUF, OR LKS

WHERE "XXXXXX" IS AN OCTAL NUMBER.

THIS ERROR PRINTOUT OCCURS PROVIDED THE ERROR THAT EXISTS WOULD NOT HINDER THE TYPEOUT. IN CASES WHERE IT IS NOT POSSIBLE TO PRINT AN ERROR MESSAGE (I.E. FATAL CONSOLE TRANSMITTER FAILURES), A HALT OCCURS. (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 PRINTOUTS. AFTER CONTINUING FROM THE ERROR HALT THE OLD SWR CONTENTS IS DISPLAYED AND THE NEW CONTENTS CAN BE ENTERED. IF ERROR HALTS ARE DISABLED, THE CONTROL-G RESPONSE OCCURS IMMEDIATELY FOLLOWING THE TYPEOUT.

3.2 ERROR HALTS

THERE ARE 4 ERRORS IN THIS PROGRAM THAT CAUSE A HALT. THESE ERRORS ARE IN TESTS 1, 2, AND 3.

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.

4.0 PERFORMANCE AND PROGRESS REPORTS

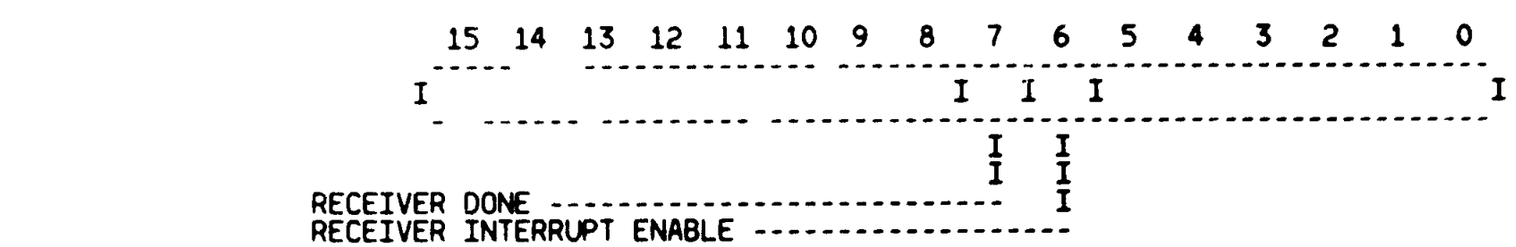
THE ONLY REPORT FROM THIS PROGRAM OTHER THAN ERROR REPORTS IS THE END PASS MESSAGE. IT IS IN THE FORM:

END PASS XXXXXX

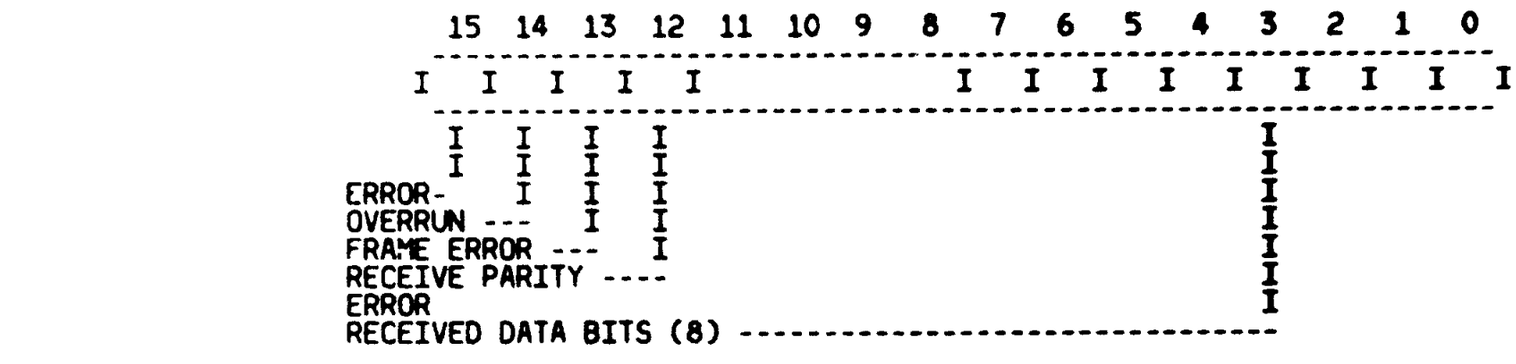
WHERE XXXXXX IS THE DECIMAL NUMBER OF END OF PASSES COMPLETED.

5.0 DEVICE INFORMATION TABLES

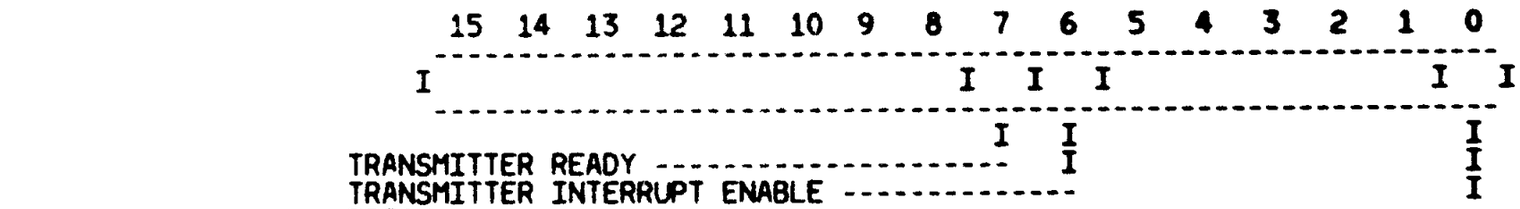
SLU1 RCSR



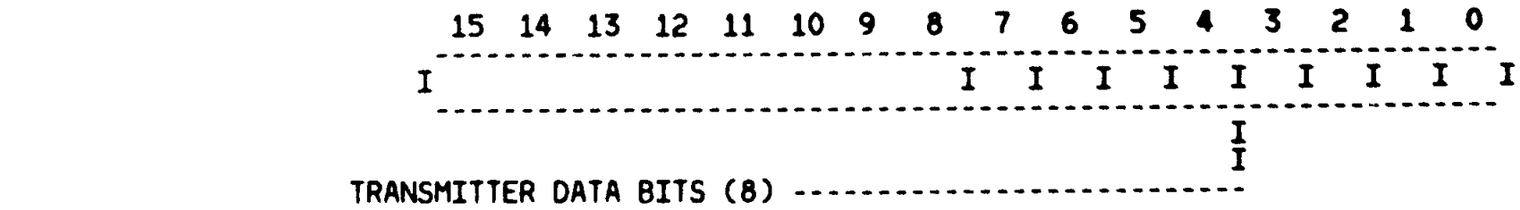
SLU1 RBUF



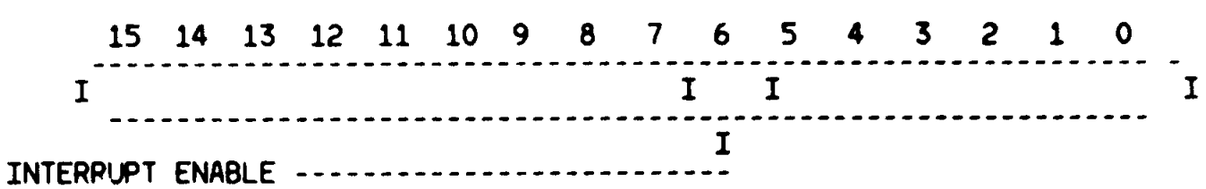
SLU1 XCSR



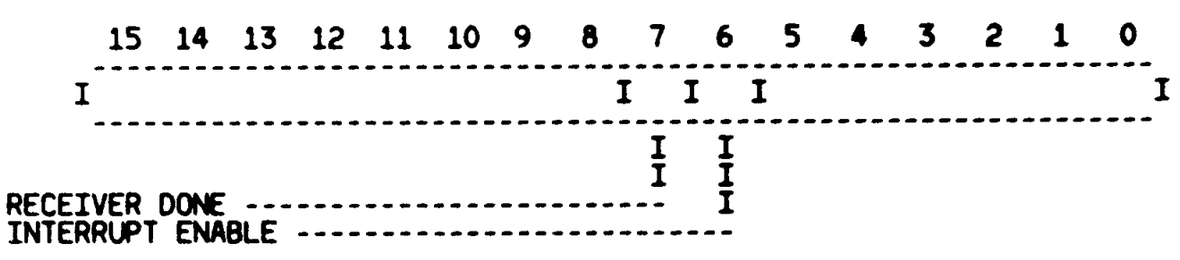
SLU1 XBUF



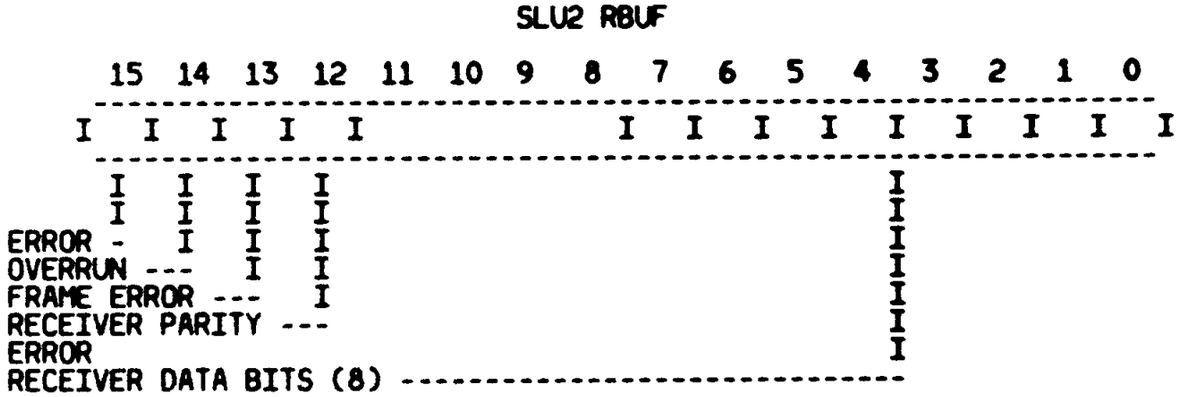
LTC CSR



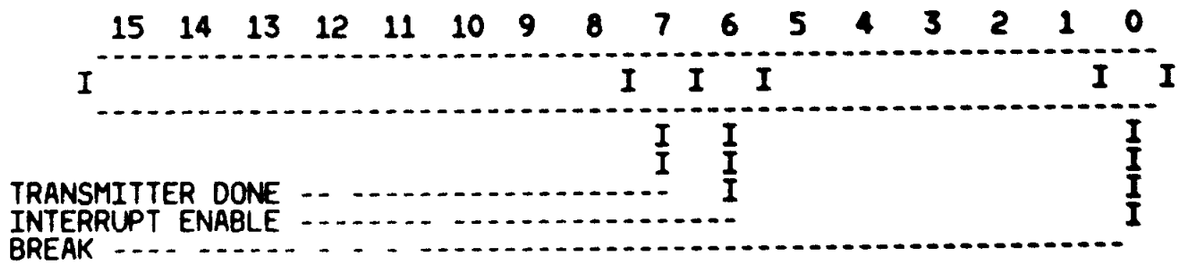
SLU2 RCSR

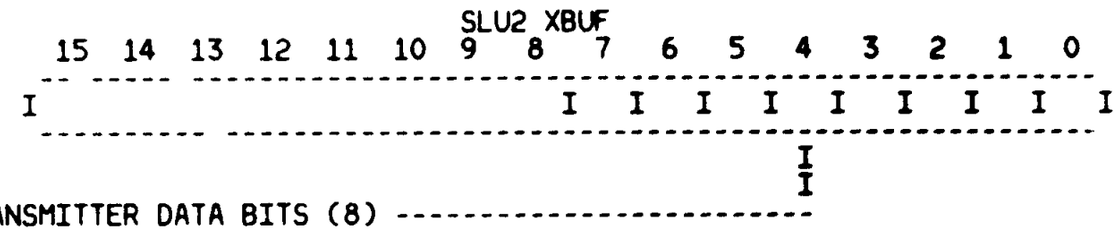


SLU2 RBUF



SLU2 XCSR





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 TIME OUT TRAP.

TEST2 TEST ABILITY TO ACCESS SLU1 TRANSMITTER BUFFER. HALTS IF ACCESS CAUSES TIME-OUT 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 TIME-OUT

```
                        TRAP.

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


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".

TEST7       TEST THAT SLU1 RECEIVER INTERRUPT ENABLE (BIT6) CAN  BE  SET
            AND RESET.  SAME TEST AS FOR TRANSMITTER INTERRUPT ENABLE.

TEST10      TEST THAT SLU1 BIT6 OF RECEIVER CSR CAN BE SET AND RESET.
            PERFORM BIS (BIT SET) AND BIC (BIT CLEAR) ON THE RCSR.
            AN ERROR HERE CAUSES AN ERROR PRINT OUT TO TERMINAL.

TEST11      SAME AS TEST1 BUT DONE ON SLU2 AND ERROR DOES NOT CAUSE HALT.

TEST12      SAME AS TEST2 BUT FOR SLU2 AND ERROR DOES NOT CAUSE HALT.

TEST13      SAME AS TEST3 BUT FOR SLU2 AND ERROR DOES NOT CAUSE HALT.

TEST14-16 SAME AS TEST4-6 BUT FOR SLU2.

TEST17      TEST SLU2 BREAK BIT (BIT0) CAN BE SET, CLEARED, AND RESET.
            PERFORM BIS, BIC AND BIT ON SLU2 TCSR.

TEST20-21 SAME AS TEST 7-10 BUT FOR SLU2.

TEST22      TEST THAT SLU2 RCVR DONE BIT (BIT7) SET AND CLEAR PROPERLY.
            WITH A LOOP BACK CONNECTOR PLACED ON SLU2, THE CODE WILL LOAD
            THE TRANSMIT BUFFER AND EXPECT TO RECEIVE DATA IN THE RECEIVER
            BUFFER.  A CORRECT RESULT WILL CAUSE THE RCSR DONE BIT TO BE SET.
            THE TEST WILL ALSO PERFORM A RESET INSTRUCTION AND TEST THAT THE
            RCSR DONE BIT 7 IS CLEARED.

TEST23      TEST SLU2 THAT READING THE RBUF CLEARS THE RECEIVER DONE BIT 7.

TEST24      TEST ABILITY TO ACCESS LINE CLOCK  STATUS  REGISTER.  ERROR
            REPORTED IF ACCESS CAUSES TIME-OUT TRAP.

TEST25      TEST THAT LINE CLOCK INTERRUPT ENABLE  (BIT6)  CAN  BE  SET,
            CLEARED, AND "RESET".  THIS TEST PERFORMS THE BIT, BIC AND BIS
            INSTRUCTIONS ON BIT 6 OF THE LKS.

TEST26      UNIQUE INTERNAL ADDRESS TEST.  THIS TEST WRITES A  BIT  INTO
            ONE  OF  THE  DEVICE REGISTERS AND THEN VERIFIES THAT BIT IS
            NOT SET IN  OTHER REGISTERS.  THE TEST IS REPEATED FOR OTHER
            REGISTERS.

TEST27      TEST THAT SLU1 TRANSMITTER  INTERRUPTS  ONLY  WHEN  ENABLED.
            THIS  TEST  CHECKS THAT THE TRANSMITTER ONLY INTERRUPTS WHEN
            ITS INTERRUPT ENABLE BIT IS SET.
```

```
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
```

TEST30 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.

TEST31 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".

TEST32 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.

TEST33-36 SAME AS TESTS 27-32 EXCEPT FOR SLU2 INSTEAD OF SLU1.

TEST37-41 SAME AS TESTS 27-31 EXCEPT FOR SLU2 RECEIVER INSTEAD OF SLU1 TRANSMITTER.

TEST42 TEST THAT SLU2 RECEIVER INTERRUPT CLEARS BY READING RBUF. THIS TEST SETS THE PRIORITY AT 7, SETS RCVR INTERRUPT ENABLE AND FILL THE TBUF. WHEN "DONE" SETS, RBUF WILL ATTEMPT TO CLEAR THE INTERRUPT BY READING THE BUFFER.

TEST43 TEST SLU2 THAT RESET CLEARS RECEIVE INTERRUPT. THIS TEST SETS THE PRIORITY TO 7, SETS RCVR INT ENABLE AND THEN WAITS FOR RCV DONE. WHEN "DONE" SETS, A RESET INSTRUCTION IS EXECUTED. AN ERROR IS REPORTED IF THE RECEIVER INTERRUPT IS NOT CLEARED.

TEST44 TEST SLU2 THAT "OVERRUN & ERROR" BITS CAN BE SET. THIS TEST SENDS THREE CHARACTERS TO THE RCVR ONE AT A TIME. BIT14 OF THE RBUF IS CHECKED FOR OVERRUN. IF IT IS NOT SET AN ERROR IS SET. BIT15 (ERROR BIT) IS THEN CHECKED TO VERIFY THAT AN ERROR HAS OCCURRED.

TEST45 TEST THAT BREAK TRANSMITS ALL ZEROS. PUT ALL ONES INTO RECEIVER BUFFER THEN ISSUE A BREAK THE RECEIVER BUFFER SHOULD CONTAIN ZEROS.

TEST46 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.

TEST47 TEST SLU2 DATA PATH USING WRAP CABLE CONNECTOR. THIS TEST TRANSMITS A BINARY COUNT PATTERN ONE BIT AT A TIME. THE TEST THEN VERIFIES THE DATA AGAINST WHAT IT EXPECTED TO RECEIVE. ERRORS ARE REPORTED TO THE TERMINAL.

TEST50 TEST LINE TIME CLOCK INTERRUPTS PROPERLY.

TEST51 TEST LINE TIME CLOCK FOR DOUBLE INTERRUPTS.

TEST52 TEST THAT LINE TIME CLOCK INTERRUPT CLEARS WITH RESET.

TEST53 BLAST TEST. THIS TEST RUNS ALL SELECTED DEVICES SIMULTANE-
OUSLY 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 TRANS-
FERRED WAS CORRECT.

NOTE: IF RUNNING UNDER THE APT ENVIRONMENT 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.

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

```
                        .TITLE  CJKDIB0 11/238 SLU LTC REPR DIAG
                        ;*COPYRIGHT (C) SEPT-81, MARCH-86
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.
                        ;*
                        .ENABLE ABS,AMA

104401                  TYPE=104401
104402                  TYPOC=104402
104403                  TYPOS=104403
104404                  TYPON=104404
104405                  TYPDS=104405
104406                  GTSWR=104406
104407                  CKSWR=104407
```

```
                        RDCHR=104410
                        RDLIN=104411
                        $XON=000021
                        $XOFF=000023
                        $TN=1
                        $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
                        $SWRMRK=000000
                        $TN=1

                        .SBTTL  BASIC DEFINITIONS

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

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

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

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

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
```

C2


```
                        002000          SW10=   2000
                        001000          SW09=   1000
                        000400          SW08=   400
                        000200          SW07=   200
                        000100          SW06=   100
                        000040          SW05=   40
                        000020          SW04=   20
                        000010          SW03=   10
                        000004          SW02=   4
                        000002          SW01=   2
                        000001          SW00=   1
                                        .EQUIV  SW09,SW9
                                        .EQUIV  SW08,SW8
                                        .EQUIV  SW07,SW7
                                        .EQUIV  SW06,SW6
                                        .EQUIV  SW05,SW5
                                        .EQUIV  SW04,SW4
                                        .EQUIV  SW03,SW3
                                        .EQUIV  SW02,SW2
                                        .EQUIV  SW01,SW1
                                        .EQUIV  SW00,SW0

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

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

D2


```
                                ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                                ;;POWER FAIL
                                ;;EMULATOR TRAP (EMT) **ERROR**
                                ;;"TRAP" TRAP
                                ;;TTY KEYBOARD VECTOR
                                ;;TTY PRINTER VECTOR
                                ;;PROGRAM INTERRUPT REQUEST VECTOR
                                ;THIS IS THE COMMAND FOR A TRAP
                                ;   THROUGH 14 (BPT TRAP)
                                .ASECT
                                .=0
                                ;;*****************************************************************
                                ;*ALL UNUSED LOCATIONS FROM 4-776 CONTAIN A ".+2,BPT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS & INTERRUPPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                                .=14                    ;THE BPT TRAP VECTOR POINTS TO THE
000014  012370                  .WORD   CATCH           ;   ILLEGAL TRAP HANDLER "CATCH"
000016  000340                  .WORD   340
                                .=      42
000042  000000                  .WORD   0
```

E2


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

                                .=200
000200  000137  003016          JMP     START           ;DO INTERFACE TEST
000204  000137  0'5406          JMP     ECHO            ;DO ECHO TEST
000210  000137  015664          JMP     OUTTST          ;DO OUTPUT TEST TO TERMINAL

                                .=      500
                        .SBTTL  ACT11 HOOKS

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

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

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
```

F2


```
882  001002  000          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
883  001003  000          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
884  001004  000000       $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
885  001006  000000       $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
886  001010  000000       $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
887  001012  000000       $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
888  001014  000          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
889  001015  001          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
890  001016  000000       $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
891  001020  000000       $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
892  001022  000000       $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
893  001024  000000       $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
894  001026  000000       $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
895  001030  000000               .WORD   0               ;;RESERVED--NOT TO BE USED
896  001032  000000               .WORD   0
897  001034  000          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
898  001035  000          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
899  001036  000000               .WORD   0
900  001040  177570       SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
901  001042  177570       DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
902  001044  177560       $TKS:   177560                  ;;TTY KBD STATUS
903  001046  177562       $TKB:   177562                  ;;TTY KBD BUFFER
904  001050  177564       $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
905  001052  177566       $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
906  001054  000          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
907  001055  002          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
908  001056  012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
909  001057  000          $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
910  001060  000000       $TMP0:  .WORD   0               ;;USER DEFINED
911  001062  000000       $TMP1:  .WORD   0               ;;USER DEFINED
912  001064  000000       $TMP2:  .WORD   0               ;;USER DEFINED
913  001066  000000       $TMP3:  .WORD   0               ;;USER DEFINED
914  001070  000000       $TMP4:  .WORD   0               ;;USER DEFINED
915  001072  000000       $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
916  001074     077       $QUES:  .ASCII  /?/             ;;QUESTION MARK
917  001075     015       $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
918  001076  000012       $LF:    .ASCIZ  <12>            ;;LINE FEED
919                       ;;*****************************************************************
920                       .SBTTL  APT MAILBOX-ETABLE
921
922                       ;;*****************************************************************
923                       .EVEN
924  001100               $MAIL:                          ;;APT MAILBOX
925  001100  000000       $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
926  001102  000000       $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
927  001104  000000       $TESTN: .WORD   ATESTN          ;;TEST NUMBER
928  001106  000000       $PASS:  .WORD   APASS           ;;PASS COUNT
929  001110  000000       $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
930  001112  000000       $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
931  001114  000000       $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
932  001116  000000       $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
933  001120               $ETABLE:                        ;;APT ENVIRONMENT TABLE
934  001120     000       $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
935  001121     000       $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
936  001122  000000       $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
937  001124  000400       $USWR:  .WORD   AUSWR           ;;USER SWITCHES
```

```
   938  001126  000000                  $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
   939                                  ;*                      BITS 15 11=CPU TYPE
   940                                  ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
   941                                  ;*                              11/70=06,PDQ=07,Q=10
   942                                  ;*                      BIT 10=REAL TIME CLOCK
   943                                  ;*                      BIT  9=FLOATING POINT PROCESSOR
   944                                  ;*                      BIT  8=MEMORY MANAGEMENT
   945  001130     000                  $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
   946  001131     000                  $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
   947                                  ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
   948                                  ;*                              900 NSEC CORE=001
   949                                  ;*                              300 NSEC BIPOLAR=002
   950                                  ;*                              500 NSEC MOS=003
   951  001132  000000                  $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
   952                                  ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
   953  001134     000                  $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
   954  001135     000                  $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
   955  001136  000000                  $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
   956  001140     000                  $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
   957  001141     000                  $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
   958  001142  000000                  $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
   959  001144     000                  $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
   960  001145     000                  $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
   961  001146  000000                  $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
   962  001150  000300                  $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
   963  001152  000000                  $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
   964  001154  176500                  $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
   965  001156  000000                  $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
   966  001160                          $ETEND:
   967
   968                                  .SBTTL  ERROR POINTER TABLE
   969
   970                                  ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
   971                                  ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
   972                                  ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
   973                                  ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
   974                                  ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
   975
   976                                  ;*      EM              ;;POINTS TO THE ERROR MESSAGE
   977                                  ;*      DH              ;;POINTS TO THE DATA HEADER
   978                                  ;*      DT              ;;POINTS TO THE DATA
   979                                  ;*      DF              ;;POINTS TO THE DATA FORMAT
   980
   981
   982  001160                          $ERRTB:
   983
   984  001160                          .$ERRTB:
   985  001160  000020                          .BLKW   20              ;THIS BLOCK OF 16 LOCATIONS IS HERE TO PACIFY SYSMAC
   986
   987
   988
   989  001220  016010                          EM5                     ;SLU1 TCSR DONE NOT SET WITH RESET
   990  001222  024731                          DH5                     ;"TEST# ERR PC  TCSR"
   991  001224  025512                          DT5                     ;$TESTN,$ERRPC,CTCSR
   992  001226  000000                          0
   993
```

H2


```
  994  001230  016052                EM6                     ;SLU1 RCSR DID NOT RETURN SSYNC
  995  001232  024756                DH6                     ;"TEST# ERR PC  RCSR"
  996  001234  025522                DT6                     ;$TESTN,$ERRPC,CRCSR
  997  001236  000000                0
  998
  999  001240  016111                EM7                     ;SLU1 RBUF DID NOT RETURN SSYNC
 1000  001242  025003                DH7                     ;"TEST# ERR PC  RBUF"
 1001  001244  025532                DT7                     ;$TESTN,$ERRPC,CRBUF
 1002  001246  000000                0
 1003
 1004  001250  000004                .BLKW   4               ;MORE PACIFICATION
 1005
 1006
 1007  001260  016150                EM11                    ;"CAN NOT SET BIT2 OF SLU1 TCSR"
 1008  001262  024731                DH5                     ;"TEST# ERR PC  TCSR"
 1009  001264  025512                DT5                     ;$TESTN,$ERRPC,CTCSR
 1010  001266  000000                0
 1011
 1012  001270  000004                .BLKW   4               ;STILL MORE PACIFICATION
 1013
 1014
 1015  001300  016206                EM13                    ;"RESET DID NOT CLEAR BIT2 OF SLU1 TCSR"
 1016  001302  024731                DH5                     ;"TEST# ERR PC  TCSR"
 1017  001304  025512                DT5                     ;$TESTN,$ERRPC,CTCSR
 1018  001306  000000                0
 1019
 1020  001310  016254                EM14                    ;"BIT6 OF SLU1 TCSR NOT CLEAR AFTER RESET2
 1021  001312  024731                DH5                     ;"TEST# ERR PC  TCSR"
 1022  001314  025512                DT5                     ;$TESTN,$ERRPC,CTCSR
 1023  001316  000000                0
 1024
 1025  001320  016324                EM15                    ;"SLU1 XMIT INTERRUPTED WITH PRIORITY 7"
 1026  001322  024731                DH5                     ;"TEST# ERR PC  TCSR"
 1027  001324  025512                DT5                     ;$TESTN,$ERRPC,CTCSR
 1028  001326  000000                0
 1029
 1030  001330  016372                EM16                    ;"CAN NOT SET BIT6 OF SLU1 TCSR"
 1031  001332  024731                DH5                     ;"TEST# ERR PC  TCSR"
 1032  001334  025512                DT5                     ;$TESTN,$ERRPC,CTCSR
 1033  001336  000000                0
 1034
 1035  001340  016430                EM17                    ;"CAN NOT CLEAR BIT6 OF SLU1 TCSR"
 1036  001342  024731                DH5                     ;"TEST# ERR PC  TCSR"
 1037  001344  025512                DT5                     ;$TESTN,$ERRPC,CTCSR
 1038  001346  000000                0
 1039
 1040  001350  016470                EM20                    ;"RESET DID NOT CLEAR BIT6 OF SLU1 TCSR"
 1041  001352  024731                DH5                     ;"TEST# ERR PC  TCSR"
 1042  001354  025512                DT5                     ;$TESTN,$ERRPC,CTCSR
 1043  001356  000000                0
 1044
 1045  001360  016536                EM21                    ;"BIT6 OF SLU1 RCSR NOT CLEAR AFTER RESET"
 1046  001362  024756                DH6                     ;"TEST# ERR PC  RCSR"
 1047  001364  025522                DT6                     ;$TESTN,$ERRPC,CRCSR
 1048  001366  000000                0
 1049
```

```
1050  001370  016606                  EM22            ;"SLU1 RCVR INTERRUPT WITH PRIORITY 7"
1051  001372  024756                  DH6             ;"TEST# ERR PC  RCSR"
1052  001374  025522                  DT6             ;$TESTN,$ERRPC,CRCSR
1053  001376  000000                  0
1054
1055  001400  016652                  EM23            ;"CAN NOT SET BIT6 OF SLU1 RCSR"
1056  001402  024756                  DH6             ;"TEST# ERR PC  RCSR"
1057  001404  025522                  DT6             ;$TESTN,$ERRPC,CRCSR
1058  001406  000000                  0
1059
1060  001410  016710                  EM24            ;"CAN NOT CLEAR BIT6 OF SLU1 RCSR"
1061  001412  024756                  DH6             ;"TEST# ERR PC  RCSR"
1062  001414  025522                  DT6             ;$TESTN,$ERRPC,CRCSR
1063  001416  000000                  0
1064
1065  001420  016750                  EM25            ;"CAN NOT CLEAR BIT6 OF SLU1 RCSR WITH RESET2
1066  001422  024756                  DH6             ;"TEST# ERR PC  RCSR"
1067  001424  025522                  DT6             ;$TESTN,$ERRPC,CRCSR
1068  001426  000000                  0
1069
1070  001430  017023                  EM26            ;"SLU1 RECEIVER DONE NEVER SET"
1071  001432  024756                  DH6             ;"TEST# ERR PC  RCSR"
1072  001434  025522                  DT6             ;$TESTN,$ERRPC,CRCSR
1073  001436  000000                  0
1074
1075  001440  017060                  EM27            ;"RESET DID NOT CLEAR SLU1 RCVR DONE"
1076  001442  024756                  DH6             ;"TEST# ERR PC  RCSR"
1077  001444  025522                  DT6             ;$TESTN,$ERRPC,CRCSR
1078  001446  000000                  0
1079
1080  001450  017123                  EM30            ;"READING SLU1 RBUF DID NOT CLEAR RCVR DONE"
1081  001452  024756                  DH6             ;"TEST# ERR PC  RCSR"
1082  001454  025522                  DT6             ;$TESTN,$ERRPC,CRCSR
1083  001456  000000                  0
1084
1085  001460  017175                  EM31            ;SLU2 TCSR DID NOT RETURN SSYNC
1086  001462  024731                  DH5             ;"TEST# ERR PC  TCSR"
1087  001464  025542                  DT31            ;$TESTN,$ERRPC,TCSR
1088  001466  000000                  0
1089
1090  001470  017234                  EM32            ;SLU2 TBUF DID NOT RETURN SSYNC
1091  001472  025030                  DH32            ;"TEST# ERR PC  TBUF"
1092  001474  025552                  DT32            ;$TESTN,$ERRPC,TBUF
1093  001476  000000                  0
1094
1095  001500  017273                  EM33            ;"SLU2 TCSR DONE NOT CLEARED WITH TBUF FULL"
1096  001502  024731                  DH5             ;"TEST# ERR PC  TCSR"
1097  001504  025542                  DT31            ;$TESTN,$ERRPC,TCSR
1098  001506  000000                  0
1099
1100  001510  017345                  EM34            ;"SLU2 TCSR DONE NOT SET AFTER TRANSMIT"
1101  001512  024731                  DH5             ;"TEST# ERR PC  TCSR"
1102  001514  025542                  DT31            ;$TESTN,$ERRPC,TCSR
1103  001516  000000                  0
1104
1105  001520  017413                  EM35            ;"SLU2 TCSR DONE NOT SET WITH RESET"
```

J2


```
1106  001522  024731          DH5                     ;"TEST#  ERR PC  TCSR"
1107  001524  025542          DT31                    ;$TESTN,ERRPC,TCSR
1108  001526  000000          0
1109
1110  001530  017455          EM36                    ;"SLU2 RCSR DID NOT RETURN SSYNC"
1111  001532  024756          DH6                     ;"TEST#  ERR PC  RCSR"
1112  001534  025562          DT36                    ;$TESTN,ERRPC,RCSR
1113  001536  000000          0
1114
1115  001540  017514          EM37                    ;SLU2 RBUF DID NOT RETURN SSYNC"
1116  001542  025003          DH7                     ;"TEST#  ERR PC  RBUF"
1117  001544  025572          DT37                    ;$TESTN,ERRPC,RBUF
1118  001546  000000          0
1119
1120  001550  017553          EM40                    ;"BIT0 OF SLU2 TCSR NOT CLEAR AFTER RESET"
1121  001552  024731          DH5                     ;"TEST# ERR PC  TCSR
1122  001554  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1123  001556  000000          0
1124
1125  001560  017623          EM41                    ;"CAN NOT SET BIT0 OF SLU2 TCSR"
1126  001562  024731          DH5                     ;"TEST# ERR PC  TCSR
1127  001564  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1128  001566  000000          0
1129
1130  001570  017661          EM42                    ;"CAN NOT CLEAR BIT0 OF SLU2 TCSR"
1131  001572  024731          DH5                     ;"TEST# ERR PC  TCSR
1132  001574  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1133  001576  000000          0
1134
1135  001600  017721          EM43                    ;"RESET DID NOT CLEAR BIT0 OF SLU2 TCSR"
1136  001602  024731          DH5                     ;"TEST# ERR PC  TCSR
1137  001604  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1138  001606  000000          0
1139
1140  001610  017767          EM44                    ;"BIT6 OF SLU2 TCSR NOT CLEAR AFTER RESET2
1141  001612  024731          DH5                     ;"TEST# ERR PC  TCSR
1142  001614  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1143  001616  000000          0
1144
1145  001620  020037          EM45                    ;"SLU2 XMIT INTERRUPTED WITH PRIORITY 7"
1146  001622  024731          DH5                     ;"TEST# ERR PC  TCSR
1147  001624  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1148  001626  000000          0
1149
1150  001630  020105          EM46                    ;"CAN NOT SET BIT6 OF SLU2 TCSR"
1151  001632  024731          DH5                     ;"TEST# ERR PC  TCSR
1152  001634  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1153  001636  000000          0
1154
1155  001640  020143          EM47                    ;"CAN NOT CLEAR BIT6 OF SLU2 TCSR"
1156  001642  024731          DH5                     ;"TEST# ERR PC  TCSR
1157  001644  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1158  001646  000000          0
1159
1160  001650  020203          EM50                    ;"RESET DID NOT CLEAR BIT6 OF SLU2 TCSR"
1161  001652  024731          DH5                     ;"TEST# ERR PC  TCSR
```

<2


```
1162  001654  025542                  DT31                    ;$TESTN,$ERRPC,TCSR
1163  001656  000000                  0
1164
1165  001660  020251                  EM51                    ;"BIT6 OF SLU2 RCSR NOT CLEAR AFTER RESET'
1166  001662  024756                  DH6                     ;"TEST#  ERR PC  RCSR"
1167  001664  025562                  DT36                    ;$TESTN,ERRPC,RCSR
1168  001666  000000                  0
1169
1170  001670  020321                  EM52                    ;"SLU2 RCVR INTERRUPT WITH PRIORITY 7"
1171  001672  024756                  DH6                     ;"TEST#  ERR PC  RCSR"
1172  001674  025562                  DT36                    ;$TESTN,ERRPC,RCSR
1173  001676  000000                  0
1174
1175  001700  020365                  EM53                    ;"CAN NOT SET BIT6 OF SLU2 RCSR"
1176  001702  024756                  DH6                     ;"TEST#  ERR PC  RCSR"
1177  001704  025562                  DT36                    ;$TESTN,ERRPC,RCSR
1178  001706  000000                  0
1179
1180  001710  020423                  EM54                    ;"CAN NOT CLEAR BIT6 OF SLU2 RCSR"
1181  001712  024756                  DH6                     ;"TEST#  ERR PC  RCSR"
1182  001714  025562                  DT36                    ;$TESTN,ERRPC,RCSR
1183  001716  000000                  0
1184
1185  001720  020463                  EM55                    ;"CAN NOT CLEAR BIT6 OF SLU2 RCSR WITH RESET2
1186  001722  024756                  DH6                     ;"TEST#  ERR PC  RCSR"
1187  001724  025562                  DT36                    ;$TESTN,ERRPC,RCSR
1188  001726  000000                  0
1189
1190  001730  020536                  EM56                    ;"SLU2 RECEIVER DONE NEVER SET"
1191  001732  024756                  DH6                     ;"TEST#  ERR PC  RCSR"
1192  001734  025562                  DT36                    ;$TESTN,ERRPC,RCSR
1193  001736  000000                  0
1194
1195  001740  020573                  EM57                    ;"RESET DID NOT CLEAR SLU2 RCVR DONE"
1196  001742  024756                  DH6                     ;"TEST#  ERR PC  RCSR"
1197  001744  025562                  DT36                    ;$TESTN,ERRPC,RCSR
1198  001746  000000                  0
1199
1200  001750  020636                  EM60                    ;"READING SLU2 RBUF DID NOT CLEAR RCVR DONE"
1201  001752  024756                  DH6                     ;"TEST#  ERR PC  RCSR"
1202  001754  025562                  DT36                    ;$TESTN,ERRPC,RCSR
1203  001756  000000                  0
1204
1205  001760  020710                  EM61                    ;LKS DID NOT RETURN SSYNC
1206  001762  025055                  DH61                    ;"TEST#  ERR PC  LKS"
1207  001764  025602                  DT61                    ;$TESTN,ERRPC,LKS
1208  001766  000000                  0
1209
1210  001770  020741                  EM62                    ;"BIT6 OF LKS NOT CLEAR AFTER RESET"
1211  001772  025055                  DH61                    ;"TEST# ERR PC  LKS"
1212  001774  025602                  DT61                    ;$TESTN,$ERRPC,LKS
1213  001776  000000                  0
1214
1215  002000  021003                  EM63                    ;"LKS INTERRUPT WITH PRIORITY 7"
1216  002002  025055                  DH61                    ;"TEST# ERR PC  LKS"
1217  002004  025602                  DT61                    ;$TESTN,$ERRPC,LKS
```

```
1218  002006  000000                    0
1219
1220  002010  021041                    EM64                ;"CAN NOT SET BIT6 OF LKS"
1221  002012  025055                    DH61                ;"TEST# ERR PC  LKS"
1222  002014  025602                    DT61                ;$TESTN,$ERRPC,LKS
1223  002016  000000                    0
1224
1225  002020  021071                    EM65                ;"CAN NOT CLEAR BIT6 OF LKS"
1226  002022  025055                    DH61                ,"TEST# ERR PC  LKS"
1227  002024  025602                    DT61                ;$TESTN,$ERRPC,LKS
1228  002026  000000                    0
1229
1230  002030  021123                    EM66                ;"RESET DID NOT CLEAR BIT6 OF LKS"
1231  002032  025055                    DH61                ;"TEST# ERR PC  LKS"
1232  002034  025602                    DT61                ;$TESTN,$ERRPC,LKS
1233  002036  000000                    0
1234
1235  002040  021163                    EM67                ;"BIT7 OF LKS NOT SET AFTER RESET2
1236  002042  025055                    DH61                ;"TEST# ERR PC  LKS"
1237  002044  025602                    DT61                ;$TESTN,$ERRPC,LKS
1238  002046  000000                    0
1239
1240  002050  021223                    EM70                ;"CAN NOT CLEAR BIT7 OF LKS"
1241  002052  025055                    DH61                ;"TEST# ERR PC  LKS"
1242  002054  025602                    DT61                ;$TESTN,$ERRPC,LKS
1243  002056  000000                    0
1244
1245  002060  021255                    EM71                ;"BIT7 OF LKS DOES NOT SET"
1246  002062  025055                    DH61                ;"TEST# ERR PC  LKS"
1247  002064  025602                    DT61                ;$TESTN,$ERRPC,LKS
1248  002066  000000                    0
1249
1250  002070  021306                    EM72                ;WRITING TO ONE INTERNAL ADDRESS MODIFIED ANOTHER
1251  002072  025101                    DH72                ;"TEST# ERR PC  GOOD  BAD  GDDATA BDDATA
1252  002074  025612                    DT72                ;$TESTN,$ERRPC,$GDADR,$BDADR,$GDDAT,$BDDAT
1253  002076  000000                    0
1254
1255  002100  000004                    .BLKW   4           ;THE LAST IN A LONG LINE OF PACIFICATION
1256
1257
1258  002110  021367                    EM74                ;"SLU1 XMIT INTERRUPTS WHEN DISABLED"
1259  002112  024731                    DH5                 ;"TEST# ERR PC  TCSR"
1260  002114  025512                    DT5                 ;$TESTN,$ERRPC,CTCSR
1261  002116  000000                    0
1262
1263  002120  021432                    EM75                ;"SLU1 XMIT DID NOT INTERRUPT"
1264  002122  024731                    DH5                 ;"TEST# ERR PC  TCSR"
1265  002124  025512                    DT5                 ;$TESTN,$ERRPC,CTCSR
1266  002126  000000                    0
1267
1268  002130  021466                    EM76                ;"SLU1 XMIT INTERRUPT AT PRIORITY 7"
1269  002132  024731                    DH5                 ;"TEST# ERR PC  TCSR"
1270  002134  025512                    DT5                 ;$TESTN,$ERRPC,CTCSR
1271  002136  000000                    0
1272
1273  002140  021530                    EM77                ;"SLU1 XMIT INTERRUPTS WITH ENABLE CLEAR"
```

M2

SEQ 0025


```
1274  002142  024731          DH5                     ;"TEST# ERR PC  TCSR"
1275  002144  025512          DT5                     ;$TESTN,$ERRPC,CTCSR
1276  002146  000000          0
1277
1278  002150  021577          EM100                   ;"SLU1 XMIT DID NOT INTERRUPT"
1279  002152  024731          DH5                     ;"TEST# ERR PC  TCSR"
1280  002154  025512          DT5                     ;$TESTN,$ERRPC,CTCSR
1281  002156  000000          0
1282
1283  002160  021633          EM101                   ;"SLU1 XMIT RE-INTERRUPTED"
1284  002162  024731          DH5                     ;"TEST# ERR PC  TCSR"
1285  002164  025512          DT5                     ;$TESTN,$ERRPC,CTCSR
1286  002166  000000          0
1287
1288  002170  021664          EM102                   ;"LOADING SLU1 TBUF DID NOT CLEAR INTERRUPT"
1289  002172  024731          DH5                     ;"TEST# ERR PC  TCSR"
1290  002174  025512          DT5                     ;$TESTN,$ERRPC,CTCSR
1291  002176  000000          0
1292
1293  002200  021736          EM103                   ;"SLU1 RCVR INTERRUPTS WITH ENABLE CLEAR"
1294  002202  024756          DH6                     ;"TEST# ERR PC  RCSR"
1295  002204  025522          DT6                     ;$TESTN,$ERRPC,CRCSR
1296  002206  000000          0
1297
1298  002210  022005          EM104                   ;"SLU1 RCVR DID NOT INTERRUPT"
1299  002212  024756          DH6                     ;"TEST# ERR PC  RCSR"
1300  002214  025522          DT6                     ;$TESTN,$ERRPC,CRCSR
1301  002216  000000          0
1302
1303  002220  022041          EM105                   ;"SLU1 RCVR INTERRUPTS AT PRIORITY 7"
1304  002222  024756          DH6                     ;"TEST# ERR PC  RCSR"
1305  002224  025522          DT6                     ;$TESTN,$ERRPC,CRCSR
1306  002226  000000          0
1307
1308  002230  022104          EM106                   ;"SLU1 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR"
1309  002232  024756          DH6                     ;"TEST# ERR PC  RCSR"
1310  002234  025522          DT6                     ;$TESTN,$ERRPC,CRCSR
1311  002236  000000          0
1312
1313  002240  022165          EM107                   ;"SLU1 RCVR DID NOT INTERRUPT"
1314  002242  024756          DH6                     ;"TEST# ERR PC  RCSR"
1315  002244  025522          DT6                     ;$TESTN,$ERRPC,CRCSR
1316  002246  000000          0
1317
1318  002250  022221          EM110                   ;"SLU1 RECEIVER RE-INTERRUPTED"
1319  002252  024756          DH6                     ;"TEST# ERR PC  RCSR"
1320  002254  025522          DT6                     ;$TESTN,$ERRPC,CRCSR
1321  002256  000000          0
1322
1323  002260  022256          EM111                   ;"SLU1 READING RBUF DID NOT CLEAR INTERRUPT"
1324  002262  024756          DH6                     ;"TEST# ERR PC  RCSR"
1325  002264  025522          DT6                     ;$TESTN,$ERRPC,CRCSR
1326  002266  000000          0
1327
1328  002270  022330          EM112                   ;"RESET DID NOT CLEAR SLU1 RCVR INTERRUPT"
1329  002272  024756          DH6                     ;"TEST# ERR PC  RCSR"
```

N2


```
1330  002274  025522          DT6                     ;$TESTN,$ERRPC,CRCSR
1331  002276  000000          0
1332
1333  002300  022400          EM113                   ;"SLU1 'OR' FLAG DID NOT SET"
1334  002302  024756          DH6                     ;"TEST# ERR PC  RCSR"
1335  002304  025522          DT6                     ;$TESTN,$ERRPC,CRCSR
1336  002306  000000          0
1337
1338  002310  022433          EM114                   ;"SLU1 'ERROR' NOT SET WITH 'OR' FLAG"
1339  002312  024756          DH6                     ;"TEST# ERR PC  RCSR"
1340  002314  025522          DT6                     ;$TESTN,$ERRPC,CRCSR
1341  002316  000000          0
1342
1343  002320  022477          EM115                   ;"DATA COMPARE ERROR"
1344  002322  025157          DH115                   ;"TEST# ERR PC  CRCSR  GOOD  BAD"
1345  002324  025630          DT115                   ;$TESTN,$ERRPC,CRCSR,GD,BD
1346  002326  000000          0
1347
1348  002330  022522          EM116                   ;"SLU2 XMIT INTERRUPTS WHEN DISABLED"
1349  002332  024731          DH5                     ;"TEST# ERR PC  TCSR"
1350  002334  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1351  002336  000000          0
1352
1353  002340  022565          EM117                   ;"SLU2 XMIT DID NOT INTERRUPT"
1354  002342  024731          DH5                     ;"TEST# ERR PC  TCSR"
1355  002344  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1356  002346  000000          0
1357
1358  002350  022621          EM120                   ;"SLU2 XMIT INTERRUPT AT PRIORITY 7"
1359  002352  024731          DH5                     ;"TEST# ERR PC  TCSR"
1360  002354  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1361  002356  000000          0
1362
1363  002360  022663          EM121                   ;"SLU2 XMIT INTERRUPTS WITH ENABLE CLEAR"
1364  002362  024731          DH5                     ;"TEST# ERR PC  TCSR"
1365  002364  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1366  002366  000000          0
1367
1368  002370  022732          EM122                   ;"SLU2 XMIT DID NOT INTERRUPT"
1369  002372  024731          DH5                     ;"TEST# ERR PC  TCSR"
1370  002374  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1371  002376  000000          0
1372
1373  002400  022766          EM123                   ;"SLU2 XMIT RE-INTERRUPTED"
1374  002402  024731          DH5                     ;"TEST# ERR PC  TCSR"
1375  002404  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1376  002406  000000          0
1377
1378  002410  023017          EM124                   ;"LOADING SLU2 TBUF DID NOT CLEAR INTERRUPT"
1379  002412  024731          DH5                     ;"TEST# ERR PC  TCSR"
1380  002414  025542          DT31                    ;$TESTN,$ERRPC,TCSR
1381  002416  000000          0
1382
1383  002420  023071          EM125                   ;"SLU2 RCVR INTERRUPTS WITH ENABLE CLEAR"
1384  002422  024756          DH6                     ;"TEST# ERR PC  RCSR"
1385  002424  025562          DT36                    ;$TESTN,$ERRPC,RCSR
```

```
1386  002426  000000                  0
1387
1388  002430  023140                  EM126           ;"SLU2 RCVR DID NOT INTERRUPT'
1389  002432  024756                  DH6             ;"TEST# ERR PC  RCSR"
1390  002434  025562                  DT36            ;$TESTN,$ERRPC,RCSR
1391  002436  000000                  0
1392
1393  002440  023174                  EM127           ;"SLU2 RCVR INTERRUPTS AT PRIORITY 7"
1394  002442  024756                  DH6             ;"TEST# ERR PC  RCSR"
1395  002444  025562                  DT36            ;$TESTN,$ERRPC,RCSR
1396  002446  000000                  0
1397
1398  002450  023237                  EM130           ;"SLU2 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR"
1399  002452  024756                  DH6             ;"TEST# ERR PC  RCSR"
1400  002454  025562                  DT36            ;$TESTN,$ERRPC,RCSR
1401  002456  000000                  0
1402
1403  002460  023320                  EM131           ;"SLU2 RCVR DID NOT INTERRUPT"
1404  002462  024756                  DH6             ;"TEST# ERR PC  RCSR"
1405  002464  025562                  DT36            ;$TESTN,$ERRPC,RCSR
1406  002466  000000                  0
1407
1408  002470  023354                  EM132           ;"SLU2 RECEIVER RE-INTERRUPTED"
1409  002472  024756                  DH6             ;"TEST# ERR PC  RCSR"
1410  002474  025562                  DT36            ;$TESTN,$ERRPC,RCSR
1411  002476  000000                  0
1412
1413  002500  023411                  EM133           ;"SLU2 READING RBUF DID NOT CLEAR INTERRUPT"
1414  002502  024756                  DH6             ;"TEST# ERR PC  RCSR"
1415  002504  025562                  DT36            ;$TESTN,$ERRPC,RCSR
1416  002506  000000                  0
1417
1418  002510  023463                  EM134           ;"RESET DID NOT CLEAR SLU2 RCVR INTERRUPT"
1419  002512  024756                  DH6             ;"TEST# ERR PC  RCSR"
1420  002514  025562                  DT36            ;$TESTN,$ERRPC,RCSR
1421  002516  000000                  0
1422
1423  002520  023533                  EM135           ;"SLU2 'OR' FLAG DID NOT SET"
1424  002522  024756                  DH6             ;"TEST# ERR PC  RCSR"
1425  002524  025562                  DT36            ;$TESTN,$ERRPC,RCSR
1426  002526  000000                  0
1427
1428  002530  023566                  EM136           ;"SLU2 'ERROR' NOT SET WITH 'OR' FLAG"
1429  002532  024756                  DH6             ;"TEST# ERR PC  RCSR"
1430  002534  025562                  DT36            ;$TESTN,$ERRPC,RCSR
1431  002536  000000                  0
1432
1433  002540  023632                  EM137           ;"SLU2 BREAK DID NOT TRANSMIT ALL ZEROES"
1434  002542  025223                  DH137           ;"TEST# ERR PC  RCSR  DATA"
1435  002544  025644                  DT137           ;$TESTN,$ERRPC,RCSR,$BDDAT
1436  002546  000000                  0
1437
1438  002550  023701                  EM140           ;"BREAK DID NOT SET FRAMING ERROR"
1439  002552  024756                  DH6             ;"TEST# ERR PC  RCSR"
1440  002554  025522                  DT6             ;$TESTN,$ERRPC,RCSR
1441  002556  000000                  0
```

C3


```
1442
1443  002560  023741          EM141           ;"SLU2 'ERROR' NOT SET WITH 'FR' FLAG
1444  002562  024756          DH6             ;"TEST# ERR PC  RCSR"
1445  002564  025562          DT36            ;$TESTN,$ERRPC,RCSR
1446  002566  000000          0
1447
1448  002570  024005          EM142           ;"DATA COMPARE ERROR WITH CABLE"
1449  002572  025260          DH142           ;"TEST# ERR PC  RCSR    GOOD    BAD"
1450  002574  025656          DT142           ;$TESTN,$ERRPC,RCSR,GD,BD
1451  002576  000000          0
1452
1453  002600  024043          EM143           ;"RTC INTERRUPT AT PRIORITY 7"
1454  002602  025055          DH61            ;"TEST# ERR PC  LKS"
1455  002604  025602          DT61            ;$TESTN,$ERRPC,LKS
1456  002606  000000          0
1457
1458  002610  024077          EM144           ;"RTC INTERRUPTS WHEN DISABLED"
1459  002612  025055          DH61            ;"TEST# ERR PC  LKS"
1460  002614  025602          DT61            ;$TESTN,$ERRPC,LKS
1461  002616  000000          0
1462
1463  002620  024134          EM145           ;"RTC INTERRUPT DID NOT OCCUR"
1464  002622  025055          DH61            ;"TEST# ERR PC  LKS"
1465  002624  025602          DT61            ;$TESTN,$ERRPC,LKS
1466  002626  000000          0
1467
1468  002630  024170          EM146           ;"RTC INTERRUPT DID NOT OCCUR"
1469  002632  025055          DH61            ;"TEST# ERR PC  LKS"
1470  002634  025602          DT61            ;$TESTN,$ERRPC,LKS
1471  002636  000000          0
1472
1473  002640  024224          EM147           ;"RTC DOUBLE INTERRUPT"
1474  002642  025055          DH61            ;"TEST# ERR PC  LKS"
1475  002644  025602          DT61            ;$TESTN,$ERRPC,LKS
1476  002646  000000          0
1477
1478  002650  024251          EM150           ;"RESET DID NOT CLEAR RTC INTERRUPT"
1479  002652  025055          DH61            ;"TEST# ERR PC  LKS"
1480  002654  025602          DT61            ;$TESTN,$ERRPC,LKS
1481  002656  000000          0
1482
1483  002660  024313          EM151           ;"RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS"
1484  002662  025055          DH61            ;"TEST# ERR PC  LKS"
1485  002664  025602          DT61            ;$TESTN,$ERRPC,LKS
1486  002666  000000          0
1487
1488
1489  002670  024422          EM153           ;"SLU1 RECEIVER STATUS ERROR"
1490  002672  025157          DH115           ;"TEST#  ERR PC  CRCSR  GOOD  BAD"
1491  002674  025630          DT115           ;$TESTN,$ERRPC,CRCSR,$GDDTA,$BDDAT
1492  002676  000000          0
1493
1494  002700  024455          EM154           ;SLU2 RECEIVER STATUS ERROR
1495  002702  025260          DH142           ;"TESTN  ERR PC  RCSR  GOOD  BAD"
1496  002704  025656          DT142           ;$TESTN,$ERRPC,RCSR,$GDDAT,$BDDAT
1497  002706  000000          0
```

```
1498
1499
1500  002710  024545                    EM156                   ;"SLU1 DATA COMPARE ERROR IN EXERCISER'
1501  002712  025157                    DH115                   ;"TEST# ERR PC  CRCSR   GOOD    BAD"
1502  002714  025630                    DT115                   ;$TESTN,$ERRPC,CRCSR,GD,BD
1503  002716  000000                    0
1504
1505  002720  024612                    EM157                   ;"INCORRECT RECEIVE COUNT SLU2
1506  002722  025361                    DH155                   ;"TEST# ERR PC  RCSR    TRANS   RCV"
1507  002724  025670                    DT157                   ;$TESTN,$ERRPC,RCSR,XMTCT2,RCVCT2
1508  002726  000000                    0
1509
1510  002730  024647                    EM160                   ;"SLU2 DATA COMPARE ERROR IN EXERCISER'
1511  002732  025260                    DH142                   ;"TEST# ERR PC  RCSR    GOOD    BAD"
1512  002734  025556                    DT142                   ;$TESTN,$ERRPC,RCSR,GD,BD
1513  002736  000000                    0
1514
1515  002740  024714                    EM161                   ;"TRAP CATCHER"
1516  002742  025425                    DH161                   ;"TEST#   ERR PC  OLDPC  TRAP ADR"
1517  002744  025704                    DT161                   ;$TESTN,$ERRPC,OLDPC,BDVECT
1518  002746  000000                    0
1519
1520                                ;REGISTER ADDRESSES OF INTERNAL ON BOARD OPTIONS
1521
1522  002750  176500                RCSR:   .WORD   176500      ;SLU2 COMMAND/STATUS REGISTER
1523  002752  176502                RBUF:   .WORD   176502      ;SLU2 RECEIVER BUFFER
1524  002754  176504                TCSR:   .WORD   176504      ;SLU2 TRANSMITTER COMMAND/STATUS REGISTER
1525  002756  176506                TBUF:   .WORD   176506      ;SLU2 TRANSMITTER BUFFER
1526  002760  177560                CRCSR:  177560              ;SLU1 RECEIVER COMMAND/STATUS REGISTER
1527  002762  177562                CRBUF:  177562              ;SLU1 RECEIVER BUFFER
1528  002764  177564                CTCSR:  177564              ;SLU1 TRANSMITTER COMMAND/STATUS REGISTER
1529  002766  177566                CTBUF:  177566              ;SLU1 TRANSMITTER BUFFER
1530  002770  177546                LKS:    .WORD   177546      ;LTC COMMAND/STATUS REGISTER
1531
1532
1533                                ;VECTOR ADDRESSES FOR ON BOARD OPTIONS
1534
1535  002772  000300                RVECT:  .WORD   300
1536  002774  000302                RPSW:   .WORD   302
1537  002776  000304                TVECT:  .WORD   304
1538  003000  000306                TPSW:   .WORD   306
1539  003002  000060                CRVECT: 60                  ;RECEIVER INTERRUPT VECTOR
1540  003004  000062                CRPSW:  62
1541  003006  000064                CTVECT: 64                  ;TRANSMITTER INTERRUPT VECTOR
1542  003010  000066                CTPSW:  66
1543  003012  000100                RTCVT:  .WORD   100
1544  003014  000102                RTCPSW: .WORD   102
1545
1546  003016  005037  001102        START:  CLR     $FATAL      ;CLEAR ERROR NO.
1547  003022  005037  001100                CLR     $MSGTYP     ;CLEAR MESSAGE TYPE
1548  003026  005037  001104                CLR     $TESTN      ;CLEAR TEST NO.
1549  003032  005037  001156                CLR     $DEVM       ;CLEAR FLAGS INDICATING DEVICES UNDER TEST
1550  003036                            1$:
1551                                .SBTTL  INITIALIZE THE COMMON TAGS
1552                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1553  003036  012706  001000                MOV     #$CMTAG,R6  ;;FIRST LOCATION TO BE CLEARED
```

```
1554  003042  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1555  003044  022706  001040                  CMP     #SWR,R6 ;;DONE?
1556  003050  001374                          BNE     .-6             ;;LOOP BACK IF NO
1557  003052  012706  001000                  MOV     #1000,SP                ;;SETUP THE STACK POINTER
1558                                  ;;INITIALIZE A FEW VECTORS
1559  003056  012737  013114  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1560  003064  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1561  003072  012737  012420  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1562  003100  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1563  003106  012737  015326  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1564  003114  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
1565  003122  012737  012736  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1566  003130  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1567  003136  013737  012270  012262          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
1568  003144  005037  001072                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1569  003150  112737  000001  001015          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1570  003156  012737  003156  001006          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1571  003164  012737  003164  001010          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1572                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1573                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1574  003172  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1575  003176  012737  003232  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1576  003204  012737  177570  001040          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1577  003212  012737  177570  001042          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1578  003220  022777  177777  175612          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1579  003226  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1580                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
1581  003230  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
1582  003232  012716  003240          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1583  003236  000002                          RTI
1584  003240  012737  000176  001040  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1585  003246  012737  000174  001042          MOV     #DISPREG,DISPLAY
1586  003254  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1587
1588  003260  005037  001106                  CLR     $PASS           ;;CLEAR PASS COUNT
1589  003264  132737  000200  001121          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1590  003272  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
1591  003274  012737  001122  001040          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1592  003302                          67$:
1593                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1594  003302  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1595  003306  001012                          BNE     68$             ;;BRANCH IF YES
1596  003310  123727  001120  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
1597  003316  001406                          BEQ     68$             ;;BRANCH IF YES
1598  003320  023727  001040  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1599  003326  001005                          BNE     69$             ;;BRANCH IF NO
1600  003330  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
1601  003332  000403                          BR      69$
1602  003334  112737  000001  001034  68$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1603  003342                          69$:
1604  003342  032777  000060  175470          BIT     #BIT4!BIT5,@SWR ;IS SLU1 TO BE TESTED
1605  003350  001003                          BNE     2$              ;IF EITHER BIT IS SET THEN DON'T TEST IT
1606  003352  052737  000001  001156          BIS     #BIT0,$DEVM     ;SET DEVICE FLAG TO TEST SLU1
1607  003360  032777  000050  175452  2$:     BIT     #BIT3!BIT5,@SWR ;IS SLU2 TO BE TESTED
1608  003366  001003                          BNE     3$              ;IF EITHER BIT IS SET THEN DON'T TEST IT
1609  003370  052737  000002  001156          BIS     #BIT1,$DEVM     ;SET DEVICE FLAG TO TEST SLU2
```

F3


```
1610  003376  032777  000100  175434  3$:     BIT     #BIT6,@SWR      ;IS LTC TO BE TESTED
1611  003404  001003                          BNE     4$              ;IF BIT IS SET THEN DON'T TEST IT.
1612  003406  052737  000004  001156          BIS     #BIT2,$DEVM     ;SET DEVICE FLAG TO TEST LTC
1613  003414  032737  000001  001156  4$:     BIT     #BIT0,$DEVM     ;IS SLU1 UNDER TEST
1614  003422  001002                          BNE     TST1            ;IF YES TEST XMIT REG. BEFORE TYPING TITLE
1615  003424  000137  003660                  JMP     ID              ;IF NO SKIP TESTS AND TYPE IT NOW
1616
1617
1618                                          ;;*********************************************************************
1619                                          ;*TEST 1         TEST ABILITY TO ACCESS SLU1 TCSR
1620                                          ;;*********************************************************************
1621  003430  000004                  TST1:   SCOPE
1622  003432  012737  000001  001104          MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1623  003440  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1624  003444  012737  003460  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1625  003452  005777  177306                  TST     @CTCSR          ;REFERENCE THE XMIT COMMAND/STATUS REG.
1626  003456  000405                          BR      2$              ; GO TO END OF TEST
1627  003460  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1628  003462  004737  013304                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
1629  003466  000001                          1                               ;;THE ERROR NUMBER (FROM APT LIST)
1630  003470  000000                          HALT
1631  003472  010337  000004          2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1632
1633
1634
1635                                          ;;*********************************************************************
1636                                          ;*TEST 2         TEST ABILITY TO ACCESS SLU1 TBUF
1637                                          ;;*********************************************************************
1638  003476  000004                  TST2:   SCOPE
1639  003500  012737  000002  001104          MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1640  003506  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1641  003512  012737  003526  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1642  003520  005777  177242                  TST     @CTBUF          ;REFERENCE THE XMIT BUFFER
1643  003524  000405                          BR      2$              ;GO TO END OF  TEST
1644  003526  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1645  003530  004737  013304                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
1646  003534  000002                          2                               ;;THE ERROR NUMBER (FROM APT LIST)
1647  003536  000000                          HALT
1648  003540  010337  000004          2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1649
1650
1651                                          ;;*********************************************************************
1652                                          ;*TEST 3         TEST SLU1 TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
1653                                          ;;*********************************************************************
1654  003544  000004                  TST3:   SCOPE
1655  003546  012737  000003  001104          MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1656  003554  032737  000001  001120          BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
1657  003562  001405                          BEQ     70$             ;IF NO THEN DO TEST
1658  003564  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
1659  003570  001402                          BEQ     70$             ;IF YES THEN DO THIS SERIES OF TESTS
1660  003572  000137  004402                  JMP     SLU2RT          ;IF NO THEN SKIP THIS SERIES OF TESTS
1661  003576  005077  177164          70$:    CLR     @CTBUF          ;LOAD XBUF
1662  003602  105777  177156                  TSTB    @CTCSR          ;CHECK DONE
1663  003606  100011                          BPL     1$              ;BR IF CLEAR
1664                                                                  ;FILL SECOND BUFFER, BECAUSE REFRESH COULD CAUSE
1665                                                                  ;   FIRST TEST TO FAIL
```

```
1666  003610  005077  177152          CLR     @CTBUF          ;FILL DOUBLE BUFFER
1667  003614  105777  177144          TSTB    @CTCSR          ;CHECK DONE
1668  003620  100004                  BPL     1$              ;BR IF CLEAR
1669  003622  004737  013304          JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
1670  003626  000003                  3                               ;;THE ERROR NUMBER (FROM APT LIST)
1671  003630  000000                  HALT                    ;TCSR "DONE" NOT CLEARED WITH TBUF FULL
1672  003632  005000          1$:     CLR     R0              ;CLEAR TIMER
1673  003634  105777  177124  2$:     TSTB    @CTCSR          ;CHECK FOR XMIT DONE
1674  003640  100407                  BMI     ID              ;IF DONE SETS, BR TO END OF TEST
1675  003642  005200                  INC     R0              ;INCREMENT TIMER
1676  003644  001373                  BNE     2$              ;BR IF TIMER NOT DONE
1677  003646  004737  013304          JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
1678  003652  000004                  4                               ;;THE ERROR NUMBER (FROM APT LIST)
1679  003654  000000                  HALT
1680  003656  000416                  BR      TST4            ;BR TO NEXT TEST, AND SKIP THE TYPEOUT THAT FOLLOWS
1681                                                          ;  BECAUSE OF THIS FAILURE
1682
1683  003660  023737  000042  000046  ID:     CMP     @#42,@#46       ;UNDER ACT11?
1684  003666  001412                  BEQ     6$              ;IF YES, SKIP IDENT. TYPEOUT
1685  003670  005737  001106          TST     $PASS           ;IS THIS THE FIRST PASS?
1686  003674  001007                  BNE     6$              ;IF NOT BR TO NEXT TEST & SKIP THE IDENTIFICATION TYPEOU
1687  003676  005737  001110          TST     $DEVCT          ;IS THIS THE FIRST SUBPASS?
1688  003702  001004                  BNE     6$              ;IF NOT, BRTO NEXT TEST
1689  003704  104401                  TYPE                    ;TYPE PROGRAM INDENTIFICATION
1690  003706  025476                  M1
1691  003710  104401                  TYPE                    ;TYPE NUMBER OF DEVICES UNDER TEST
1692  003712  025510                  M2
1693  003714                  6$:
1694
1695                                  ;;**********************************************************************
1696                                  ;*TEST 4         TEST THAT SLU1 TCSR "DONE" SETS WITH RESET
1697                                  ;;**********************************************************************
1698  003714  000004          TST4:   SCOPE
1699  003716  012737  000004  001104          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1700  003724  032737  000001  001156          BIT     #BIT0,@#$DEVM   ;DO THESE TESTS FOR THIS DEVICE?
1701  003732  001002                  BNE     99$             ;F YES CONTINUE WITH TESTS
1702  003734  000137  004402          JMP     SLU2RT          ;IF NO GO TO START OF NEXT SET OF TESTS.
1703  003740                  99$:
1704  003740  005077  177022          CLR     @CTBUF          ;LOAD TRANSMIT BUFFER
1705  003744  105777  177014  1$:     TSTB    @CTCSR          ;WAIT FOR DONE
1706  003750  100375                  BPL     1$
1707  003752  005077  177010          CLR     @CTBUF          ;LOAD SECOND BUFFER
1708  003756  000240                  NOP
1709  003760  000005                  RESET                   ;CLEAR DONE WITH RESET
1710  003762  105777  176776          TSTB    @CTCSR          ;CHECK FOR DONE SET
1711  003766  100401                  BMI     TST5            ;BR TO NEXT TEST IF DONE SET
1712
1713  003770  104005                  ERROR   5               ;TCSR "DONE" DOES NOT SET WITH RESET
1714
1715
1716
1717                                  ;;**********************************************************************
1718                                  ;*TEST 5         TEST ABILITY TO ACCESS SLU1 RCSR
1719                                  ;;**********************************************************************
1720  003772  000004          TST5:   SCOPE
1721  003774  012737  000005  001104          MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
```

```
1722  004002  013703  000004          MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1723  004006  012737  004022  000004  MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1724  004014  005777  176740          TST     @CRCSR          ;ACCESS RCSR
1725  004020  000402                  BR      2$              ;BR TO END OF TEST
1726
1727  004022  022626          1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1728  004024  104006                  ERROR   6               ;CAN NOT ACCESS RCSR
1729  004026  010337  000004  2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1730
1731
1732                          ;;*****************************************************************
1733                          ;*TEST 6         TEST ABILITY TO ACCESS SLU1 RBUF
1734                          ;;*****************************************************************
1735  004032  000004  TST6:   SCOPE
1736  004034  012737  000006  001104  MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1737  004042  013703  000004          MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1738  004046  012737  004062  000004  MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1739  004054  005777  176702          TST     @CRBUF          ;ACCESS RBUF
1740  004060  000402                  BR      2$              ;BR TO END OF TEST
1741
1742  004062  022626          1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1743  004064  104007                  ERROR   7               ;CAN NOT ACCESS RBUF
1744  004066  010337  000004  2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1745
1746
1747
1748
1749
1750                          ;;*****************************************************************
1751                          ;*TEST 7         TEST THAT SLU1 BIT6(XMIT INT EN) CAN BE SET & RESET
1752                          ;;*****************************************************************
1753  004072  000004  TST7:   SCOPE
1754  004074  012737  000007  001104  MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1755  004102  042777  000100  176654  BIC     #BIT6,@CTCSR    ;MAKE SURE BIT UNDER TEST IS INITIALIZED
1756  004110  017703  176672          MOV     @CTVECT,R3      ;SAVE XMIT VECTOR
1757  004114  012777  004144  176664  MOV     #1$,@CTVECT     ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1758  004122  004737  012356          JSR     PC,WRPSW                ;SET PSW TO PRIORITY=7
1759  004126  000340                  .WORD   340
1760  004130  032777  000100  176626  BIT     #BIT6,@CTCSR    ;TEST BIT6 OF TCSR
1761  004136  001404                  BEQ     2$              ;BR IF ZERO
1762  004140  104014                  ERROR   14
1763                                                          ;BIT6 IN TCSR NOT CLEAR AFTER RESET
1764  004142  000402                  BR      2$
1765
1766  004144  022626          1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1767  004146  104015                  ERROR   15
1768                                                          ;XMIT INTERRUPT OCCURRED  PRIO=7
1769
1770  004150  052777  000100  176606  2$:     BIS     #BIT6,@CTCSR    ;SET BIT6 OF TCSR
1771  004156  032777  000100  176600          BIT     #BIT6,@CTCSR    ;TEST BIT6 OF TCSR
1772  004164  001001                  BNE     3$              ;BR, IF SET
1773
1774  004166  104016                  ERROR   16
1775                                                          ;CANNOT SET BIT6 OF TCSR
1776
1777  004170  042777  000100  176566  3$:     BIC     #BIT6,@CTCSR    ;CLEAR BIT6 OF TCSR
```

```
1778  004176  032777  000100  176560          BIT     #BIT6,@CTCSR    ;TEST BIT6 OF TCSR
1779  004204  001401                          BEQ     4$              ;BR IF CLEAR
1780  004206  104017                          ERROR   17
1781                                                                  ;CANNOT CLEAR BIT6 OF TCSR
1782
1783  004210                          4$:
1784  004210  052777  000100  176546          BIS     #BIT6,@CTCSR    ;SET BIT6 OF TCSR
1785  004216  000005                          RESET                   ;CLEAR BIT6 WITH RESET
1786  004220  032777  000100  176536          BIT     #BIT6,@CTCSR    ;TEST BIT6 OF TCSR
1787  004226  001401                          BEQ     5$              ;BR IF CLEAR
1788
1789  004230  104020                          ERROR   20
1790                                                                  ;CANNOT CLEAR BIT6 OF TCSR WITH RESET
1791  004232  010377  176550          5$:     MOV     R3,@CTVECT      ;RESTORE XMIT VECTOR
1792
1793
1794                                  ;;*****************************************************************
1795                                  ;*TEST 10        TEST THAT SLU1 BIT6 OF RCSR CAN BE SET & RESET
1796                                  ;;*****************************************************************
1797  004236  000004                  TST10:  SCOPE
1798  004240  012737  000010  001104          MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1799  004246  042777  000100  176504          BIC     #BIT6,@CRCSR    ;MAKE SURE BIT UNDER TEST IS INITIALIZED
1800  004254  017703  176522                  MOV     @CRVECT,R3      ;SAVE RECEIVE VECTOR
1801  004260  012777  004310  176514          MOV     #1$,@CRVECT     ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1802  004266  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY=7
1803  004272  000340                                  .WORD   340
1804  004274  032777  000100  176456          BIT     #BIT6,@CRCSR    ;TEST BIT6 OF RCSR
1805  004302  001404                          BEQ     2$
1806  004304  104021                          ERROR   21
1807                                                                  ;BIT6 OF RCSR NOT CLEAR AFTER RESET
1808  004306  000402                          BR      2$
1809
1810  004310  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1811  004312  104022                          ERROR   22
1812                                                                  ;RCVR INTERRUPT WITH PRIORITY=7
1813
1814  004314  052777  000100  176436  2$:     BIS     #BIT6,@CRCSR    ;SET BIT6 OF RCSR
1815  004322  032777  000100  176430          BIT     #BIT6,@CRCSR    ;TEST BIT6 OF RCSR
1816  004330  001001                          BNE     3$              ;BR, IF SET
1817
1818  004332  104023                          ERROR   23
1819                                                                  ;CANNOT SET BIT6 OF RCSR
1820
1821  004334  042777  000100  176416  3$:     BIC     #BIT6,@CRCSR    ;CLEAR BIT6 OF RCSR
1822  004342  032777  000100  176410          BIT     #BIT6,@CRCSR    ;TEST BIT6 OF RCSR
1823  004350  001401                          BEQ     4$              ;BR, IF CLEAR
1824
1825  004352  104024                          ERROR   24
1826                                                                  ;CANNOT CLEAR BIT6 OF RCSR
1827
1828  004354                          4$:
1829  004354  052777  000100  176376          BIS     #BIT6,@CRCSR    ;SET BIT6 OF RCSR
1830  004362  000005                          RESET                   ;CLEAR BIT6OF RCSR WITH RESET
1831  004364  032777  000100  176366          BIT     #BIT6,@CRCSR    ;TEST BIT6 OF RCSR
1832  004372  001401                          BEQ     5$              ;BR, IF CLEAR
1833
```

J3


```
1834  004374  104025                    ERROR   25
1835                                                ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
1836  004376  010377  176400    5$:     MOV     R3,@CRVECT      ;RESTORE RECEIVE VECTOR
1837
1838
1839  004402                    SLU2RT:
1840
1841                            ;;**********************************************************************
1842                            ;*TEST 11       TEST ABILITY TO ACCESS SLU2 TCSR
1843                            ;;**********************************************************************
1844  004402  000004            TST11:  SCOPE
1845  004404  012737  000011  001104    MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1846  004412  032737  000002  001156    BIT     #BIT1,@#$DEVM   ;DO THESE TESTS FOR THIS DEVICE?
1847  004420  001002                    BNE     99$             ;F YES CONTINUE WITH TESTS
1848  004422  000137  005626            JMP     LTCRT           ;IF NO GO TO START OF NEXT SET OF TESTS.
1849  004426                    99$:
1850  004426  013703  000004            MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1851  004432  012737  004446  000004    MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1852  004440  005777  176310            TST     @TCSR           ;REFERENCE THE XMIT COMMAND/STATUS REG.
1853  004444  000402                    BR      2$              ; GO TO END OF TEST
1854  004446  022626            1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1855  004450  104031                    ERROR   31              ;XMIT CSR ADDRESS DOES NOT RETURN SSYNC
1856  004452  010337  000004    2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1857
1858
1859
1860                            ;;**********************************************************************
1861                            ;*TEST 12       TEST ABILITY TO ACCESS SLU2 TBUF
1862                            ;;**********************************************************************
1863  004456  000004            TST12:  SCOPE
1864  004460  012737  000012  001104    MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1865  004466  013703  000004            MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1866  004472  012737  004506  000004    MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1867  004500  005777  176252            TST     @TBUF           ;REFERENCE THE XMIT BUFFER
1868  004504  000402                    BR      2$              ;GO TO END OF  TEST
1869  004506  022626            1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1870  004510  104032                    ERROR   32              ;XMIT BUFFER ADDRESS DOES NOT REUTRN SSYNC
1871  004512  010337  000004    2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1872
1873
1874                            ;;**********************************************************************
1875                            ;*TEST 13       TEST SLU2 TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
1876                            ;;**********************************************************************
1877  004516  000004            TST13:  SCOPE
1878  004520  012737  000013  001104    MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1879  004526  032737  000001  001120    BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
1880  004534  001403                    BEQ     70$             ;IF NO THEN DO TEST
1881  004536  005737  001106            TST     @#$PASS         ;IS THIS FIRST PASS
1882  004542  001022                    BNE     TST14           ;IF NO THEN SKIP TO NEXT TEST
1883  004544                    70$:
1884  004544  005077  176206            CLR     @TBUF           ;LOAD XBUF
1885  004550  105777  176200            TSTB    @TCSR           ;CHECK DONE
1886  004554  100006                    BPL     1$              ;BR IF CLEAR
1887                                                ;FILL SECOND BUFFER, BECAUSE REFRESH COULD CAUSE
1888                                                ;   FIRST TEST TO FAIL
1889  004556  005077  176174            CLR     @TBUF           ;FILL DOUBLE BUFFER
```

```
                                        TSTB    @TCSR           ;CHECK DONE
                                        BPL     1$              ;BR IF CLEAR
                                        ERROR   33              ;TCSR "DONE" NOT CLEARED WITH TBUF FULL
                                1$:     CLR     R0              ;CLEAR TIMER
                                2$:     TSTB    @TCSR           ;CHECK FOR XMIT DONE
                                        BMI     TST17           ;IF DONE SETS, BR TO NEXT TEST
                                        INC     R0              ;INCREMENT TIMER
                                        BNE     2$              ;BR IF TIMER NOT DONE
                                        ERROR   34              ;XMIT DONE BIT DOES NOT RESET AFTER TRANSMIT


                                ;;**********************************************************************
                                ;*TEST 14        TEST THAT SLU2 TCSR "DONE" SETS WITH RESET
                                ;;**********************************************************************
                                TST14:  SCOPE
                                        MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                        BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
                                        BEQ     70$             ;IF NO THEN DO TEST
                                        TST     @#$PASS         ;IS THIS FIRST PASS
                                        BNE     TST15           ;IF NO THEN SKIP TO NEXT TEST
                                70$:
                                        CLR     @TBUF           ;LOAD TRANSMIT BUFFER
                                1$:     TSTB    @TCSR           ;WAIT FOR DONE
                                        BPL     1$
                                        CLR     @TBUF           ;LOAD SECOND BUFFER
                                        NOP
                                        RESET                   ;CLEAR DONE WITH RESET
                                        TSTB    @TCSR           ;CHECK FOR DONE SET
                                        BMI     TST20           ;BR TO NEXT TEST IF DONE SET

                                        ERROR   35              ;TCSR "DONE" DOES NOT SET WITH RESET


                                ;;**********************************************************************
                                ;*TEST 15        TEST ABILITY TO ACCESS SLU2 RCSR
                                ;;**********************************************************************
                                TST15:  SCOPE
                                        MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                        MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
                                        MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
                                        TST     @RCSR           ;ACCESS RCSR
                                        BR      2$              ;BR TO END OF TEST

                                1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
                                        ERROR   36              ;CAN NOT ACCESS RCSR
                                2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR


                                ;;**********************************************************************
                                ;*TEST 16        TEST ABILITY TO ACCESS SLU2 RBUF
                                ;;**********************************************************************
                                TST16:  SCOPE
                                        MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                        MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
```

L.3


```
1946  004744  012737  004760  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1947  004752  005777  175774                  TST     @RBUF           ;ACCESS RBUF
1948  004756  000402                          BR      2$              ;BR TO END OF TEST
1949
1950  004760  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1951  004762  104037                          ERROR   37              ;CAN NOT ACCESS RBUF
1952  004764  010337  000004          2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1953
1954
1955
1956                                  ;;**********************************************************************
1957                                  ;*TEST 17        TEST SLU2 BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET
1958                                  ;;**********************************************************************
1959  004770  000004                  TST17:  SCOPE
1960  004772  012737  000017  001104          MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1961  005000  042777  000001  175746          BIC     #BIT0,@TCSR     ;MAKE SURE BIT UNDER TEST IS INITIALIZED
1962  005006  032777  000001  175740          BIT     #BIT0,@TCSR     ;CHECK BIT0 OF TCSR CLEAR
1963  005014  001401                          BEQ     1$              ;BR IF CLEAR
1964  005016  104040                          ERROR   40              ;BIT0 WAS NOT CLEAR AFTER RESET
1965  005020  052777  000001  175726  1$:     BIS     #BIT0,@TCSR     ;SET BIT0 IN TCSR
1966  005026  032777  000001  175720          BIT     #BIT0,@TCSR     ;TEST BIT0 OF TCSR
1967  005034  001001                          BNE     2$              ;BR IF SET
1968  005036  104041                          ERROR   41              ;BIT0 OF TCSR WILL NOT SET
1969  005040  042777  000001  175706  2$:     BIC     #BIT0,@TCSR     ;CLEAR BIT0 OF TCSR
1970  005046  032777  000001  175700          BIT     #BIT0,@TCSR     ;TEST BIT0 OF TCSR
1971  005054  001401                          BEQ     3$
1972  005056  104042                          ERROR   42              ;BIT0 OF TCSR WILL NOT CLEAR
1973  005060                          3$:
1974  005060  032737  000001  001120          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
1975  005066  001403                          BEQ     70$             ;IF NO THEN DO TEST
1976  005070  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
1977  005074  001014                          BNE     TST20           ;IF NO THEN SKIP TO NEXT TEST
1978  005076                          70$:
1979  005076  052777  000001  175650          BIS     #BIT0,@TCSR     ;SET BIT0 IN TCSR
1980  005104  000005                          RESET                   ;CLEAR BIT0 WITH RESET
1981  005106  032777  000001  175640          BIT     #BIT0,@TCSR     ;TEST BIT0 CLEAR
1982  005114  001404                          BEQ     TST20                   ;BR IF CLEAR
1983  005116  042777  000001  175630          BIC     #BIT0,@TCSR     ;CLEAR BIT0, TO PRINT ERROR
1984  005124  104043                          ERROR   43              ;RESET DID NOT CLEAR BIT0 OF TCSR
1985
1986
1987                                  ;;**********************************************************************
1988                                  ;*TEST 20        TEST THAT SLU2 BIT6(XMIT INT EN) CAN BE SET & RESET
1989                                  ;;**********************************************************************
1990  005126  000004                  TST20:  SCOPE
1991  005130  012737  000020  001104          MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1992  005136  042777  000100  175610          BIC     #BIT6,@TCSR     ;MAKE SURE BIT UNDER TEST IS INITIALIZED
1993  005144  017703  175626                  MOV     @TVECT,R3       ;SAVE XMIT VECTOR
1994  005150  012777  005200  175620          MOV     #1$,@TVECT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1995  005156  004737  012356                  JSR     PC,WRPSW                ;SET PSW TO PRIORITY=7
1996  005162  000340                          .WORD   340
1997  005164  032777  000100  175562          BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
1998  005172  001404                          BEQ     2$              ;BR IF ZERO
1999  005174  104044                          ERROR   44
2000                                                                  ;BIT6 IN TCSR NOT CLEAR AFTER RESET
2001  005176  000402                          BR      2$
```

M3


```
2002
2003  005200  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2004  005202  104045                          ERROR   45
2005                                                                  ;XMIT INTERRUPT OCCURRED  PRIO=7
2006
2007  005204  052777  000100  175542  2$:     BIS     #BIT6,@TCSR     ;SET BIT6 OF TCSR
2008  005212  032777  000100  175534          BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
2009  005220  001001                          BNE     3$              ;BR, IF SET
2010
2011  005222  104046                          ERROR   46
2012                                                                  ;CANNOT SET BIT6 OF TCSR
2013
2014  005224  042777  000100  175522  3$:     BIC     #BIT6,@TCSR     ;CLEAR BIT6 OF TCSR
2015  005232  032777  000100  175514          BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
2016  005240  001401                          BEQ     4$              ;BR IF CLEAR
2017  005242  104047                          ERROR   47
2018                                                                  ;CANNOT CLEAR BIT6 OF TCSR
2019
2020  005244  032737  000001  001120  4$:     BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
2021  005252  001403                          BEQ     70$             ;IF NO THEN DO TEST
2022  005254  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
2023  005260  001011                          BNE     5$              ;IF NO THEN SKIP TO END OF TEST
2024  005262                          70$:
2025  005262  052777  000100  175464          BIS     #BIT6,@TCSR     ;SET BIT6 OF TCSR
2026  005270  000005                          RESET                   ;CLEAR BIT6 WITH RESET
2027  005272  032777  000100  175454          BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
2028  005300  001401                          BEQ     5$              ;BR IF CLEAR
2029
2030  005302  104050                          ERROR   50
2031                                                                  ;CANNOT CLEAR BIT6 OF TCSR WITH RESET
2032  005304  010377  175466          5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
2033
2034
2035                                  ;;**********************************************************************
2036                                  ;*TEST 21        TEST THAT SLU2 BIT6 OF RCSR CAN BE SET & RESET
2037                                  ;;**********************************************************************
2038  005310  000004                  TST21:  SCOPE
2039  005312  012737  000021  001104          MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2040  005320  042777  000100  175422          BIC     #BIT6,@RCSR     ;MAKE SURE BIT UNDER TEST IS INITIALIZED
2041  005326  017703  175440                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
2042  005332  012777  005362  175432          MOV     #1$,@RVECT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
2043  005340  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY=7
2044  005344  000340                          .WORD   340
2045  005346  032777  000100  175374          BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
2046  005354  001404                          BEQ     2$
2047  005356  104051                          ERROR   51
2048                                                                  ;BIT6 OF RCSR NOT CLEAR AFTER RESET
2049  005360  000402                          BR      2$
2050
2051  005362  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2052  005364  104052                          ERROR   52
2053                                                                  ;RCVR INTERRUPT WITH PRIORITY=7
2054
2055  005366  052777  000100  175354  2$:     BIS     #BIT6,@RCSR     ;SET BIT6 OF RCSR
2056  005374  032777  000100  175346          BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
2057  005402  001001                          BNE     3$              ;BR, IF SET
```

N3


```
2058
2059  005404  104053                          ERROR   53
2060                                                    ;CANNOT SET BIT6 OF RCSR
2061
2062  005406  042777  000100  175334  3$:     BIC     #BIT6,@RCSR     ;CLEAR BIT6 OF RCSR
2063  005414  032777  000100  175326          BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
2064  005422  001401                          BEQ     4$              ;BR, IF CLEAR
2065
2066  005424  104054                          ERROR   54
2067                                                    ;CANNOT CLEAR BIT6 OF RCSR
2068
2069  005426  032737  000001  001120  4$:     BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
2070  005434  001403                          BEQ     70$             ;IF NO THEN DO TEST
2071  005436  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
2072  005442  001011                          BNE     5$              .IF NO THEN SKIP TO END OF TEST
2073  005444                          70$:
2074  005444  052777  000100  175276          BIS     #BIT6,@RCSR     ;SET BIT6 OF RCSR
2075  005452  000005                          RESET                   .CLEAR BIT6OF RCSR WITH RESET
2076  005454  032777  000100  175266          BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
2077  005462  001401                          BEQ     5$              ;BR, IF CLEAR
2078
2079  005464  104055                          ERROR   55
2080                                                    ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
2081  005466  010377  175300          5$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
2082
2083
2084
2085                                  ;;**********************************************************************
2086                                  ;*TEST 22       TEST THAT SLU2 RCVR DONE (7) SET & CLEAR PROPERLY
2087                                  ;;**********************************************************************
2088  005472  000004                  TST22:  SCOPE
2089  005474  012737  000022  001104          MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2090  005502  005077  175244                  CLR     @RBUF           ;INITIALIZE REGISTER BEFORE TESTING
2091  005506  005000                          CLR     R0              ;CLEAR TIMER
2092  005510  005077  175242                  CLR     @TBUF           ;LOAD TRANSMIT BUFFER
2093  005514  105777  175230          WDONE:  TSTB    @RCSR           ;CHECK FOR RECEIVER DONE
2094  005520  100403                          BMI     1$              ;BR, IF DONE
2095  005522  005200                          INC     R0              ;INCREMENT TIMER, IF NOT DONE
2096  005524  001373                          BNE     WDONE           ;CONTINUE WAIT IF TIME REMAINS
2097  005526  104056                          ERROR   56
2098                                                    ;RECEIVER DONE NEVER SET
2099
2100  005530  032737  000001  001120  1$:     BIT #1,@#$ENV   ;ARE WE RUNNING UNDER APT
2101  005536  001403                          BEQ     70$             ;IF NO THEN DO TEST
2102  005540  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
2103  005544  001005                          BNE     2$              ;IF NO THEN SKIP TO END OF TEST
2104  005546                          70$:
2105  005546  000005                          RESET                   ;CLEAR DONE WITH RESET
2106  005550  105777  175174                  TSTB    @RCSR           ;CHECK FOR DONE CLEAR
2107  005554  001401                          BEQ     2$
2108
2109  005556  104057                          ERROR   57
2110                                                    ;RESET DID NOT CLEAR RCVR DONE
2111  005560  005777  175166          2$:     TST     @RBUF           ;CLEAR RECEIVER BUFFER
2112
2113
```

```
2114
2115                                    ;;*****************************************************************
2116                                    ;*TEST 23       TEST SLU2 THAT READING RBUF CLEARS RECEIVER DONE
2117                                    ;;*****************************************************************
2118  005564  000004                  TST23:  SCOPE
2119  005566  012737  000023  001104          MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2120  005574  005077  175152                  CLR     @RBUF           ;INITIALIZE REGISTER BEFORE TESTING
2121  005600  005077  175152                  CLR     @TBUF           ;LOAD TRANSMITTER
2122  005604  105777  175140          1$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
2123  005610  100375                          BPL     1$
2124  005612  017700  175134                  MOV     @RBUF,R0        ;READ RECEIVE BUFFER
2125  005616  105777  175126                  TSTB    @RCSR           ;CHECK FOR RECEIVE DONE CLEAR
2126  005622  001401                          BEQ     TST24           ;BR, IF CLEAR TO NEXT TEST
2127  005624  104060                          ERROR   60
2128                                                                  ;READING RBUF DID NOT CLEAR RCVR DONE
2129  005626                          LTCRT:
2130
2131                                    ;;*****************************************************************
2132                                    ;*TEST 24       TEST ABILITY TO ACCESS LKS
2133                                    ;;*****************************************************************
2134  005626  000004                  TST24:  SCOPE
2135  005630  012737  000024  001104          MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2136  005636  032737  000004  001156          BIT     #BIT2,@#$DEVM   ;DO THESE TESTS FOR THIS DEVICE?
2137  005644  001002                          BNE     99$             ;F YES CONTINUE WITH TESTS
2138  005646  000137  006100                  JMP     UNIQUE          ;IF NO GO TO START OF NEXT SET OF TESTS.
2139  005652                          99$:
2140  005652  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
2141  005656  012737  005672  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
2142  005664  005777  175100                  TST     @LKS            ;ACCESS LKS
2143  005670  000402                          BR      2$              ;NO TIMEOUT -  BR TO END OF TEST
2144
2145  005672  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
2146  005674  104061                          ERROR   61              ;CAN NOT ACCESS LKS
2147
2148  005676  010337  000004          2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
2149
2150                                    ;;*****************************************************************
2151                                    ;*TEST 25       TEST THAT BIT6 OF LKS CAN BE SET & RESET
2152                                    ;;*****************************************************************
2153  005702  000004                  TST25:  SCOPE
2154  005704  012737  000025  001104          MOV     #25,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2155  005712  042777  000100  175050          BIC     #BIT6,@LKS      ;MAKE SURE BIT UNDER TEST IS INITIALIZED BEFORE TESTING
2156  005720  017703  175066                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
2157  005724  012777  005754  175060          MOV     #1$,@RTCVT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
2158  005732  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2159  005736  000340                                  .WORD   340
2160  005740  032777  000100  175022          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
2161  005746  001404                          BEQ     2$
2162  005750  104062                          ERROR   62
2163                                                                  ;BIT6 OF LKS NOT CLEAR AFTER RESET
2164  005752  000402                          BR      2$
2165
2166  005754  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2167  005756  104063                          ERROR   63
2168                                                                  ;LKS INTERRUPT WITH PRIORITY=7
2169
```

C4


```
2170  005760  052777  000100  175002  2$:     BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
2171  005766  032777  000100  174774          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
2172  005774  001001                          BNE     3$              ;BR IF SET
2173
2174  005776  104064                          ERROR   64
2175                                                                  ;CANNOT SET BIT6 OF LKS
2176
2177  006000  042777  000100  174762  3$:     BIC     #BIT6,@LKS      ;CLEAR BIT6 OF LKS
2178  006006  032777  000100  174754          BIT     #BIT6,@LKS      ;TEST BIT6 OF LK
2179  006014  001401                          BEQ     4$
2180  006016  104065                          ERROR   65
2181                                                                  ;CANNOT CLEAR BIT6 OF LKS
2182  006020  032737  000001  001120  4$:     BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
2183  006026  001403                          BEQ     70$             ;IF NO THEN DO TEST
2184  006030  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
2185  006034  001011                          BNE     5$              ;IF NO THEN SKIP TO END OF TEST
2186  006036                          70$:
2187  006036  052777  000100  174724          BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
2188  006044  000005                          RESET                   ;CLEAR BIT6 OF LKS WITH RESET
2189  006046  032777  000100  174714          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
2190  006054  001401                          BEQ     5$              ;BR IF CLEAR
2191
2192  006056  104066                          ERROR   66
2193                                                                  ;CANNOT CLEAR BIT6 OF LKS WITH RESET
2194  006060  010377  174726          5$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
2195  006064  000405                          BR      UNIQUE                  ;GO TO NEXT TEST
2196
2197
2198  006066  177560                  DADTBL: .WORD   177560
2199  006070  177564                          .WORD   177564
2200  006072  176500                          .WORD   176500
2201  006074  176504                          .WORD   176504
2202  006076  177564                  TBLEND: .WORD   177564
2203
2204
2205  006100                          UNIQUE:
2206
2207                                  ;;*******************************************************************
2208                                  ;*TEST 26        UNIQUE INTERNAL ADDRESS TEST
2209                                  ;;*******************************************************************
2210  006100  000004                  TST26:  SCOPE
2211  006102  012737  000026  001104          MOV     #26,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2212  006110  032737  000001  001120          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
2213  006116  001403                          BEQ     70$             ;IF NO THEN DO TEST
2214  006120  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
2215  006124  001056                          BNE     TST27           ;IF NO THEN SKIP TO NEXT TEST
2216  006126                          70$:
2217  006126  012737  000340  177776          MOV     #340,PS         ;WE WILL BE PLAYING WITH BIT6
2218                                                                  ;SO LOCK OUT EXTRAINEOUS INTERRUPTS
2219  006134  012700  006066                  MOV     #DADTBL,R0      ;GET LOCATION OF FIRST REGISTER ADDRESS
2220  006140  012703  006066          1$:     MOV     #DADTBL,R3      ;MAKE R3 POINT TO LOCATION OF FIRST
2221                                                                  ;REGISTER ADDRESS
2222  006144  012701  000005                  MOV     #5,R1           ;SET LOOP COUNTER TO CLEAR ALL REG.
2223  006150  005033                  2$:     CLR     @(R3)+          ;CLEAR A REGISTER
2224  006152  077102                          SOB     R1,2$           ;LOOP UNTIL ALL REGISTERS CLEARED
2225  006154  012770  000100  000000          MOV     #BIT6,@(R0)     ;SET TEST BIT IN DEVICE REGISTERS
```

```
2226  006162  012701  006066          MOV     #DADTBL,R1      ;GET LOCATION OF FIRST REGISTER ADDRESS
2227  006166  012702  000005          MOV     #5,R2           ;SET UP TEST LOOP COUNTER
2228  006172  032731  000100  3$:     BIT     #BIT6,@(R1)+    ;IS TEST BIT SET IN THIS REGISTER
2229  006176  001006                  BNE     5$              ;IF YES GO SEE IF THERE IS AN ERROR
2230  006200  077204          4$:     SOB     R2,3$           ;LOOP UNTIL ALL REGISTER CHECKED
2231  006202  005030                  CLR     @(R0)+          ;CLEAR REGISTER JUST TESTED AND POINT
2232                                                          ;TO NEXT ONE
2233  006204  020027  006076          CMP     R0,#TBLEND
2234  006210  001421                  BEQ     7$
2235  006212  000752                  BR      1$              ;CONTINUE TESTING
2236  006214  021041          5$:     CMP     (R0),-(R1)      ;DID WE COMPARE THE REGISTER TO ITSELF?
2237  006216  001413                  BEQ     6$              ;IF YES GET OVER ERROR CALL
2238  006220  011037  001020          MOV     (R0),$GDADR     ;IF NO SET UP ERROR INFORMATION
2239  006224  011137  001022          MOV     (R1),$BDADR
2240  006230  017037  000000  001024  MOV     @(R0),$GDDAT
2241  006236  017137  000000  001026  MOV     @(R1),$BDDAT
2242  006244  104072                  ERROR   72              ;WRITE TO 1 INTERNAL ADDRESS MODIFIED
2243                                                          ;ANOTHER SO ADDRESS NOT UNIQUE
2244  006246  062701  000002  6$:     ADD     #2,R1           ;RESTORE POINTER
2245  006252  000752                  BR      4$              ;GET BACK IN TEST LOOP
2246  006254  005000          7$:     CLR     R0
2247  006256  005200          8$:     INC     R0
2248  006260  001376                  BNE     8$
2249
2250  006262                  SLU1IT:
2251
2252                          ;;**************************************************************
2253                          ;*TEST 27       TEST THAT SLU1 XMIT INTERRUPTS ONLY WHEN ENABLED
2254                          ;;**************************************************************
2255  006262  000004  TST27:  SCOPE
2256  006264  012737  000027  001104  MOV     #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2257  006272  032737  000001  001120  BIT     #1,@#$ENV       ;IF NOT UNDER APT
2258  006300  001405                  BEQ     70$             ;  THEN RUN THIS SERIES OF TESTS
2259  006302  005737  001106          TST     @#$PASS         ;  ELSE IF FIRST PASS
2260  006306  001402                  BEQ     70$             ;         THEN RUN THESE TESTS
2261  006310  000137  007016          JMP     SLU2IT          ;         ELSE DO NOT RUN THESE TESTS
2262  006314          70$:
2263  006314  032737  000001  001156  BIT     #BIT0,@#$DEVM   ;DO THESE TESTS FOR THIS DEVICE?
2264  006322  001002                  BNE     99$             ;F YES CONTINUE WITH TESTS
2265  006324  000137  007016          JMP     SLU2IT          ;IF NO GO TO START OF NEXT SET OF TESTS.
2266  006330          99$:
2267  006330  000005                  RESET                   ;CLEAR THE WORLD
2268  006332  042777  000100  174424  BIC     #BIT6,@CTCSR    ;CLEAR TRANSMIT INTERRUPT ENABLE
2269  006340  017703  174442          MOV     @CTVECT,R3      ;SAVE XMIT VECTOR
2270  006344  012777  006370  174434  MOV     #2$,@CTVECT     ;POINT XMIT VECTOR TO ERROR REPORT
2271  006352  105777  174406  1$:     TSTB    @CTCSR          ;WAIT FOR DONE
2272  006356  100375                  BPL     1$
2273  006360  004737  012356          JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2274  006364  000140                  .WORD   140
2275  006366  000402                  BR      3$
2276
2277  006370  022626          2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2278  006372  104074                  ERROR   74
2279                                                          ;XMIT INTERRUPTS WITH INTERRUPT ENABLE CLEAR
2280  006374  012777  006414  174404  3$:     MOV     #4$,@CTVECT     ;SET XMIT VECTOR TO END OF TEST
2281  006402  052777  000100  174354          BIS     #BIT6,@CTCSR    ;ENABLE INTERRUPTS
```

```
2282  006410  000240                          NOP
2283
2284  006412  104075                          ERROR   75              ;XMIT DID NOT INTERRUPT
2285
2286  006414  042777  000100  174342  4$:     BIC     #BIT6,@CTCSR    ;DISABLE INTERRUPTS
2287  006422  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2288  006424  010377  174356                  MOV     R3,@CTVECT      ;RESTORE XMIT VECTOR
2289
2290
2291                                          ;;**********************************************************************
2292                                          ;*TEST 30        TEST SLU1 XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
2293                                          ;;**********************************************************************
2294  006430  000004                  TST30:  SCOPE
2295  006432  012737  000030  001104          MOV     #30,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2296  006440  042777  000100  174316          BIC     #BIT6,@CTCSR    ;DISABLE INTERRUPTS
2297  006446  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIOR.TY 7
2298  006452  000340                          .WORD   340
2299  006454  017703  174326                  MOV     @CTVECT,R3      ;SAVE XMIT VECTOR
2300  006460  012777  006506  174320          MOV     #2$,@CTVECT     ;POINT XMIT VECTOR TO ERROR REPORT
2301  006466  105777  174272          1$:     TSTB    @CTCSR          ;WAIT FOR DONE
2302  006472  100375                          BPL     1$
2303  006474  052777  000100  174262          BIS     #BIT6,@CTCSR    ;ENABLE INTERRUPT
2304  006502  000240                          NOP
2305  006504  000402                          BR      3$              ;CONTINUE TEST
2306
2307  006506  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2308  006510  104076                          ERROR   76
2309                                                                  ;XMIT INTERRUPTS AT PRIORITY=7
2310  006512  042777  000100  174244  3$:     BIC     #BIT6,@CTCSR    ;CLEAR INTERRUPT ENABLE
2311  006520  012777  006540  174260          MOV     #4$,@CTVECT     ;POINT XMIT VECTOR TO ERROR REPORT
2312  006526  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2313  006532  000140                          .WORD   140
2314  006534  000240                          NOP
2315  006536  000402                          BR      5$              ;BR TO END OF TEST-NO INTERRUPT
2316
2317  006540  022626                  4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2318  006542  104077                          ERROR   77
2319                                                                  ;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
2320  006544  010377  174236          5$:     MOV     R3,@CTVECT      ;RESTORE XMIT VECTOR
2321
2322
2323                                          ;;**********************************************************************
2324                                          ;*TEST 31        TEST SLU1 TRANSMITTER FOR DOUBLE INTERRUPTS
2325                                          ;;**********************************************************************
2326  006550  000004                  TST31:  SCOPE
2327  006552  012737  000031  001104          MOV     #31,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2328  006560  042777  000100  174176          BIC     #BIT6,@CTCSR    ;CLEAR INTERRUPT ENABLE
2329  006566  017703  174214                  MOV     @CTVECT,R3      ;SAVE XMIT VECTOR
2330  006572  017704  174212                  MOV     @CTPSW,R4       ;SAVE XMIT PSW VECTOR
2331  006576  012777  006640  174202          MOV     #2$,@CTVECT     ;SET UP XMIT VECTOR
2332  006604  012777  000340  174176          MOV     #340,@CTPSW     ;SET PIO 7 AFTER INTERRUPT
2333  006612  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2334  006616  000140                          .WORD   140
2335  006620  105777  174140          1$:     TSTB    @CTCSR          ;WAIT FOR DONE
2336  006624  100375                          BPL     1$
2337  006626  052777  000100  174130          BIS     #BIT6,@CTCSR    ;ENABLE INTERRUPTS
```

F4


```
2338  006634  000240                          NOP
2339
2340  006636  104100                          ERROR   100
2341                                                          ;XMIT INTERRUPT DID NOT OCCUR
2342  006640  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2343  006642  012777  006670  174136          MOV     #4$,@CTVECT     ;POINT XMIT VECTOR TO ERROR
2344  006650  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2345  006654  000140                          .WORD   140
2346  006656  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPTS
2347  006660  042777  000100  174076          BIC     #BIT6,@CTCSR    ;DISABLE INTERRUPTS
2348  006666  000402                          BR      5$              ;BR TO END OF TEST
2349
2350  006670  022626                  4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2351  006672  104101                          ERROR   101
2352                                                          ;XMIT RE-INTERRUPTED
2353  006674  010377  174106          5$:     MOV     R3,@CTVECT      ;RESTORE XMIT VECTOR
2354  006700  010477  174104                  MOV     R4,@CTPSW       ;RESTORE XMIT PSW VECTOR
2355
2356                                  ;;**********************************************************************
2357                                  ;*TEST 32        TEST THAT SLU1 XMIT INTERRUPT CLEARS WITH LOADING TBUF
2358                                  ;;**********************************************************************
2359  006704  000004                  TST32:  SCOPE
2360  006706  012737  000032  001104          MOV     #32,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2361  006714  042777  000100  174042          BIC     #BIT6,@CTCSR    ;DISABLE INTERRUPTS
2362  006722  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2363  006726  000340                          .WORD   340
2364  006730  017703  174052                  MOV     @CTVECT,R3      ;SAVE XMIT VECTOR
2365  006734  012777  007006  174044          MOV     #2$,@CTVECT     ;POINT XMIT VECTOR TO ERROR
2366  006742  052777  000100  174014          BIS     #BIT6,@CTCSR    ;ENABLE INTERRUPTS
2367  006750  005077  174012                  CLR     @CTBUF          ;LOAD TBUF
2368  006754  105777  174004          1$:     TSTB    @CTCSR          ;WAIT FOR DONE (INTERRUPT)
2369  006760  100375                          BPL     1$
2370  006762  005077  174000                  CLR     @CTBUF          ;FILL SECOND BUFFER TO RESET INT.
2371  006766  004737  012356                  JSR     PC,WRPSW        ;ALLOW INTERRUPTS
2372  006772  000140                          .WORD   140
2373  006774  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPTS
2374  006776  042777  000100  173760          BIC     #BIT6,@CTCSR    ;DISABLE INTERRUPTS
2375  007004  000402                          BR      3$              ;BR TO END OF TEST
2376
2377  007006  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2378  007010  104102                          ERROR   102
2379                                                          ;LOADING TBUF DID NOT CLEAR INTERRUPT.
2380  007012  010377  173770          3$:     MOV     R3,@CTVECT      ;RESTORE XMIT VECTOR
2381
2382
2383  007016                          SLU2IT:
2384
2385                                  ;;**********************************************************************
2386                                  ;*TEST 33        TEST THAT SLU2 XMIT INTERRUPTS ONLY WHEN ENABLED
2387                                  ;;**********************************************************************
2388  007016  000004                  TST33:  SCOPE
2389  007020  012737  000033  001104          MOV     #33,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2390  007026  032737  000002  001156          BIT     #BIT1,@#$DEVM   ;DO THESE TESTS FOR THIS DEVICE?
2391  007034  001002                          BNE     99$             ;F YES CONTINUE WITH TESTS
2392  007036  000137  010762                  JMP     LTCIT           ;IF NO GO TO START OF NEXT SET OF TESTS.
2393  007042                          99$:
```

```
2394  007042  042777  000100  173704          BIC     #BIT6,@TCSR     ;CLEAR TRANSMIT INTERRUPT ENABLE
2395  007050  017703  173722                  MOV     @TVECT,R3       ;SAVE XMIT VECTOR
2396  007054  012777  007100  173714          MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
2397  007062  105777  173666          1$:     TSTB    @TCSR           ;WAIT FOR DONE
2398  007066  100375                  BPL     1$
2399  007070  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2400  007074  000140                          .WORD   140
2401  007076  000402                  BR      3$
2402
2403  007100  022626          2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2404  007102  104116                  ERROR   116
2405                                          ;XMIT INTERRUPTS WITH INTERRUPT ENABLE CLEAR
2406  007104  012777  007124  173664  3$:     MOV     #4$,@TVECT      ;SET XMIT VECTOR TO END OF TEST
2407  007112  052777  000100  173634          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
2408  007120  000240                  NOP
2409
2410  007122  104117                  ERROR   117             ;XMIT DID NOT INTERRUPT
2411
2412  007124  042777  000100  173622  4$:     BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
2413  007132  022626                  CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2414  007134  010377  173636                  MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
2415
2416
2417                                  ;;*******************************************************************
2418                                  ;*TEST 34       TEST SLU2 XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
2419                                  ;;*******************************************************************
2420  007140  000004          TST34:  SCOPE
2421  007142  012737  000034  001104          MOV     #34,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2422  007150  042777  000100  173576          BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
2423  007156  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2424  007162  000340                          .WORD   340
2425  007164  017703  173606                  MOV     @TVECT,R3       ;SAVE XMIT VECTOR
2426  007170  012777  007216  173600          MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
2427  007176  105777  173552          1$:     TSTB    @TCSR           ;WAIT FOR DONE
2428  007202  100375                  BPL     1$
2429  007204  052777  000100  173542          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPT
2430  007212  000240                  NOP
2431  007214  000402                  BR      3$              ;CONTINUE TEST
2432
2433  007216  022626          2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2434  007220  104120                  ERROR   120
2435                                          ;XMIT INTERRUPTS AT PRIORITY=7
2436  007222  042777  000100  173524  3$:     BIC     #BIT6,@TCSR     ;CLEAR INTERRUPT ENABLE
2437  007230  012777  007250  173540          MOV     #4$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
2438  007236  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2439  007242  000140                          .WORD   140
2440  007244  000240                  NOP
2441  007246  000402                  BR      5$              ;BR TO END OF TEST-NO INTERRUPT
2442
2443  007250  022626          4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2444  007252  104121                  ERROR   121
2445                                          ;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
2446  007254  010377  173516          5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
2447
2448
2449
```

H4

SEQ 0046


```
                                ;;*****************************************************************
                                ;*TEST 35        TEST SLU2 TRANSMITTER FOR DOUBLE INTERRUPTS
                                ;;*****************************************************************
007260  000004          TST35:  SCOPE
007262  012737  000035  001104          MOV     #35,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
007270  042777  000100  173456          BIC     #BIT6,@TCSR     ;CLEAR INTERRUPT ENABLE
007276  017703  173474                  MOV     @TVECT,R3       ;SAVE XMIT VECTOR
007302  017704  173472                  MOV     @TPSW,R4        ;SAVE XMIT PSW VECTOR
007306  012777  007350  173462          MOV     #2$,@TVECT      ;SET UP XMIT VECTOR
007314  012777  000340  173456          MOV     #340,@TPSW      ;SET PIO 7 AFTER INTERRUPT
007322  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
007326  000140                          .WORD   140
007330  105777  173420          1$:     TSTB    @TCSR           ;WAIT FOR DONE
007334  100375                  BPL     1$
007336  052777  000100  173410          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
007344  000240                  NOP
007346  104122                  ERROR   122
                                                                ;XMIT INTERRUPT DID NOT OCCUR
007350  022626          2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
007352  012777  007400  173416          MOV     #4$,@TVECT      ;POINT XMIT VECTOR TO ERROR
007360  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
007364  000140                          .WORD   140
007366  000240                  NOP                             ;GIVE TIME FOR ANY INTERRUPTS
007370  042777  000100  173356          BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
007376  000402                  BR      5$              ;BR TO END OF TEST
007400  022626          4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
007402  104123                  ERRCR   123
                                                                ;XMIT RE-INTERRUPTED
007404  010377  173366          5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
007410  010477  173364                  MOV     R4,@TPSW        ;RESTORE XMIT PSW VECTOR

                                ;;*****************************************************************
                                ;*TEST 36        TEST THAT SLU2 XMIT INTERRUPT CLEARS WITH LOADING TBUF
                                ;;*****************************************************************
007414  000004          TST36:  SCOPE
007416  012737  000036  001104          MOV     #36,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
007424  032737  000001  001120          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
007432  001403                  BEQ     70$             ;IF NO THEN DO TEST
007434  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
007440  001046                  BNE     TST37           ;IF NO THEN SKIP TO NEXT TEST
007442                          70$:
007442  042777  000100  173304          BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
007450  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
007454  000340                          .WORD   340
007456  017703  173314                  MOV     @TVECT,R3       ;SAVE XMIT VECTOR
007462  012777  007534  173306          MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR
007470  052777  000100  173256          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
007476  005077  173254                  CLR     @TBUF           ;LOAD TBUF
007502  105777  173246          1$:     TSTB    @TCSR           ;WAIT FOR DONE (INTERRUPT)
007506  100375                  BPL     1$
007510  005077  173242                  CLR     @TBUF           ;FILL SECOND BUFFER TO RESET INT.
007514  004737  012356                  JSR     PC,WRPSW        ;ALLOW INTERRUPTS
007520  000140                          .WORD   140
```

```
2506  007522  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPTS
2507  007524  042777  000100  173222          BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
2508  007532  000402                          BR      3$              ;BR TO END OF TEST
2509
2510  007534  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2511  007536  104124                          ERROR   124
2512                                                                  ;LOADING TBUF DID NOT CLEAR INTERRUPT.
2513  007540  005001                  3$:     CLR     R1              ;INITIALIZE LOOP COUNTER
2514  007542  005201                  4$:     INC     R1              ;INCREMENT LOOP COUNTER
2515  007544  001376                          BNE     4$              ;UNTIL LOOP COUNTER EQUALS 0
2516  007546  005777  173200                  TST     @RBUF           ;CLEAR RECEIVER BUFFER
2517  007552  010377  173220                  MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
2518
2519
2520
2521
2522                                  ;;**********************************************************************
2523                                  ;*TEST 37        TEST THAT RCVR INTERRUPTS ONLY WHEN ENABLED
2524                                  ;;**********************************************************************
2525  007556  000004                  TST37:  SCOPE
2526  007560  012737  000037  001104          MOV     #37,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2527  007566  042777  000100  173160          BIC     #BIT6,@TCSR     ;DISABLE TRANSMIT INTERRUPTS
2528  007574  042777  000100  173146          BIC     #BIT6,@RCSR     ;DISABLE RECEIVER INTERRUPTS
2529  007602  017703  173164                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
2530  007606  012777  007636  173156          MOV     #2$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
2531  007614  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2532  007620  000140                                  .WORD   140
2533  007622  005077  173130                  CLR     @TBUF           ;SEND A CHARACTER
2534  007626  105777  173116          1$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
2535  007632  100375                          BPL     1$
2536  007634  000402                          BR      3$              ;CONTINUE TEST
2537
2538  007636  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2539  007640  104125                          ERROR   125
2540                                                                  ;RECEIVER INTERRPUTS WITH INT. ENABLE CLEAR
2541
2542  007642  012777  007662  173122  3$:     MOV     #4$,@RVECT      ;POINT RCV VECTOR TO END OF TEST
2543  007650  052777  000100  173072          BIS     #BIT6,@RCSR     ;ENABLE RCV INTERRUPTS
2544  007656  000240                          NOP                     ;GIVE ANY INTERRUPTS TIME
2545  007660  104126                          ERROR   126
2546                                                                  ;RCVR DID NOT INTERRUPT
2547
2548  007662  042777  000100  173060  4$:     BIC     #BIT6,@RCSR     ;DISABLE INTERRUPTS
2549  007670  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2550  007672  010377  173074                  MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
2551
2552
2553                                  ;;**********************************************************************
2554                                  ;*TEST 40        TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
2555                                  ;;**********************************************************************
2556  007676  000004                  TST40:  SCOPE
2557  007700  012737  000040  001104          MOV     #40,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2558  007706  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2559  007712  000340                                  .WORD   340
2560  007714  017703  173052                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
2561  007720  012777  007752  173044          MOV     #2$,@RVECT      ;POINT RCVR VECTOR TO ERROR REPORT
```

J4

SEQ 0048


```
2562  007726  005077  173024            CLR     @TBUF           ;SEND A CHARACTER
2563  007732  105777  173012  1$:       TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
2564  007736  100375                    BPL     1$
2565  007740  052777  000100  173002    BIS     #BIT6,@RCSR     ;ENABLE INTERRUPTS
2566  007746  000240                    NOP                     ;GIVE TIME FOR INTERRUPT
2567  007750  000402                    BR      3$              ;CONTINUE TEST
2568  007752  022626          2$:       CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2569  007754  104127                    ERROR   127
2570                                                            ;RCVR INTERRUPTS AT PRIORITY 7
2571
2572  007756  042777  000100  172764  3$:       BIC     #BIT6,@RCSR     ;CLEAR INTERRUPT ENABLE
2573  007764  012777  010004  173000            MOV     #4$,@RVECT      ;POINT RCVR VECTOR TO ERROR REPORT
2574  007772  004737  012356                    JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2575  007776  000140                            .WORD   140
2576  010000  000240                    NOP                     ;GIVE TIME FOR ANY INTERRUPT
2577  010002  000402                    BR      5$              ;BR TO END OF TEST, IF NO INTERRUPT
2578
2579  010004  022626          4$:       CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2580  010006  104130                    ERROR   130
2581                                                            ;RCVR INTERRUPT REQUEST PASSED WITH BIT6 CLEAR
2582  010010  010377  172756  5$:       MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
2583
2584
2585                          ;;*****************************************************************
2586                          ;*TEST 41       TEST SLU2 RECEIVER FOR DOUBLE INTERRUPTS
2587                          ;;*****************************************************************
2588  010014  000004  TST41:  SCOPE
2589  010016  012737  000041  001104            MOV     #41,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2590  010024  017703  172742                    MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
2591  010030  017704  172740                    MOV     @RPSW,R4        ;SAVE RECEIVE PSW VECTOR
2592  010034  012777  010102  172730            MOV     #2$,@RVECT      ;POINT RCV VECTOR TO CONTINUE TEST
2593  010042  012777  000340  172724            MOV     #340,@RPSW      ;SET PRIORITY TO 7 AFTER INTERRUPT
2594  010050  004737  012356                    JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2595  010054  000140                            .WORD   140
2596  010056  005077  172674                    CLR     @TBUF           ,SEND A CHARACTER
2597  010062  105777  172662          1$:       TSTB    @RCSR           ;WAIT FOR RCVR DONE
2598  010066  100375                    BPL     1$
2599  010070  052777  000100  172652    BIS     #BIT6,@RCSR     ;ENABLE RCV INTERRUPTS
2600  010076  000240                    NOP                     ;GIVE SOME TIME
2601
2602  010100  104131                    ERROR   131
2603                                                            ;RCVR INTERRUPT DID NOT OCCUR
2604
2605  010102  022626          2$:       CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2606  010104  012777  010136  172660    MOV     #3$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
2607  010112  004737  012356            JSR     PC,WRPSW        ;RESET PSW TO PRIORITY 3
2608  010116  000140                    .WORD   140
2609  010120  000240                    NOP                     ;GIVE SOME TIME
2610  010122  042777  000100  172620    BIC     #BIT6,@RCSR     ;CLEAR INTERRUPT ENABLE
2611  010130  010477  172640            MOV     R4,@RPSW        ;RESTORE RECEIVE PSW VECTOR
2612  010134  000402                    BR      4$              ;BR TO END OF TEST
2613
2614  010136  022626          3$:       CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2615  010140  104132                    ERROR   132
2616                                                            ;RECEIVER RE-INTERRUPTED
2617  010142  010377  172624  4$:       MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
```

K4


```
                                        ;;*****************************************************************
                                        ;*TEST 42       TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
                                        ;;*****************************************************************
010146  000004                  TST42:  SCOPE
010150  012737  000042  001104          MOV     #42,$TESTN      ;;SET TEST NUMBER IN APT MAIL BCX
010156  004737  012356                  JSR     PC,WRPSW        ;SET PSW PRIORITY TO 7
010162  000340                          .WORD   340
010164  017703  172602                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
010170  012777  010242  172574          MOV     #2$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
010176  052777  000100  172544          BIS     #BIT6,@RCSR     ;SET RCVR INTERRUPT ENABLE
010204  005077  172546                  CLR     @TBUF           ;SEND A CHARACTER
010210  105777  172534          1$:     TSTB    @RCSR           ;WAIT FOR DONE (INTERRUPT)
010214  100375                          BPL     1$
010216  005777  172530                  TST     @RBUF           ;READ RBUF TO CLEAR PENDING INTERRPUT
010222  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
010226  000140                          .WORD   140
010230  000240                          NOP                     ;ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
010232  042777  000100  172510          BIC     #BIT6,@RCSR     ;NO INTERRUPT-CLEAR INT. ENABLE
010240  000402                          BR      3$
                                
010242  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
010244  104133                          ERROR   133             ;READING RBUF DID NOT CLEAR INTERRUPT
                                
010246  010377  172520          3$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
                                
                                
                                
                                        ;;*****************************************************************
                                        ;*TEST 43       TEST SLU2 THAT RESET CLEARS RECEIVE INTERRUPT
                                        ;;*****************************************************************
010252  000004                  TST43:  SCOPE
010254  012737  000043  001104          MOV     #43,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
010262  032737  000001  001120          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
010270  001403                          BEQ     70$             ;IF NO THEN DO TEST
010272  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
010276  001037                          BNE     TST44           ;IF NO THEN SKIP TO NEXT TEST
010300                          70$:
010300  000005                          RESET                   ;CLEAR EVERYTHING
010302  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
010306  000340                          .WORD   340
010310  017703  172456                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
010314  012777  010366  172450          MOV     #2$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
010322  052777  000100  172420          BIS     #BIT6,@RCSR     ;SET RCV INTERRUPT ENABLE
010330  012777  000377  172420          MOV     #377,@TBUF      ;SEND AN ALL 1'S CHARACTER
010336  105777  172406          1$:     TSTB    @RCSR           ;WAIT FOR RCV DONE
010342  100375                          BPL     1$
010344  000005                          RESET                   ;CLEAR RCV INTERRUPT & RBUF
010346  004737  012356                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
010352  000140                          .WORD   140
010354  000240                          NOP                     ;ALLOW TIME FOR AN ERRONEOUS INTERRUPT
010356  042777  000100  172364          BIC     #BIT6,@RCSR     ;NO INTERRUPT-CLEAR INT. ENABLE
010364  000402                          BR      3$              ;CONTINUE TEST
```

```
                                2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
010366  022626                  
010370  104134                          ERROR   134             ;RESET DID NOT CLEAR RCVR INTERRUPT

010372  010377  172374          3$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR


                                ;;**********************************************************************
                                ;*TEST 44        TEST SLU2 THAT "OVERRUN & ERROR" BITS CAN BE SET
                                ;;**********************************************************************
010376  000004                  TST44:  SCOPE
010400  012737  000044  001104          MOV     #44,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
010406  032777  002000  170424          BIT     #BIT10,@SWR     ;IS THIS TEST DISABLED
010414  001023                        BNE     TST45           ;IF NOT ENABLED, BR TO NEXT TEST
010416  012700  000003                MOV     #3,R0           ;SET CHARACTER COUNT TO SEND 3 CHAR.
010422  005077  172330          1$:     CLR     @TBUF           ;LOAD TRANSMIT BUFFER
010426  105777  172322          2$:     TSTB    @TCSR           ;WAIT FOR TRANSMIT DONE
010432  100375                        BPL     2$
010434  005300                        DEC     R0              ;DECREMENT CHARACTER COUNT
010436  001371                        BNE     1$              ;BR IF ALL CHARACTERS NOT TRANSMITTED
010440  032777  040000  172304        BIT     #BIT14,@RBUF    ;TEST FOR "OR" ERROR FLAG
010446  001001                        BNE     3$              ;BR, IF SET
010450  104135                        ERROR   135
                                                                ;"OR" ERROR FLAG DID NOT SET

010452  032777  100000  172272  3$:     BIT     #BIT15,@RBUF    ;TEST "ERROR" FLAG
010460  001001                        BNE     4$              ;BR, IF SET
010462  104136                        ERROR   136
                                                                ;"ERROR" FLAG DID NOT SET WITH "OR" FLAG
010464                          4$:


                                ;;**********************************************************************
                                ;*TEST 45        TEST THAT BREAK TRANSMITS ALL ZEROES
                                ;;**********************************************************************
010464  000004                  TST45:  SCOPE
010466  012737  000045  001104          MOV     #45,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
010474  012777  177777  172254          MOV     #-1,@TBUF       ;TRANSMIT ALL ONES TO RCVR
010502  105777  172242          1$:     TSTB    @RCSR           ;WAIT FOR RCVR DONE
010506  100375                        BPL     1$
010510  005777  172236                TST     @RBUF           ;CLEAR DONE (LEAVING ALL ONES IN RBUF)
010514  052777  000001  172232        BIS     #BIT0,@TCSR     ;TRANSMIT BREAK
010522  005000                        CLR     R0              ;CLEAR A TIMER
010524  105777  172220          2$:     TSTB    @RCSR           ;WAIT FOR RCVR DONE
010530  100406                        BMI     CONT41          ;BR IF DONE
010532  005200                        INC     R0              ;IF NOT, INCREMENT TIMER
010534  001373                        BNE     2$              ;BR IF TIME REMAINS

010536  042777  000001  172210        BIC     #BIT0,@TCSR       ;CLEAR BREAK BITS
010544  104137                        ERROR   137             ;BREAK DID NOT TRANSMIT ANYTHING

010546  105777  172200          CONT41: TSTB    @RBUF           ;CHECK RECEIVE BUFFER FOR ZERO
010552  001404                        BEQ     3$              ;BR, IF ZERO
010554  042777  000001  172172        BIC     #BIT0,@TCSR       ;CLEAR BREAK BITS

010562  104137                        ERROR   137             ;BREAK DID NOT TRANSMIT ALL ZEROES
```

```
2730  010564  042777  000001  172162  3$:     BIC     #BIT0,@TCSR      ;CLEAR BREAK BITS
2731
2732
2733                                    ;;*****************************************************************
2734                                    ;*TEST 46       TEST THAT "FR" ERROR CAN BE SET DURING BREAK
2735                                    ;;*****************************************************************
2736  010572  000004                    TST46:  SCOPE
2737  010574  012737  000046  001104            MOV     #46,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
2738  010602  032777  002000  170230            BIT     #BIT10,@SWR      ;IS THIS TEST DISABLED
2739  010610  001025                            BNE     TST47            ;BR TO NEXT TEST, IF DISABLED
2740  010612  052777  000001  172134            BIS     #BIT0,@TCSR      ;SEND BREAK
2741  010620  005077  172132                    CLR     @TBUF            ;TRANSMIT A CHARACTER TO TIME BREAK
2742  010624  105777  172120          1$:       TSTB    @RCSR            ;WAIT FOR RCVR DONE
2743  010630  100375                            BPL     1$
2744  010632  042777  000001  172114            BIC     #BIT0,@TCSR      ;CLEAR BREAK BITS
2745  010640  032777  020000  172104            BIT     #BIT13,@RBUF     ;CHECK FOR FRAMING ERROR FLAG
2746  010646  001001                            BNE     2$               ;BR, IF SET
2747
2748  010650  104140                            ERROR   140
2749                                                                     ;BREAK DID NOT SET FRAMING ERROR
2750  010652  032777  100000  172072  2$:       BIT     #BIT15,@RBUF     ;TEST "ERROR"FLAG
2751  010660  001001                            BNE     3$               ;BR, IF SET
2752
2753  010662  104141                            ERROR   141
2754                                                                     ;"ERROR" FLAG DID NOT SET WITH "OR" FLAG
2755  010664                          3$:
2756                                    ;;*****************************************************************
2757                                    ;*TEST 47       TEST DATA PATHS USING WRAP CABLE
2758                                    ;;*****************************************************************
2759  010664  000004                    TST47:  SCOPE
2760  010666  012737  000047  001104            MOV     #47,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
2761  010674  032777  000200  170136            BIT     #BIT7,@SWR       ;IS THIS TEST ENABLED
2762  010702  001027                            BNE     TST50            ;BR, IF NOT
2763  010704  005001                            CLR     R1               ;CLEAR REGISTER FOR TEST DATA
2764                                                                     ;TRANSMIT A BINARY COUNT PATTERN - UP
2765                                                                     ;TO THE BIT POSITION INDICATED BY THE
2766                                                                     ;CONTENTS OF LOCATION "$USWR"
2767  010706  105201                  1$:       INCB    R1               ;INCREMENT THE TEST DATA
2768  010710  010177  172042                    MOV     R1,@TBUF         ;XMIT A CHARACTER
2769  010714  005000                            CLR     R0               ;CLEAR A TIMER
2770  010716  105777  172026          2$:       TSTB    @RCSR            ;WAIT FOR RECEIVER DONE
2771  010722  100403                            BMI     3$               ;BR IF DONE
2772  010724  005200                            INC     R0               ;INCREMENT TIMER IF NOT
2773  010726  001373                            BNE     2$               ;BR IF TIME REMAINS
2774
2775  010730  104056                            ERROR   56               ;RECEIVER DONE NOT SET
2776
2777  010732  017702  172014          3$:       MOV     @RBUF,R2         ;GET RECEIVED CHARACTER
2778  010736  020102                            CMP     R1,R2            ;COMPARE DATA
2779  010740  001003                            BNE     4$               ;BR, IF NON-COMPARE
2780  010742  105701                            TSTB    R1               ;TEST XMIT DATA FOR ZERO
2781  010744  001406                            BEQ     TST50            ;BR, IF FINISHED
2782  010746  000757                            BR      1$               ;CONTINUE IF NOT
2783  010750  010137  001024          4$:       MOV     R1,$GDDAT        ;STORE EXPECTED DATA
2784  010754  010237  001026                    MOV     R2,$BDDAT        ;STORE RECEIVED DATA
2785
```

```
                                ERROR   142             ;DATA COMPARE ERROR WITH WRAP CABLE


                        LTCIT:

                        ;;*******************************************************************
                        ;*TEST 50       TEST THAT THE REAL TIME CLOCK INTERRUPTS PROPERLY
                        ;;*******************************************************************
                        TST50:  SCOPE
                                MOV     #50,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                BIT     #BIT2,@#$DEVM   ;DO THESE TESTS FOR THIS DEVICE?
                                BNE     99$             ;F YES CONTINUE WITH TESTS
                                JMP     BLAST           ;IF NO GO TO START OF NEXT SET OF TESTS.
                        99$:
                                CLR     R0
                                JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
                                .WORD   340
                                MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
                                MOV     @RTCPSW,R4      ;SAVE LINE CLOCK PSW VECTOR
                                MOV     #2$,@RTCVT      ;SET RTC INTERRUPT VECTOR TO ERROR REPORT
                                MOV     #340,@RTCPSW    ;KEEP PRIORITY AT 7

                                BIS     #BIT6,@LKS      ;SET INTERRUPT ENABLE
                        1$:     INC     R0
                                BNE     1$
                                NOP                     ;GIVE TIME FOR ANY INTERRUPTS
                                BR      3$              ;BR, IF NO INTERRUPT OCCURS

                        2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
                                ERROR   143             ;RTC INTERRUPTS AT PRIORITY 7

                        3$:     CLR     @LKS    ;DISABLE RTC INTERRUPTS
                                CLR     R0
                                MOV     #XXX,@RTCVT     ;SET RTC INTERRUPT VECTOR FOR ERROR
                                JSR     PC,WRPSW        ;CHANGE PSW TO PRIORITY 5
                                .WORD   240
                                ERROR   146
                        XXX:    MOV     #4$,@RTCVT
                                MTPS    #240
                        20$:    INC     R0
                                TST     R0
                                BPL     20$
                                NOP                     ;GIVE TIME FOR ANY INTERRUPT
                                BR      5$              ;IF NO INTERRUPT - BR TO CONTINUE TEST

                        4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
                                ERROR   144             ;RTC INTERRUPTS WITH INTERRUPTS DISABLED

                        5$:     MOV     #7$,@RTCVT      ;POINT RTC VECTOR TO END OF TEST
                                CLR     R0
                                BIS     #BIT6,@LKS      ;ALLOW INTERRUPTS
                        6$:     INC     R0
                                TST     R0
                                BPL     6$
                                NOP                     ;GIVE TIME FOR INTERRUPT
```

```
                                        ERROR   145             ;RTC INTERRUPT DID NOT OCCUR

                                7$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
                                        BIC     #BIT6,@LKS      ;DISABLE INTERRUPTS
                                        MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
                                        MOV     R4,@RTCPSW      ;RESTORE LINE CLOCK PSW VECTOR


                                ;;**********************************************************************
                                ;*TEST 51        TEST RTC FOR DOUBLE INTERRUPTS
                                ;;**********************************************************************
                                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     @#$PASS         ;IS THIS FIRST PASS
                                        BNE     TST52           ;IF NO THEN SKIP TO NEXT TEST
                                70$:
                                        MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
                                        MOV     @RTCPSW,R4      ;SAVE LINE CLOCK PSW VECTOR
                                        MOV     #2$,@RTCVT      ;SET UP RTC INTERRUPT VECTOR
                                        MOV     #340,@RTCPSW    ;DISALLOW INTERRUPTS AFTER THE INTERRUPT
                                        JSR     PC,WRPSW        ;SET PRIORITY TO 5
                                                .WORD   240

                                        CLR     R0
                                        BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
                                1$:     INC     R0
                                        TST     R0
                                        BPL     1$
                                        NOP                     ;GIVE TIME FOR ANY INTERRUPT

                                        ERROR   146             ;RTC INTERRUPT DID NOT OCCUR

                                2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
                                        MOV     #3$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
                                        JSR     PC,WRPSW        ;SET PSW TO PRIORITY 5
                                                .WORD   240
                                        NOP                     ;GIVE SOME TIME FOR AN INTERRUPT
                                        BR      4$              ;NO INTERRUPT - BR TO END OF TEST

                                3$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
                                        ERROR   147             ;INTERRUPT SEQUENCE DID NOT CLEAR
                                                                ;INTERRUPT REQUEST

                                4$:     BIC     #BIT6,@LKS      ;DISABLE CLOCK INTERRUPTS
                                        MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
                                        MOV     R4,@RTCPSW      ;RESTORE LINE CLOCK PSW VECTOR


                                ;;**********************************************************************
                                ;*TEST 52        TEST THAT RTC INTERRUPT CLEARS WITH RESET
                                ;;**********************************************************************
                                TST52:  SCOPE
                                        MOV     #52,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
```

```
2898  011366  032737  000001  001120          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
2899  011374  001403                          BEQ     70$             ;IF NO THEN DO TEST
2900  011376  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
2901  011402  001034                          BNE     TST53           ;IF NO THEN SKIP TO NEXT TEST
2902  011404                          70$:
2903  011404  004737  012356                  JSR     PC,WRPSW        ;SET PRIORITY TO 7
2904  011410  000340                          .WORD   340
2905  011412  017703  171374                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
2906  011416  012777  011464  171366          MOV     #2$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
2907
2908  011424  005000                          CLR     R0
2909  011426  052777  000100  171334          BIS     #BIT6,@LKS      ,ENABLE CLOCK INTERRUPTS
2910  011434  005200                  1$:     INC     R0
2911  011436  005700                          TST     R0
2912  011440  100375                          BPL     1$
2913  011442  000005                          RESET                   ;CLEAR PENDING INTERRUPT WITH RESET
2914  011444  004737  012356                  JSR     PC,WRPSW        ;SET PRIORITY TO 5
2915  011450  000240                          .WORD   240
2916  011452  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPT
2917  011454  042777  000100  171306          BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
2918  011462  000402                          BR      3$              ;BR TO END OF TEST
2919
2920  011464  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2921  011466  104150                          ERROR   150             ;RESET DID NOT CLEAR INTERRUPT
2922
2923  011470  010377  171316          3$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
2924
2925  011474                          BLAST:
2926
2927                                  ;;**********************************************************************
2928                                  ;*TEST 53       TEST ALL INTERNAL OPTIONS SIMULTANEOUSLY
2929                                  ;;**********************************************************************
2930  011474  000004                  TST53:  SCOPE
2931  011476  012737  000053  001104          MOV     #53,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2932  011504  032737  000001  001120          BIT     #BIT0,@#$ENV    ;ARE WE RUNNING UNDER APT
2933  011512  001405                          BEQ     70$             ;IF NO DO TEST
2934  011514  005737  001106                  TST     @#$PASS         ;IS THIS FIRST PASS
2935  011520  001402                          BEQ     70$             ;IF YES DO TEST
2936  011522  000137  012240                  JMP     $EOP            ;IF NO DO NOT DO TEST
2937  011526  000005                  70$:    RESET                   ;CLEAR EVERY BODY
2938  011530  012737  000340  177776          MOV     #340,PS         ;SET PROCESSOR PRIORITY TO 7
2939  011536  017737  171234  001060          MOV     @TVECT,$TMP0
2940  011544  017737  171222  001062          MOV     @RVECT,$TMP1
2941  011552  017737  171230  001064          MOV     @CTVECT,$TMP2
2942  011560  017737  171216  001066          MOV     @CRVECT,$TMP3
2943  011566  017737  171220  001070          MOV     @RTCVT,$TMP4
2944  011574  005037  012232                  CLR     XMTCT2
2945  011600  005037  012234                  CLR     RECCT2
2946  011604  005037  012236                  CLR     COUNT
2947
2948  011610  012777  011726  171160          MOV     #XMIT2, @TVECT  ;SET UP SLU2 TRANSMIT VECTOR
2949  011616  012777  000340  171154          MOV     #340, @TPSW     ;AND PSW
2950  011624  012777  011762  171140          MOV     #REC2, @RVECT   ;SET UP SLU2 RECEIVER VECTOR
2951  011632  012777  000340  171134          MOV     #340, @RPSW     ;AND PSW
2952  011640  012777  011716  171144          MOV     #TICKER, @RTCVT ;SET UP RTC VECTOR
2953  011646  012777  000340  171140          MOV     #340, @RTCPSW   ;AND PSW
```

```
2954
2955  011654  052777  000100  171072          BIS     #BIT6, @TCSR    ;ENABLE SLU2 XMIT INTERRUPT
2956  011662  052777  000100  171060          BIS     #BIT6, @RCSR    ;ENABLE SLU2 RECEIVER INTERRUPT
2957  011670  012703  026320                  MOV     #BUF2, R3       ;SET UP RECEIVER BUFFER
2958
2959  011674  052777  000100  171066          BIS     #BIT6, @LKS     ;ENABLE RTC INTERRUPTS
2960  011702  012701  177777          3$:     MOV     #-1, R1         ;INITIALIZE DATA FOR SLU2(1ST CHR. 0)
2961  011706  005037  177776                  CLR     PS              ;DROP PROCESSOR PRIORITY TO ZERO
2962  011712  000001                  WAITIO: WAIT                    ;WAIT FOR INTERRUPT
2963  011714  000776                          BR      WAITIO
2964
2965  011716  005237  012236          TICKER: INC     COUNT           ;UPDATE COUNT
2966  011722  000137  012036                  JMP     IOHAND          ;GO TO INTERRUPT HANDLER
2967
2968
2969  011726  005237  012232          XMIT2:  INC     XMTCT2          ;UPDATE XMIT INTERRUPT COUNT
2970  011732  005201                          INC     R1              ;UPDATE XMIT DATA
2971  011734  010177  171016                  MOV     R1, @TBUF       ;SEND NEXT CHARACTER
2972  011740  023727  012232  000400          CMP     XMTCT2, #400    ;IF 256 CHARACTERS HAVE NOT
2973  011746  002403                          BLT     1$              ;BEEN TRANSFERRED CONTINUE
2974  011750  042777  000100  170776          BIC     #BIT6, @TCSR    ;ELSE NO MORE XMIT INTERRUPTS
2975  011756  000137  012036          1$:     JMP     IOHAND          ;GO TO INTERRUPT HANDLER
2976
2977  011762  005237  012234          REC2:   INC     RECCT2          ;UPDATE RECEIVER INTERRUPT COUNT
2978  011766  005777  170760                  TST     @RBUF           ;BIT 15 SETS IF ANY ERRORS OCCURRED
2979  011772  100017                          BPL     3$              ;IF BIT IS CLEAR NO ERRORS
2980  011774  017737  170752  001026          MOV     @RBUF, $BDDAT   ;GET ERROR INFORMATION
2981  012002  000005                          RESET                   ;CLEAR THE WORLD - STOP ALL
2982                                                                  ;INTERRUPTS
2983  012004  020327  026320                  CMP     R3, #BUF2       ;WAS MORE THAN 1 WORD TRANSFERRED
2984  012010  003004                          BGT     1$              ;IF YES GET LAST GOOD DATA
2985  012012  012737  177777  001024          MOV     #-1, $GDDAT     ;IF NO MAKE GOOD DATA -1
2986  012020  000403                          BR      2$              ;AND GET TO ERROR REPORT
2987  012022  116337  177777  001024  1$:     MOVB    -1(R3), $GDDAT  ;GET LAST GOOD DATA RECEIVED
2988  012030  104154                  2$:     ERROR   154             ;RECEIVER STATUS ERROR
2989  012032  117723  170714          3$:     MOVB    @RBUF, (R3)+    ;GET DATA AND STORE IT
2990
2991  012036  023727  012236  000074  IOHAND: CMP     COUNT, #74      ;HAS 1 SEC ELAPSED
2992  012044  002412                          BLT     3$              ;IF NO CONTINUE TEST
2993  012046  042777  000100  170710          BIC     #BIT6, @CTCSR   ;IF YES STOP TRANSMISSIONS
2994  012054  042777  000100  170672          BIC     #BIT6, @TCSR    ;
2995  012062  042777  000100  170700          BIC     #BIT6,  @LKS    ;TURN OFF LINE CLOCK
2996  012070  000401                          BR      WAITER
2997  012072  000002                  3$:     RTI                     ;RETURN FROM INTERRUPT TO AWAIT NEXT
2998
2999  012074  005037  177776          WAITER: CLR     PS              ;MAKE PROCESSOR PRIORITY 0
3000  012100  012705  140000                  MOV     #-40000, R5     ;SET UP LOOP COUNTER
3001  012104  062705  000001          1$:     ADD     #1, R5          ;DO LOOP UNTIL R5 = 0
3002  012110  001375                          BNE     1$
3003  012112  000005                          RESET                   ;STOP EVERYONE SHOULD BE DONE
3004  012114  012706  001000                  MOV     #1000,SP        ;RESET STACK AFTER LAST INTERRUPT
3005
3006
3007  012120  023737  012232  012234  CHECK2: CMP     XMTCT2, RECCT2  ;#OF XMIT INTERRUPTS = REC INTERRUPTS
3008  012126  001401                          BEQ     1$              ;IF YES CHECK DATA
3009  012130  104157                          ERROR   157             ;INTERRUPT COMPARISON ERROR
```

```
3010  012132  012703  026320          1$:     MOV     #BUF2, R3       ;INITIALIZE TO FIRST RECEIVED DATA
3011  012136  005001                          CLR     R1              ;INITIALIZE TO FIRST XMIT DATA
3012  012140  013704  012232                  MOV     XMTCT2, R4      ;GET # OF BYTES TRANSFERRED
3013  012144  122301                  2$:     CMPB    (R3)+, R1       ;IS RECEIVED DATA = EXPECTED DATA
3014  012146  001406                          BEQ     3$              ;IF YES CONTINUE TESTING
3015  012150  114337  001026                  MOVB    -(R3), $BDDAT   ;IF NO GET ERROR INFORMATION
3016  012154  010137  001024                  MOV     R1, $GDDAT
3017  012160  104160                          ERROR   160             ;SLU2 DATA COMPARISON ERROR
3018  012162  005203                          INC     R3              ;IF COONTINUE ON ERROR RESET POINTER
3019  012164  005201                  3$:     INC     R1              ;UPDATE TO NEXT GOOD DATA
3020  012166  077412                          SOB     R4,2$           ;LOOP UNTIL ALL DATA CHECKED
3021  012170  013777  001060  170600  FINIE:  MOV     $TMP0, @TVECT   ;RESTORE VECTORS
3022  012176  013777  001062  170566          MOV     $TMP1, @RVECT
3023  012204  013777  001064  170574          MOV     $TMP2, @CTVECT
3024  012212  013777  001066  170562          MOV     $TMP3, @CRVECT
3025  012220  013777  001070  170564          MOV     $TMP4, @RTCVT
3026  012226  000137  012240                  JMP     $EOP            ;FINISHED TESTING GO TO END OF PASS
3027
3028  012232  000000                  XMTCT2: .WORD   0
3029  012234  000000                  RECCT2: .WORD   0
3030  012236  000000                  COUNT:  .WORD   0
3031
3032
3033                                  .SBTTL  END OF PASS ROUTINE
3034
3035                                  ;;*****************************************************************
3036                                  ;*INCREMENT THE PASS NUMBER ($PASS)
3037                                  ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
303A                                  ;*IF THERES A MONITOR GO TO IT
303B                                  ;*IF THERE ISN'T JUMP TO TST1
3040
3041  012240                  $EOP:
3042  012240  000004                          SCOPE
3043  012242  005037  001002                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
3044  012246  005237  001106                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
3045  012252  042737  100000  001106          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
3046  012260  005327                          DEC     (PC)+           ;;LOOP?
3047  012262  000001          $EOPCT: .WORD   1
3048  012264  003022                          BGT     $DOAGN          ;;YES
3049  012266  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
3050  012270  000001          $ENDCT: .WORD   1
3051  012272  012262                          $EOPCT
3052  012274  104401  012341                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
3053  012300  013746  001106                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
3054  012304  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3055  012306  104401  012336                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
3056  012312  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
3057  012316  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
3058  012320  000005                          RESET                   ;;CLEAR THE WORLD
3059  012322  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
3060  012324  000240                          NOP                     ;;SAVE ROOM
3061  012326  000240                          NOP                     ;;FOR
3062  012330  000240                          NOP                     ;;ACT11
3063  012332                  $DOAGN:
3064  012332  000137                          JMP     @(PC)+          ;;RETURN
3065  012334  003430          $RTNAD: .WORD   TST1
```

```
 3066  012336     377     377     000  $ENULL: .BYTE    1,-1,0          ;;NULL CHARACTER STRING
 3067  012341     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
 3068  012346  050040  051501  020123
 3069  012354  000043
 3070
 3071
 3072
 3073  012356  011646                  WRPSW:  MOV(SP),-(SP)            ;COPY RETUN PC
 3074  012360  013616                          MOV     @(SP)+,(SP)      ;MOVE NEW PSW TO STACK
 3075  012362  062746  000002                  ADD     #2,-(SP)         ;ADJUST JSR RETURN
 3076  012366  000002                          RTI                      ;POP RETURN PC & NEW PSW
 3077
 3078                                  ;SUBROUTINE TO REPORT UNEXPECTED OR ERRONEOUS TRAPS OR INTERRUPTS
 3079
 3080  012370  012600                  CATCH:  MOV     (SP)+,R0         ;GET ADDRESS OF TRAP VECTOR + 4
 3081  012372  162700  000004                  SUB     #4,R0            ;ADJUST TO POINT TO TRAP ADDRESS
 3082  012376  010037  012416                  MOV     R0,BDVECT        ;STORE TRAP OR INTERRUPT ADDRESS
 3083  012402  016637  000002  012414          MOV     2(SP),OLDPC      ;GET PC WHERE TRAP OR INTERRUPT OCCURRED
 3084  012410  104161                          ERROR   161              ;REPORT ERROR
 3085
 3086  012412  000000                          HALT                     ;PROGRAM MUST BE RESTARTED AT THIS POINT
 3087  012414  000000                  OLDPC:  .WORD   0
 3088  012416  000000                  BDVECT: .WORD   0
 3089
 3090
 3091                                  ;;*******************************************************************
 3092                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 3093                                  ;*SAVE THE ERROR ITEM NUMBER AND ADDRESS OF THE ERROR CALL
 3094                                  ;*AND GO TO $ERRTYP ON ERROR
 3095                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 3096                                  ;*SW15=1         HALT ON ERROR
 3097                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
 3098                                  ;*SW09=1         LOOP IN ERROR
 3099                                  ;*CALL
 3100                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 3101                                  ;;*******************************************************************
 3102
 3103  012420                  $ERROR:
 3104  012420  105237  001003  7$:     INCB    $ERFLG           ;SET THE ERROR FLAG
 3105  012424  001775                  BEQ     7$               ;DON'T LET FLAG GO TO ZERO
 3106  012426  013777  001002  166406  MOV     $TSTNM,@DISPLAY  ;DISPLAY TEST NUMBER AND ERROR FLAG
 3107  012434  005237  001012          INC     $ERTTL           ;INCREMENT ERROR COUNT
 3108  012440  011637  001016          MOV     (SP),$ERRPC      ;GET ADDRESS OF ERROR INSTRUCTION
 3109  012444  162737  000002  001016  SUB     #2,$ERRPC
 3110  012452  117737  166340  001014  MOVB    @$ERRPC,$ITEMB   ;STRIP AND SAVE THE ERROR ITEM CODE
 3111  012460  032777  020000  166352  BIT     #BIT13,@SWR      ;SKIP TYPEOUT IF SET
 3112  012466  001004                  BNE     20$              ;SKIP TYPEOUTS
 3113  012470  004737  012602          JSR     PC,$ERRTYP       ;GO TO USER ERROR ROUTINE
 3114  012474  104401  001075          TYPE    ,$CRLF
 3115  012500                  20$:
 3116  012500  122737  000001  001120  CMPB    #APTENV,$ENV     ;RUNNING IN APT MODE
 3117  012506  001007                  BNE     2$               ;NO, SKIP APT ERROR REPORT
 3118  012510  113737  001014  012522  MOVB    $ITEMB,21$       ;SET ITEM NUMBER AS ERROR NUMBER
 3119  012516  004737  013304          JSR     PC,$ATY4         ;REPORT FATAL ERROR TO APT
 3120  012522     000                  21$:    .BYTE   0
 3121  012523     000                  .BYTE   0
```

```
 3122  012524  000777                  22$:    BR      22$             ;APT ERROR LOOP
 3123  012526  005777  166306          2$:     TST     @SWR            ;HALT ON ERROR
 3124  012532  100001                          BPL     3$              ;SKIP IF CONTINUE
 3125  012534  000000                          HALT                    ;HALT ON ERROR!
 3126  012536  104407                  3$:     CKSWR                   ;TEST FOR CHANGE IN SOFT-SWR
 3127  012540  032777  001000  166272          BIT     #BIT09,@SWR     ;LOOP ON ERROR SWITCH SET?
 3128  012546  001402                          BEQ     4$              ;BR IF NO
 3129  012550  013716  001010                  MOV     $LPERR,(SP)     ;FUDGE RETURN FOR LOOPING
 3130  012554  005737  001072          4$:     TST     $ESCAPE         ;CHECK FOR AN ESCAPE ADDRESS
 3131  012560  001402                          BEQ     5$              ;BR IF NONE
 3132  012562  013716  001072                  MOV     $ESCAPE,(SP)    ;FUDGE RETURN ADDRESS FOR ESCAPE
 3133  012566                          5$:
 3134  012566  022737  012322  000042          CMP     #$ENDAD,@#42    ;ACT-11 AUTO-ACCEPT?
 3135  012574  001001                          BNE     6$              ;BR IF NO
 3136  012576  000000                          HALT                    ;YES
 3137  012600                          6$:
 3138  012600  000002                          RTI                     ;RETURN
 3139
 3140
 3141                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
 3142
 3143                                  ;;*******************************************************************
 3144                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
 3145                                  ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
 3146                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
 3147
 3148  012602                          $ERRTYP:
 3149  012602  104401  001075                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 3150  012606  010046                          MOV     R0,-(SP)        ;;SAVE R0
 3151  012610  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
 3152  012612  153700  001014                  BISB    @#$ITEMB,R0
 3153  012616  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
 3154                                                                  ;;TYPE THE PC OF THE ERROR
 3155  012620  013746  001016                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
 3156                                                                  ;;ERROR ADDRESS
 3157  012624  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 3158  012626  000426                          BR      6$              ;;GET OUT
 3159  012630  005300                  1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
 3160  012632  006300                          ASL     R0              ;;      WORK FOR THE ERROR TABLE
 3161  012634  006300                          ASL     R0
 3162  012636  006300                          ASL     R0
 3163  012640  062700  001160                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
 3164  012644  012037  012654                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
 3165  012650  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
 3166  012652  104401                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
 3167  012654  000000                  2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
 3168  012656  104401  001075                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 3169  012662  012037  012672          3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
 3170  012666  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
 3171  012670  104401                          TYPE                    ;;TYPE THE "DATA HEADER"
 3172  012672  000000                  4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
 3173  012674  104401  001075                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 3174  012700  011000                  5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
 3175  012702  001004                          BNE     7$              ;;GO TYPE THE DATA
 3176  012704  012600                  6$:     MOV     (SP)+,R0        ;;RESTORE R0
 3177  012706  104401  001075                  TYPE    ,$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


                .SBTTL  POWER DOWN AND UP ROUTINES
                ;;**********************************************************************
                        ;*POWER DOWN ROUTINE
                ;;**********************************************************************
                $PWRDN: MOV     #$ILLUP,@#PWRVEC ;SET FOR FAST L
                        MOV     #340,@#PWRVEC+2 ;PRIO:7
                        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     R4,-(SP)        ;PUSH R4 ON STACK
                        MOV     R5,-(SP)        ;PUSH R5 ON STACK
                        MOV     @SWR,-(SP)      ;PUSH @SWR ON STACK
                        MOV     SP,$SAVR6       ;SAVE SP
                        MOV     #$PWRUP,@#PWRVEC ;SET UP VECTOR
                        HALT
                        BR      .-2             ;HANG UP


                ;;**********************************************************************
                        ;*POWER UP ROUTINE
                ;;**********************************************************************
                $PWRUP: MOV     #$ILLUP,@#PWRVEC ;SET FOR FAST DOWN
                        MOV     $SAVR6,SP       ;GET SP
                        MOV     (SP)+,@SWR      ;POP STACK INTO @SWR
                        MOV     (SP)+,R5
                        MOV     (SP)+,R4        ;POP STACK INTO R4
                        MOV     (SP)+,R3        ;POP STACK INTO R3
                        MOV     (SP)+,R2        ;POP STACK INTO R2
                        MOV     (SP)+,R1        ;POP STACK INTO R1
                        MOV     (SP)+,R0        ;POP STACK INTO R0
                        MOV     #$PWRDN,@#PWRVEC ;SET UP THE POWER DOWN VECTOR
                        MOV     #340,@#PWRVEC+2 ;PRIO:7
                        CLR     $SAVR6          ;WAIT LOOP FOR THE TTY
                1$:     INC     $SAVR6          ;WAIT FOR THE INC
                        BNE     1$              ;OF WORD
                        TYPE                    ;REPORT THE POWER FAILURE
                $PWRMG: .WORD   $POWER          ;POWER FAIL MESSAGE POINTER
                        RTI
                $ILLUP: HALT                    ;THE POWER UP SEQUENCE WAS STARTED
                        BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
                $SAVR6: 0                       ;PUT THE SP HERE
                $POWER: .ASCIZ  <15><12>"POWER"
```

```
3234
3235
3236
3237                                    .SBTTL  SCOPE HANDLER ROUTINE
3238
3239                                    ;;**********************************************************************
3240                                    ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
3241                                    ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
3242                                    ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
3243                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3244                                    ;*SW14=1        LOOP ON TEST
3245                                    ;*SW09=1        LOOP ON ERROR
3246                                    ;*CALL
3247                                    ;*      SCOPE           ;;SCOPE=IOT
3248
3249  013114                            $SCOPE:
3250  013114  104407                            CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
3251  013116  032777  040000  165714    1$:     BIT     @#BIT14,@SWR    ;;LOOP ON PRESENT TEST?
3252  013124  001052                            BNE     $OVER           ;;YES IF SW14=1
3253                                    ;@@@@@START OF CODE FOR THE XOR TESTER@@@@@
3254  013126  000416                    $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
3255                                                                    ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
3256  013130  013746  000004                   MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
3257  013134  012737  013154  000004           MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
3258  013142  005737  177060                   TST     @#177060        ;;TIME OUT ON XOR?
3259  013146  012637  000004                   MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
3260  013152  000421                            BR      $SVLAD          ;;GO TO THE NEXT TEST
3261  013154  022626                    5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
3262  013156  012637  000004                   MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
3263  013162  000407                            BR      7$              ;;LOOP ON THE PRESENT TEST
3264  013164                            6$:;@@@@@END OF CODE FOR THE XOR TESTER@@@@@
3265  013164  105737  001003            2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
3266  013170  001412                            BEQ     $SVLAD          ;;BR IF NO
3267  013172  032777  001000  165640           BIT     @#BIT09,@SWR    ;;LOOP ON ERROR?
3268  013200  001404                            BEQ     4$              ;;BR IF NO
3269  013202  013737  001010  001006    7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
3270  013210  000420                            BR      $OVER
3271  013212  105037  001003            4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
3272  013216  105237  001002            $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
3273  013222  113737  001002  001104           MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
3274  013230  011637  001006                   MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
3275  013234  011637  001010                   MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
3276  013240  005037  001072                   CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
3277  013244  112737  000001  001015           MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3278  013252  013777  001002  165562    $OVER:  MOV     $TSTNM,@#DISPLAY ;;DISPLAY TEST NUMBER
3279  013260  013716  001006                   MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
3280  013264  000002                            RTI                     ;;FIXES PS
3281
3282
3283                                    ;;**********************************************************************
3284                                    .SBTTL  APT COMMUNICATIONS ROUTINE
3285                                    ;;**********************************************************************
3286
3287  013266  112737  000001  013532    $ATY1:  MOVB    #1,$FFLG        ;TO REPORT FATAL ERROR
3288  013274  112737  000001  013530    $ATY3:  MOVB    #1,$MFLG        ;TO TYPE A MESSAGE
3289  013302  000403                            BR      $ATYC
```

```
3290  013304  112737  000001  013532  $ATY4:  MOVB    #1,$FFLG        ;TO ONLY REPORT FATAL ERROR
3291  013312                          $ATYC:
3292  013312  010046                          MOV     R0,-(SP)        ;PUSH R0 ON STACK
3293  013314  010146                          MOV     R1,-(SP)        ;PUSH R1 ON STACK
3294  013316  105737  013530                  TSTB    $MFLG           ;SHOULD TYPE A MESSAGE?
3295  013322  001450                          BEQ     5$              ;IF NOT: BR
3296  013324  122737  000001  001120          CMPB    #APTENV,$ENV    ;OPERATING UNDER APT?
3297  013332  001031                          BNE     3$              ;IF NOT: BR
3298  013334  132737  000100  001121          BITB    #APTSPOOL,$ENVM ;SHOULD SPOOL MESSAGE?
3299  013342  001425                          BEQ     3$              ;IF NOT: BR
3300  013344  017600  000004                  MOV     @4(SP),R0       ;GET MESSAGE ADDRESS
3301  013350  062766  000002  000004          ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
3302  013356  005737  001100          1$:     TST     $MSGTYPE                ;SEE IF DONE W/ LAST XMISSION?
3303  013362  001375                          BNE     1$              ;IF NOT: WAIT
3304  013364  010037  001114                  MOV     R0,$MSGAD       ;PUT ADDRESS IN MAILBOX
3305  013370  105720                  2$:     TSTB    (R0)+           ;FIND END OF MESSAGE
3306  013372  001376                          BNE     2$
3307  013374  163700  001114                  SUB     $MSGAD,R0       ;SUB START OF MESSAGE
3308  013400  006200                          ASR     R0              ;GET MESSAGE LENGTH IN WORDS
3309  013402  010037  001116                  MOV     R0,$MSGLGT      ;PUT LENGTH IN MAILBOX
3310  013406  012737  000004  001100          MOV     #4,$MSGTYPE     ;TELL APT TO TAKE MESSAGE
3311  013414  000413                          BR      5$
3312  013416  017637  000004  013442  3$:     MOV     @4(SP),4$       ;PUT MSG ADDR IN JSR LINKAGE
3313  013424  062766  000002  000004          ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
3314  013432  013746  177776                  MOV     177776,-(SP)    ;PUSH 177776 ON STACK
3315  013436  004737  013534                  JSR     PC,$TYPE        ;CALL TYPE MACRO
3316  013442  000000                  4$:     .WORD   0
3317  013444                          5$:
3318  013444  105737  013532          10$:    TSTB    $FFLG           ;SHOULD REPORT FATAL ERROR?
3319  013450  001413                          BEQ     12$             ;IF NOT: BR
3320  013452  005737  001120                  TST     $ENV            ;RUNNING UNDER APT?
3321  013456  001410                          BEQ     12$             ;IF NOT: BR
3322  013460  005737  001100          11$:    TST     $MSGTYPE        ;FINISHED LAST MESSAGE?
3323  013464  001375                          BNE     11$             ;IF NOT: WAIT
3324  013466  017637  000004  001102          MOV     @4(SP),$FATAL   ;GET ERROR #
3325  013474  005237  001100                  INC     $MSGTYPE        ;TELL APT TO TAKE ERROR
3326  013500  062766  000002  000004  12$:    ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
3327  013506  105037  013532                  CLRB    $FFLG           ;CLEAR FATAL FLAG
3328  013512  105037  013531                  CLRB    $LFLG           ;CLEAR LOG FLAG
3329  013516  105037  013530                  CLRB    $MFLG           ;CLEAR MESSAGE FLAG
3330  013522  012601                          MOV     (SP)+,R1        ;POP STACK INTO R1
3331  013524  012600                          MOV     (SP)+,R0        ;POP STACK INTO R1
3332  013526  000207                          RTS     PC              ;RETURN
3333  013530     000          $MFLG:  .BYTE   0
3334  013531     000          $LFLG:  .BYTE   0               ;LOG FLAG
3335  013532     000          $FFLG:  .BYTE   0               ;FATAL FLAG
3336
3337          013534                  .EVEN
3338          000200          APTSIZE=200
3339          000001          APTENV=001
3340          000100          APTSPOOL=100
3341          000040          APTCSUP=040
3342
3343
3344                          .SBTTL  TYPE ROUTINE
3345
```

```
3346                                    ;;*****************************************************************
3347                                    ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
3348                                    ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
3349                                    ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
3350                                    ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
3351                                    ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
3352                                    ;*
3353                                    ;*CALL:
3354                                    ;*1) USING A TRAP INSTRUCTION
3355                                    ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
3356                                    ;*OR
3357                                    ;*      TYPE
3358                                    ;*      MESADR
3359                                    ;*
3360
3361  013534  105737  001057           $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
3362  013540  100002                          BPL     1$              ;;BR IF YES
3363  013542  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
3364  013544  000430                          BR      3$              ;;LEAVE
3365  013546  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
3366  013550  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
3367  013554  122737  000001  001120          CMPB    @APTENV,$ENV    ;;RUNNING IN APT MODE
3368  013562  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
3369  013564  132737  000100  001121          BITB    @APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
3370  013572  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
3371  013574  010037  013604                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
3372  013600  004737  013274                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
3373  013604  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
3374  013606  132737  000040  001121  62$:    BITB    @APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
3375  013614  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
3376  013616  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
3377  013620  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
3378  013622  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
3379  013624  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
3380  013626  062716  000002          3$:     ADD     @2,(SP)         ;;ADJUST RETURN PC
3381  013632  000002                          RTI                     ;;RETURN
3382  013634  122716  000011          4$:     CMPB    @HT,(SP)        ;;BRANCH IF <HT>
3383  013640  001430                          BEQ     8$
3384  013642  122716  000200                  CMPB    @CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
3385  013646  001006                          BNE     5$
3386  013650  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
3387  013652  104401                          TYPE                    ;;TYPE A CR AND LF
3388  013654  001075                          $CRLF
3389  013656  105037  014064                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
3390  013662  000755                          BR      2$              ;;GET NEXT CHARACTER
3391  013664  004737  013746          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
3392  013670  123726  001056          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
3393  013674  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
3394  013676  013746  001054                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
3395                                          ;;AND THE NULL CHAR.
3396  013702  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
3397  013706  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
3398  013710  004737  013746                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
3399  013714  105337  014064                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
3400  013720  000770                          BR      7$              ;;LOOP
3401
```

L5


```
3402                                    ;HORIZONTAL TAB PROCESSOR
3403
3404  013722  112716  000040            8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
3405  013726  004737  013746            9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
3406  013732  132737  000007  014064            BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
3407  013740  001372                            BNE     9$              ;;TAB STOP
3408  013742  005726                            TST     (SP)+           ;;POP SPACE OFF STACK
3409  013744  000724                            BR      2$              ;;GET NEXT CHARACTER
3410  013746                          $TYPEC:
3411  013746  105777  165072                    TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?          ;MJD001
3412  013752  100022                            BPL     10$             ;;BR IF NOT                     ;MJD001
3413  013754  017746  165066                    MOV     @$TKB,-(SP)     ;;GET CHAR                      ;MJD001
3414  013760  042716  177600                    BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS         ;MJD001
3415  013764  122716  000023                    CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF                 ;MJD001
3416  013770  001012                            BNE     102$            ;;BR IF NOT                     ;MJD001
3417  013772                          101$:                                                           ;MJD001
3418  013772  105777  165046                    TSTB    @$TKS           ;;WAIT FOR CHAR                 ;MJD001
3419  013776  100375                            BPL     101$                                            ;MJD001
3420  014000  117716  165042                    MOVB    @$TKB,(SP)      ;;GET CHAR                      ;MJD001
3421  014004  042716  177600                    BIC     #177600,(SP)    ;;STRIP IT                      ;MJD001
3422  014010  122716  000021                    CMPB    #$XON,(SP)      ;;WAS IT XON?                   ;MJD001
3423  014014  001366                            BNE     101$            ;;BR IF NOT                     ;MJD001
3424  014016                          102$:                                                           ;MJD001
3425  014016  005726                            TST     (SP)+           ;;FIX STACK                     ;MJD001
3426  014020                          10$:                                                            ;MJD001
3427  014020  105777  165024                    TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
3428  014024  100375                            BPL     10$                                             ;MJD001
3429  014026  116677  000002  165016            MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
3430  014034  122766  000015  000002            CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
3431  014042  001003                            BNE     1$              ;;BRANCH IF NO
3432  014044  105037  014064                    CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
3433  014050  000406                            BR      $TYPEX          ;;EXIT
3434  014052  122766  000012  000002  1$:       CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
3435  014060  001402                            BEQ     $TYPEX          ;;BRANCH IF YES
3436  014062  105227                            INCB    (PC)+           ;;COUNT THE CHARACTER
3437  014064  000000                  $CHARCNT:.WORD  0                ;;CHARACTER COUNT STORAGE
3438  014066  000207                  $TYPEX: RTS     PC
3439
3440                                    .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3441
3442                                    ;;**********************************************************************
3443                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3444                                    ;*OCTAL (ASCII) NUMBER AND TYPE IT.
3445                                    ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3446                                    ;*CALL:
3447                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3448                                    ;*      TYPOS                   ;;CALL FOR TYPEOUT
3449                                    ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3450                                    ;*      .BYTE   M               ;;M=1 OR 0
3451                                    ;*                                      ;;1=TYPE LEADING ZEROS
3452                                    ;*                                      ;;0=SUPPRESS LEADING ZEROS
3453                                    ;*
3454                                    ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3455                                    ;*$TYPOS OR $TYPOC
3456                                    ;*CALL:
3457                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
```

```
3458                                    ;*      TYPON                   ;;CALL FOR TYPEOUT
3459                                    ;*
3460                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3461                                    ;*CALL:
3462                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3463                                    ;*      TYPOC                   ;;CALL FOR TYPEOUT
3464
3465  014070  017646  000000           $TYPOS: MOV     @(SP),-(SP)     ,;PICKUP THE MODE
3466  014074  116637  000001  014313           MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
3467  014102  112637  014315                   MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
3468  014106  062716  000002                   ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
3469  014112  000406                           BR      $TYPON
3470  014114  112737  000001  014313  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
3471  014122  112737  000006  014315           MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
3472  014130  112737  000005  014312  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
3473  014136  010346                           MOV     R3,-(SP)        ;;SAVE R3
3474  014140  010446                           MOV     R4,-(SP)        ;;SAVE R4
3475  014142  010546                           MOV     R5,-(SP)        ;;SAVE R5
3476  014144  113704  014315                   MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
3477  014150  005404                           NEG     R4
3478  014152  062704  000006                   ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
3479  014156  110437  014314                   MOVB    R4,$OMODE       ;;SAVE IT FOR USE
3480  014162  113704  014313                   MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
3481  014166  016605  000012                   MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
3482  014172  005003                           CLR     R3              ;;CLEAR THE OUTPUT WORD
3483  014174  006105                   1$:     ROL     R5              ;;ROTATE MSB INTO "C"
3484  014176  000404                           BR      3$              ;;GO DO MSB
3485  014200  006105                   2$:     ROL     R5              ;;FORM THIS DIGIT
3486  014202  006105                           ROL     R5
3487  014204  006105                           ROL     R5
3488  014206  010503                           MOV     R5,R3
3489  014210  006103                   3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
3490  014212  105337  014314                   DECB    $OMODE          ;;TYPE THIS DIGIT?
3491  014216  100016                           BPL     7$              ;;BR IF NO
3492  014220  042703  177770                   BIC     #177770,R3      ;;GET RID OF JUNK
3493  014224  001002                           BNE     4$              ;;TEST FOR 0
3494  014226  005704                           TST     R4              ;;SUPPRESS THIS 0?
3495  014230  001403                           BEQ     5$              ;;BR IF YES
3496  014232  005204                   4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
3497  014234  052703  000060                   BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
3498  014240  052703  000040           5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
3499  014244  110337  014310                   MOVB    R3,8$           ;;SAVE FOR TYPING
3500  014250  104401  014310                   TYPE    ,8$             ;;GO TYPE THIS DIGIT
3501  014254  105337  014312           7$:     DECB    $OCNT           ;;COUNT BY 1
3502  014260  003347                           BGT     2$              ;;BR IF MORE TO DO
3503  014262  002402                           BLT     6$              ;;BR IF DONE
3504  014264  005204                           INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
3505  014266  000744                           BR      2$              ;;GO DO THE LAST DIGIT
3506  014270  012605                   6$:     MOV     (SP)+,R5        ;;RESTORE R5
3507  014272  012604                           MOV     (SP)+,R4        ;;RESTORE R4
3508  014274  012603                           MOV     (SP)+,R3        ;;RESTORE R3
3509  014276  016666  000002  000004           MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
3510  014304  012616                           MOV     (SP)+,(SP)
3511  014306  000002                           RTI                     ;;RETURN
3512  014310     000                   8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
3513  014311     000                           .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
```

```
3514  014312  000                 $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
3515  014313  000                 $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
3516  014314  000000              $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
3517                              .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
3518
3519                              ;;**************************************************************
3520                              ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
3521                              ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
3522                              ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
3523                              ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
3524                              ;*REPLACED WITH SPACES.
3525                              ;*CALL:
3526                              ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
3527                              ;*      TYPDS                   ;;GO TO THE ROUTINE
3528
3529  014316                      $TYPDS:
3530  014316  010046                      MOV     R0,-(SP)        ;;PUSH R0 ON STACK
3531  014320  010146                      MOV     R1,-(SP)        ;;PUSH R1 ON STACK
3532  014322  010246                      MOV     R2,-(SP)        ;;PUSH R2 ON STACK
3533  014324  010346                      MOV     R3,-(SP)        ;;PUSH R3 ON STACK
3534  014326  010546                      MOV     R5,-(SP)        ;;PUSH R5 ON STACK
3535  014330  012746  020200              MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
3536  014334  016605  000020              MOV     20(SP),R5       ;;GET THE INPUT NUMBER
3537  014340  100004                      BPL     1$              ;;BR IF INPUT IS POS.
3538  014342  005405                      NEG     R5              ;;MAKE THE BINARY NUMBER POS.
3539  014344  112766  000055  000001      MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
3540  014352  005000              1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
3541  014354  012703  014532              MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
3542  014360  112723  000040              MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
3543  014364  005002              2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
3544  014366  016001  014522              MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
3545  014372  160105              3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
3546  014374  002402                      BLT     4$              ;;BR IF DONE
3547  014376  005202                      INC     R2              ;;INCREASE THE BCD DIGIT BY 1
3548  014400  000774                      BR      3$
3549  014402  060105              4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
3550  014404  005702                      TST     R2              ;;CHECK IF BCD DIGIT=0
3551  014406  001002                      BNE     5$              ;;FALL THROUGH IF 0
3552  014410  105716                      TSTB    (SP)            ;;STILL DOING LEADING 0'S?
3553  014412  100407                      BMI     7$              ;;BR IF YES
3554  014414  106316              5$:     ASLB    (SP)            ;;MSD?
3555  014416  103003                      BCC     6$              ;;BR IF NO
3556  014420  116663  000001  177777      MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
3557  014426  052702  000060      6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
3558  014432  052702  000040      7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
3559  014436  110223                      MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
3560  014440  005720                      TST     (R0)+           ;;JUST INCREMENTING
3561  014442  020027  000010              CMP     R0,#10          ;;CHECK THE TABLE INDEX
3562  014446  002746                      BLT     2$              ;;GO DO THE NEXT DIGIT
3563  014450  003002                      BGT     8$              ;;GO TO EXIT
3564  014452  010502                      MOV     R5,R2           ;;GET THE LSD
3565  014454  000764                      BR      6$              ;;GO CHANGE TO ASCII
3566  014456  105726              8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
3567  014460  100003                      BPL     9$              ;;BR IF NO
3568  014462  116663  177777  177776      MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
3569  014470  105013              9$:     CLRB    (R3)            ;;SET THE TERMINATOR
```

```
3570  014472  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
3571  014474  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
3572  014476  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
3573  014500  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
3574  014502  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
3575  014504  104401  014532                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
3576  014510  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
3577  014516  012616                          MOV     (SP)+,(SP)
3578  014520  000002                          RTI                     ;;RETURN TO USER
3579  014522  023420                  $DTBL:  10000.
3580  014524  001750                          1000.
3581  014526  000144                          100.
3582  014530  000012                          10.
3583  014532  000004                  $DBLK:  .BLKW   4
3584
3585
3586                                  .SBTTL  TTY INPUT ROUTINE
3587
3588                                  ;;**********************************************************************
3589                                  .ENABL  LSB
3590
3591                                  ;;**********************************************************************
3592                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
3593                                  ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
3594                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
3595                                  ;*WHEN OPERATING IN TTY FLAG MODE.
3596  014542  022737  000176  001040  $CKSWR: CMP     @#SWREG,SWR     ;;IS THE SOFT-SWR SELECTED?
3597  014550  001074                          BNE     15$             ;;BRANCH IF NO
3598  014552  105777  164266                  TSTB    @#TKS           ;;CHAR THERE?
3599  014556  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
3600  014560  117746  164262                  MOVB    @#TKB,-(SP)     ;;SAVE THE CHAR
3601  014564  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
3602  014570  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
3603  014574  001062                          BNE     15$             ;;NO, RETURN TO USER
3604  014576  123727  001034  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
3605  014604  001456                          BEQ     15$             ;;BRANCH IF YES
3606
3607  014606  104401  015277                  TYPE    .$CNTLG         ;;ECHO THE CONTROL-G (^G)
3608  014612  104401  015304          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
3609  014616  013746  000176                  MOV     @#SWREG,-(SP)   ;;SAVE SWREG FOR TYPEOUT
3610  014622  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3611  014624  104401  015315                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
3612  014630  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
3613  014632  005046                          CLR     -(SP)           ;;THE NEW SWR
3614  014634  105777  164204          7$:     TSTB    @#TKS           ;;CHAR THERE?
3615  014640  100375                          BPL     7$              ;;IF NOT TRY AGAIN
3616
3617  014642  117746  164200                  MOVB    @#TKB,-(SP)     ;;PICK UP CHAR
3618  014646  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
3619
3620
3621
3622  014652  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
3623  014656  001005                          BNE     10$             ;;BRANCH IF NOT
3624  014660  104401  015272                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
3625  014664  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
```

```
3626  014670  000757                          BR      19$             ;;LET'S TRY IT AGAIN
3627
3628
3629  014672  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
3630  014676  001022                          BNE     16$             ;;BRANCH IF NO
3631  014700  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
3632  014704  001403                          BEQ     11$             ;;BRANCH IF YES
3633  014706  016677  000002  164124          MOV     2(SP),@SWR      ;;SAVE NEW SWR
3634  014714  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
3635  014720  104401  001075          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
3636  014724  123727  001035  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
3637  014732  001003                          BNE     15$             ;;BRANCH IF NOT
3638  014734  012777  000100  164102          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
3639  014742  000002                  15$:    RTI                     ;;RETURN
3640  014744  004737  013746          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
3641  014750  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
3642  014754  002420                          BLT     18$             ;;BRANCH IF YES
3643  014756  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
3644  014762  003015                          BGT     18$             ;;BRANCH IF YES
3645  014764  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
3646  014770  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
3647  014774  001403                          BEQ     17$             ;;BRANCH IF YES
3648  014776  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
3649  015000  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
3650  015002  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
3651  015004  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
3652  015010  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
3653  015014  000707                          BR      7$              ;;GET THE NEXT ONE
3654  015016  104401  001074          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
3655  015022  000720                          BR      20$             ;;SIMULATE CONTROL-U
3656                                  .DSABL  LSB
3657
3658
3659                                  ;;**************************************************************************
3660                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
3661                                  ;*CALL:
3662                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
3663                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
3664                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
3665                                  ;
3666
3667  015024  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
3668  015026  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
3669  015034  105777  164004          1$:     TSTB    @$TKS           ;;WAIT FOR
3670  015040  100375                          BPL     1$              ;;A CHARACTER
3671  015042  117766  164000  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
3672  015050  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
3673  015056  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
3674  015064  001013                          BNE     3$              ;;BRANCH IF NO
3675  015066  105777  163752          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
3676  015072  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
3677  015074  117746  163746                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
3678  015100  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
3679  015104  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
3680  015110  001366                          BNE     2$              ;;IF NOT DISCARD IT
3681  015112  000750                          BR      1$              ;;YES, RESUME
```

```
3682  015114  026627  000004  000021  3$:     CMP     4(SP),#$XON     ;;IS IT A RANDOM XON?                    ;RAN001
3683  015122  001744                          BEQ     1$              ;;BRANCH IF YES                          ;RAN001
3684  015124  026627  000004  000140          CMP     4(SP),#140      ;;IS IT UPPER CASE?
3685  015132  002407                          BLT     4$              ;;BRANCH IF YES
3686  015134  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
3687  015142  003003                          BGT     4$              ;;BRANCH IF YES
3688  015144  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
3689  015152  000002                  4$:     RTI                     ;;GO BACK TO USER
3690                                          ;;**************************************************************
3691                                          ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
3692                                          ;*CALL:
3693                                          ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
3694                                          ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
3695                                          ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
3696
3697  015154  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
3698  015156  012703  015262          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
3699  015162  022703  015272          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
3700  015166  101405                          BLOS    4$              ;;BR IF YES
3701  015170  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
3702  015172  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
3703  015174  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
3704  015200  001003                          BNE     3$              ;;SKIP IF NOT
3705  015202  104401  001074          4$:     TYPE    ,$QUES          ;;TYPE A '?'
3706  015206  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
3707  015210  111337  015260          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
3708  015214  104401  015260                  TYPE    ,9$
3709  015220  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
3710  015224  001356                          BNE     2$              ;;LOOP IF NOT RETURN
3711  015226  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
3712  015232  104401  001076                  TYPE    ,$LF            ;;TYPE A LINE FEED
3713  015236  012603                          MOV     (SP)+,R3        ;;RESTORE R3
3714  015240  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
3715  015242  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
3716  015250  012766  015262  000004          MOV     #$TTYIN,4(SP)
3717  015256  000002                          RTI                     ;;RETURN
3718  015260     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
3719  015261     000                          .BYTE   0               ;;TERMINATOR
3720  015262  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
3721  015272  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
3722  015277     136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
3723  015304  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
3724  015312  020075     000
3725
3726  015315     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
3727  015322  036440  000040
3728
3729
3730
3731
3732                                          .SBTTL  TRAP DECODER
3733
3734                                          ;;**************************************************************
3735                                          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3736                                          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3737                                          ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
```

```
                                        ;*GO TO THAT ROUTINE.
                                        
015326  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
015330  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
015334  005740                          TST     -(R0)           ;;BACKUP BY 2
015336  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
015340  006300                          ASL     R0              ;;POSITION FOR INDEXING
015342  016000  015362                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
015346  000200                          RTS     R0              ;;GO TO ROUTINE
                                        
                                        ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
                                        
015350  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
015352  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
015360  000002                          RTI                     ;;RESTORE THE PSW
                                        
                                        .SBTTL  TRAP TABLE
                                        
                                        ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                        ;*BY THE "TRAP" INSTRUCTION.
                                        
                                        ;       ROUTINE
                                        ;       -------
015362  015350                  $TRPAD: .WORD   $TRAP2
015364  013534                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
015366  014114                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
015370  014070                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
015372  014130                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
015374  014316                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
                                        
015376  014612                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
                                        
015400  014542                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
015402  015024                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
015404  015154                          $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 RESTARTED AFTER HALTING BY JUST CONTINUING
                                    ;;**********************************************************************
015406  012737  000176  001040  ECHO:   MOV     #SWREG,SWR              ;SET UP FOR SOFTWARE SWITCH REGISTER
015414  012737  000174  001042          MOV     #DISPREG,DISPLAY        ;AND DISPLAY REGISTER
015422  032777  000020  163410          BIT     #BIT4, @SWR             ;CHECK IF BIT4 SET IN SWITCH REG
015430  001403                          BEQ     1$                      ;IF NOT THEN SELECT SLU1
015432  012703  002750                  MOV     #RCSR, R3               ;IF BIT4 SET THEN SELECT SLU2
015436  000402                          BR      2$
015440  012703  002760          1$:     MOV     #CRCSR, R3              ;SELECT SLU1 (THIS IS DEFAULT DEVICE)
015444  000005                  2$:     RESET                   ;CLEAR EVERYTHING
015446  112773  000052  000006          MOVB    #'*,@6(R3)      ;TRANSMIT PROMPT "*"
015454  105773  000000          3$:     TSTB    @(R3)           ;WAIT FOR INPUT
015460  100375                          BPL     3$
015462  117373  000002  000006          MOVB    @2(R3),@6(R3)   ;ECHO INPUT
```

F6


```
3794  015470  017300  000002          MOV     @2(R3),R0       ;STORE INPUT
3795  015474  100023                  BPL     6$              ;BR IF "ERROR"NOT SET
3796  015476  052701  010000          BIS     #BIT12,R1       ;SET PARITY ERROR TEST MASK
3797  015502  030100                  BIT     R1,R0           ;CHECK  FOR PARITY ERROR FLAG
3798  015504  001403                  BEQ     4$              ;BR IF NOT SET
3799  015506  004737  015570          JSR     PC,MSG          ;REPORT PARITY ERROR
3800  015512  015616                  MPAR
3801  015514  006301          4$:     ASL     R1              ;SHIFT MASK TO TEST "FR" FLAG
3802  015516  030100                  BIT     R1,R0           ;TEST FOR FRAMING ERROR FLAG
3803  015520  001403                  BEQ     5$              ;BR IF NOT SET
3804  015522  004737  015570          JSR     PC,MSG          ;REPSORT FRAMING ERROR
3805  015526  015627                  MFR
3806  015530  006301          5$:     ASL     R1              ;SHIFT MASK TO TEST "OR" FLAG
3807  015532  030100                  BIT     R1,R0           ;TEST FOR OVERFLOW ERROR
3808  015534  001403                  BEQ     6$              ;BR IF NOT SET
3809  015536  004737  015570          JSR     PC,MSG          ;REPORT OVERFLOW ERROR
3810  015542  015641                  MOR
3811  015544  042700  000200  6$:     BIC     #BIT7,R0        ;CLEAR ANY PARITY BIT
3812  015550  022700  000003          CMP     #3,R0           ;WAS INPUT CONTROL-C
3813  015554  001337                  BNE     3$              ;BR IF IS NOT
3814  015556  004737  015570          JSR     PC,MSG          ;REPORT PROGRAM STOP
3815  015562  015654                  MSTOP
3816  015564  000000                  HALT                    ;END OF TEST HALT
3817  015566  000707                  BR      ECHO            ;AFTER END OF TEST HALT
3818                                                          ;   PRESS CONTINUE TO RESTART ECHO TEST
3819
3820  015570  013600          MSG:    MOV     @(SP)+,R0       ;PICK UP MESSAGE POINTER
3821  015572  062746  000002          ADD     #2,-(SP)        ;ADJUST RETURN PC
3822  015576  105773  000004  WAIT:   TSTB    @4(R3)          ;WAIT FOR XMIT DONE
3823  015602  100375                  BPL     WAIT
3824  015604  112073  000006          MOVB    (R0)+,@6(R3)    ;SEND CHARACTER
3825  015610  105710                  TSTB    (R0)            ;IS THIS END OF MESSAGE?
3826  015612  001371                  BNE     WAIT            ;BR IF NOT
3827  015614  000207                  RTS     PC              ;RETURN
3828
3829  015616  005015  040520  044522  MPAR:   .ASCIZ  <CR><LF>/PARITY/
3830  015624  054524     000
3831
3832  015627     015  043012  040522  MFR:    .ASCIZ  <CR><LF>/FRAMING/
3833  015634  044515  043516     000
3834
3835  015641     015  047412  042520  MOR:    .ASCIZ  <CR><LF>/OVERFLOW/
3836  015646  043122  047514  000127
3837
3838  015654  005015  052123  050117  MSTOP:  .ASCIZ  <CR><LF>/STOP/
3839  015662     000
3840
3841
3842
3843          015664                  .EVEN
3844
3845                          .SBTTL  TERMINAL OUTPUT TEST
3846
3847                          ;;*****************************************************************
3848                                  ;*THIS ROUTINE WILL OUTPUT ALL WRITABLE CHARACTERS FOR THE
3849                                  ;*THE OCTAL CODE 040 --> 377
```

```
                                        ;*32 CHARACTERS ARE PRINTED ON EACH LINE
                                        ;*THE PATTERN IS REPEATED EVERY THREE LINES
                                        ;*
                                    ;;*****************************************************************
015664  012737  000176  001040  OUTTST: MOV     #SWREG,SWR              ;SET UP FOR SOFTWARE SWITCH REGISTER
015672  012737  000174  001042          MOV     #DISPREG,DISPLAY        ;AND DISPLAY REGISTER
015700  032777  000020  163132          BIT     #BIT4, @SWR             ;CHECK IF BIT4 SET IN SWITCH REG
015706  001403                          BEQ     1$                      ;IF NOT THEN SELECT SLU1
015710  012703  002750                  MOV     #RCSR, R3               ;IF BIT4 SET THEN SELECT SLU2
015714  000402                          BR      2$
015716  012703  002760          1$:     MOV     #CRCSR, R3              ;SELECT SLU1 (THIS IS DEFAULT DEVICE)
015722  000005                  2$:     RESET                   ;CLEAR THE WORLD
015724  012701  000040          3$:     MOV     #40,R1          ;LOAD FIRST WRITABLE CHARACTER
015730  012700  000040          4$:     MOV     #40,R0          ;LOAD CHAR COUNT PER LINE
015734  105773  000004          5$:     TSTB    @4(R3)          ;WAIT FOR DONE
015740  100375                          BPL     5$
015742  010173  000006                  MOV     R1,@6(R3)       ;TRANSMIT A CHARACTER
015746  105201                          INCB    R1              ;INCREMENT CHARACTER CODE
015750  005300                          DEC     R0              ;DECREMENT CHAR COUNT
015752  001370                          BNE     5$              ;BR IF LINE NOT COMPLETE
015754  004737  015570                  JSR     PC,MSG          ;SSUE CR,LINE FEED

015760  001075                          $CRLF
015762  105773  000000                  TSTB    @(R3)           ;ANY CHARACTER RECEIVED?
015766  100404                          BMI     6$              ;BR IF YES
015770  032701  000200                  BIT     #BIT7,R1        ;FINISHED ONE PASS OF WRITABLE CHARACTERS?
015774  001353                          BNE     3$              ;BR IF YES
015776  000754                          BR      4$              ;IF NOT WRITE NEXT LINE

016000  005073  000002          6$:     CLR     @2(R3)          ;CLEAR RECEIVER
016004  000000                          HALT                    ;STOP TEST
016006  000726                          BR      OUTTST          ;RESTART TEST IF CONTINUED


016010  046123  030525  052040  EM5:    .ASCIZ  /SLU1 TCSR DONE NOT SET WITH RESET/
016016  051503  020122  047504
016024  042516  047040  052117
016032  051440  052105  053440
016040  052111  020110  042522
016046  042523  000124


016052  046123  030525  051040  EM6:    .ASCIZ  /SLU1 RCSR DID NOT RETURN SSYNC/
016060  051503  020122  044504
016066  020104  047516  020124
016074  042522  052524  047122
016102  051440  054523  041516
016110     000
```

H6


```
3906
3907  016111     123  052514  020061  EM7:    .ASCIZ  /SLU1 RBUF DID NOT RETURN SSYNC/
3908  016116  041122  043125  042040
3909  016124  042111  047040  052117
3910  016132  051040  052105  052125
3911  016140  020116  051523  047131
3912  016146  000103
3913
3914
3915
3916
3917
3918  016150  040503  020116  047516  EM11:   .ASCIZ  /CAN NOT SET BIT2 OF SLU1 TCSR/
3919  016156  020124  042523  020124
3920  016164  044502  031124  047440
3921  016172  020106  046123  030525
3922  016200  052040  051503  000122
3923
3924
3925
3926
3927  016206  042522  042523  020124  EM13:   .ASCIZ  /RESET DID NOT CLEAR BIT2 OF SLU1 TCSR/
3928  016214  044504  020104  047516
3929  016222  020124  046103  040505
3930  016230  020122  044502  031124
3931  016236  047440  020106  046123
3932  016244  030525  052040  051503
3933  016252  000122
3934
3935
3936
3937
3938
3939
3940  016254  044502  033124  047440  EM14:   .ASCIZ  /BIT6 OF SLU1 TCSR NOT CLEAR AFTER RESET/
3941  016262  020106  046123  030525
3942  016270  052040  051503  020122
3943  016276  047516  020124  046103
3944  016304  040505  020122  043101
3945  016312  042524  020122  042522
3946  016320  042523  000124
3947
3948
3949
3950
3951
3952
3953  016324  046123  030525  054040  EM15:   .ASCIZ  /SLU1 XMIT INTERRUPTED WITH PRIORITY 7/
3954  016332  044515  020124  047111
3955  016340  042524  051122  050125
3956  016346  042524  020104  044527
3957  016354  044124  050040  044522
3958  016362  051117  052111  020131
3959  016370  000067
3960
3961
```

```
3962
3963
3964
3965
3966  016372  040503  020116  047516  EM16:   .ASCIZ  /CAN NOT SET BIT6 OF SLU1 TCSR/
3967  016400  020124  042523  020124
3968  016406  044502  033124  047440
3969  016414  020106  046123  030525
3970  016422  052040  051503  000122
3971
3972
3973
3974
3975  016430  040503  020116  047516  EM17:   .ASCIZ  /CAN NOT CLEAR BIT6 OF SLU1 TCSR/
3976  016436  020124  046103  040505
3977  016444  020122  044502  033124
3978  016452  047440  020106  046123
3979  016460  030525  052040  051503
3980  016466  000122
3981
3982
3983
3984
3985
3986  016470  042522  042523  020124  EM20:   .ASCIZ  /RESET DID NOT CLEAR BIT6 OF SLU1 TCSR/
3987  016476  044504  020104  047516
3988  016504  020124  046103  040505
3989  016512  020122  044502  033124
3990  016520  047440  020106  046123
3991  016526  030525  052040  051503
3992  016534  000122
3993
3994
3995
3996
3997
3998
3999  016536  044502  033124  047440  EM21:   .ASCIZ  /BIT6 OF SLU1 RCSR NOT CLEAR AFTER RESET/
4000  016544  020106  046123  030525
4001  016552  051040  051503  020122
4002  016560  047516  020124  046103
4003  016566  040505  020122  043101
4004  016574  042524  020122  042522
4005  016602  042523  000124
4006
4007
4008
4009
4010
4011
4012  016606  046123  030525  051040  EM22:   .ASCIZ  /SLU1 RCVR INTERRUPT WITH PRIORITY 7/
4013  016614  053103  020122  047111
4014  016622  042524  051122  050125
4015  016630  020124  044527  044124
4016  016636  050040  044522  051117
4017  016644  052111  020131  000067
```

J6


```
4018
4019
4020
4021
4022
4023  016652  040503  020116  047516  EM23:   .ASCIZ  /CAN NOT SET BIT6 OF SLU1 RCSR/
4024  016660  020124  042523  020124
4025  016666  044502  033124  047440
4026  016674  020106  046123  030525
4027  016702  051040  051503  000122
4028
4029
4030
4031
4032  016710  040503  020116  047516  EM24:   .ASCIZ  /CAN NOT CLEAR BIT6 OF SLU1 RCSR/
4033  016716  020124  046103  040505
4034  016724  020122  044502  033124
4035  016732  047440  020106  046123
4036  016740  030525  051040  051503
4037  016746  000122
4038
4039
4040
4041
4042
4043  016750  040503  020116  047516  EM25:   .ASCIZ  /CAN NOT CLEAR BIT6 OF SLU1 RCSR WITH RESET/
4044  016756  020124  046103  040505
4045  016764  020122  044502  033124
4046  016772  047440  020106  046123
4047  017000  030525  051040  051503
4048  017006  020122  044527  044124
4049  017014  051040  051505  052105
4050  017022     000
4051
4052
4053
4054
4055
4056
4057
4058  017023     123  052514  020061  EM26:   .ASCIZ  /SLU1 RECEIVER DONE NEVER SET/
4059  017030  042522  042503  053111
4060  017036  051105  042040  047117
4061  017044  020105  042516  042526
4062  017052  020122  042523  000124
4063
4064
4065
4066
4067  017060  042522  042523  020124  EM27:   .ASCIZ  /RESET DID NOT CLEAR SLU1 RCVR DONE/
4068  017066  044504  020104  047516
4069  017074  020124  046103  040505
4070  017102  020122  046123  030525
4071  017110  051040  053103  020122
4072  017116  047504  042516     000
4073
```

K6


```
4074
4075
4076
4077
4078  017123     122  040505  044504  EM30:   .ASCIZ  /READING SLU1 RBUF DID NOT CLEAR RCVR DONE/
4079  017130  043516  051440  052514
4080  017136  020061  041122  043125
4081  017144  042040  042111  047040
4082  017152  052117  041440  042514
4083  017160  051101  051040  053103
4084  017166  020122  047504  042516
4085  017174     000
4086
4087
4088
4089
4090
4091
4092
4093  017175     123  052514  020062  EM31:   .ASCIZ  /SLU2 TCSR DID NOT RETURN SSYNC/
4094  017202  041524  051123  042040
4095  017210  042111  047040  052117
4096  017216  051040  052105  051125
4097  017224  020116  051523  047131
4098  017232  000103
4099
4100
4101
4102
4103
4104  017234  046123  031125  052040  EM32:   .ASCIZ  /SLU2 TBUF DID NOT RETURN SSYNC/
4105  017242  052502  020106  044504
4106  017250  020104  047516  020124
4107  017256  042522  052524  047122
4108  017264  051440  054523  041516
4109  017272     000
4110
4111
4112
4113
4114
4115  017273     123  052514  020062  EM33:   .ASCIZ  /SLU2 TCSR DONE NOT CLEARED WITH TBUF FULL/
4116  017300  041524  051123  042040
4117  017306  047117  020105  047516
4118  017314  020124  046103  040505
4119  017322  042522  020104  044527
4120  017330  044124  052040  052502
4121  017336  020106  052506  046114
4122  017344     000
4123
4124
4125
4126
4127
4128
4129
```

L6


```
 4130  017345     123  052514  020062  EM34:   .ASCI7  /SLU2 TCSR DONE NOT SET AFTER TRANSMIT/
 4131  017352  041524  051123  042040
 4132  017360  047117  020105  047516
 4133  017366  020124  042523  020124
 4134  017374  043101  042524  020122
 4135  017402  051124  047101  046523
 4136  017410  052111     000
 4137
 4138
 4139
 4140
 4141
 4142
 4143  017413     123  052514  020062  EM35:   .ASCIZ  /SLU2 TCSR DONE NOT SET WITH RESET/
 4144  017420  041524  051123  042040
 4145  017426  047117  020105  047516
 4146  017434  020124  042523  020124
 4147  017442  044527  044124  051040
 4148  017450  051505  052105     000
 4149
 4150
 4151
 4152
 4153
 4154  017455     123  052514  020062  EM36:   .ASCIZ  /SLU2 RCSR DID NOT RETURN SSYNC/
 4155  017462  041522  051123  042040
 4156  017470  042111  047040  052117
 4157  017476  051040  052105  051125
 4158  017504  020116  051523  047131
 4159  017512  000103
 4160
 4161
 4162
 4163
 4164
 4165  017514  046123  031125  051040  EM37:   .ASCIZ  /SLU2 RBUF DID NOT RETURN SSYNC/
 4166  017522  052502  020106  044504
 4167  017530  020104  047516  020124
 4168  017536  042522  052524  047122
 4169  017544  051440  054523  041516
 4170  017552     000
 4171
 4172
 4173
 4174
 4175
 4176  017553     102  052111  020060  EM40:   .ASCIZ  /BIT0 OF SLU2 TCSR NOT CLEAR AFTER RESET/
 4177  017560  043117  051440  052514
 4178  017566  020062  041524  051123
 4179  017574  047040  052117  041440
 4180  017602  042514  051101  040440
 4181  017610  052106  051105  051040
 4182  017616  051505  052105     000
 4183
 4184
 4185
```

M6


```
017623      103  047101  047040  EM41:   .ASCIZ  /CAN NOT SET BIT0 OF SLU2 TCSR/
017630   052117  051440  052105
017636   041040  052111  020060
017644   043117  051440  052514
017652   020062  041524  051123
017660      000

017661      103  047101  047040  EM42:   .ASCIZ  /CAN NOT CLEAR BIT0 OF SLU2 TCSR/
017666   052117  041440  042514
017674   051101  041040  052111
017702   020060  043117  051440
017710   052514  020062  041524
017716   051123     000

017721      122  051505  052105  EM43:   .ASCIZ  /RESET DID NOT CLEAR BIT0 OF SLU2 TCSR/
017726   042040  042111  047040
017734   052117  041440  042514
017742   051101  041040  052111
017750   020060  043117  051440
017756   052514  020062  041524
017764   051123     000

017767      102  052111  020066  EM44:   .ASCIZ  /BIT6 OF SLU2 TCSR NOT CLEAR AFTER RESET/
017774   043117  051440  052514
020002   020062  041524  051123
020010   047040  052117  041440
020016   042514  051101  040440
020024   052106  051105  051040
020032   051505  052105     000

020037      123  052514  020062  EM45:   .ASCIZ  /SLU2 XMIT INTERRUPTED WITH PRIORITY 7/
020044   046530  052111  044440
020052   052116  051105  052522
020060   052120  042105  053440
020066   052111  020110  051120
```

N6


```
4242  020074  047511  044522  054524
4243  020102  033440     000
4244
4245
4246
4247
4248
4249
4250  020105     103  047101  047040  EM46:   .ASCIZ  /CAN NOT SET BIT6 OF SLU2 TCSR/
4251  020112  052117  051440  052105
4252  020120  041040  052111  020066
4253  020126  043117  051440  052514
4254  020134  020062  041524  051123
4255  020142     000
4256
4257
4258
4259
4260
4261  020143     103  047101  047040  EM47:   .ASCIZ  /CAN NOT CLEAR BIT6 OF SLU2 TCSR/
4262  020150  052117  041440  042514
4263  020156  051101  041040  052111
4264  020164  020066  043117  051440
4265  020172  052514  020062  041524
4266  020200  051123     000
4267
4268
4269
4270
4271
4272  020203     122  051505  052105  EM50:   .ASCIZ  /RESET DID NOT CLEAR BIT6 OF SLU2 TCSR/
4273  020210  042040  042111  047040
4274  020216  052117  041440  042514
4275  020224  051101  041040  052111
4276  020232  020066  043117  051440
4277  020240  052514  020062  041524
4278  020246  051123     000
4279
4280
4281
4282
4283
4284
4285  020251     102  052111  020066  EM51:   .ASCIZ  /BIT6 OF SLU2 RCSR NOT CLEAR AFTER RESET/
4286  020256  043117  051440  052514
4287  020264  020062  041522  051123
4288  020272  047040  052117  041440
4289  020300  042514  051101  040440
4290  020306  052106  051105  051040
4291  020314  051505  052105     000
4292
4293
4294
4295
4296
4297
```

B7


```
4298  020321     123  052514  020062  EM52:   .ASCIZ  /SLU2 RCVR INTERRUPT WITH PRIORITY 7/
4299  020326  041522  051126  044440
4300  020334  052116  051105  052522
4301  020342  052120  053440  052111
4302  020350  020110  051120  047511
4303  020356  044522  054524  033440
4304  020364     000
4305
4306
4307
4308
4309
4310
4311  020365     103  047101  047040  EM53:   .ASCIZ  /CAN NOT SET BIT6 OF SLU2 RCSR/
4312  020372  052117  051440  052105
4313  020400  041440  052111  020066
4314  020406  043117  051440  052514
4315  020414  020062  041522  051123
4316  020422     000
4317
4318
4319
4320
4321
4322  020423     103  047101  047040  EM54:   .ASCIZ  /CAN NOT CLEAR BIT6 OF SLU2 RCSR/
4323  020430  052117  041440  042514
4324  020436  051101  041040  052111
4325  020444  020066  043117  051440
4326  020452  052514  020062  041522
4327  020460  051123     000
4328
4329
4330
4331
4332
4333  020463     103  047101  047040  EM55:   .ASCIZ  /CAN NOT CLEAR BIT6 OF SLU2 RCSR WITH RESET/
4334  020470  052117  041440  042514
4335  020476  051101  041040  052111
4336  020504  020066  043117  051440
4337  020512  052514  020062  041522
4338  020520  051123  053440  052111
4339  020526  020110  042522  042523
4340  020534  000124
4341
4342
4343
4344
4345
4346
4347
4348  020536  046123  031125  051040  EM56:   .ASCIZ  /SLU2 RECEIVER DONE NEVER SET/
4349  020544  041505  044505  042526
4350  020552  020122  047504  042516
4351  020560  047040  053105  051105
4352  020566  051440  052105     000
4353
```

C7


```
4354
4355
4356
4357  020573     122  051505  052105  EM57:   .ASCIZ  /RESET DID NOT CLEAR SLU2 RCVR DONE/
4358  020600  042040  042111  047040
4359  020606  052117  041440  042514
4360  020614  051101  051440  052514
4361  020622  020062  041522  051126
4362  020630  042040  047117  000105
4363
4364
4365
4366
4367
4368  020636  042522  042101  047111  EM60:   .ASCIZ  /READING SLU2 RBUF DID NOT CLEAR RCVR DONE/
4369  020644  020107  046123  031125
4370  020652  051040  052502  020106
4371  020660  044504  020104  047516
4372  020666  020124  046103  040505
4373  020674  020122  041522  051126
4374  020702  042040  047117  000105
4375
4376
4377
4378
4379
4380
4381  020710  045514  020123  044504  EM61:   .ASCIZ  /LKS DID NOT RETURN SSYNC/
4382  020716  020104  047516  020124
4383  020724  042522  052524  047122
4384  020732  051440  054523  041516
4385  020740     000
4386
4387
4388
4389
4390  020741     102  052111  020066  EM62:   .ASCIZ  /BIT6 OF LKS NOT CLEAR AFTER RESET/
4391  020746  043117  046040  051513
4392  020754  047040  052117  041440
4393  020762  042514  051101  040440
4394  020770  052106  051105  051040
4395  020776  051505  052105     000
4396
4397
4398
4399
4400
4401  021003     114  051513  044440  EM63:   .ASCIZ  /LKS INTERRUPT WITH PRIORITY 7/
4402  021010  052116  051105  052522
4403  021016  052120  053440  052111
4404  021024  020110  051120  047511
4405  021032  044522  054524  033440
4406  021040     000
4407
4408
4409
```

```
4410
4411
4412  021041     103  047101  047040  EM64:   .ASCIZ  /CAN NOT SET BIT6 OF LKS/
4413  021046  052117  051440  052105
4414  021054  041040  052111  020066
4415  021062  043117  046040  051513
4416  021070     000
4417
4418
4419
4420
4421  021071     103  047101  047040  EM65:   .ASCIZ  /CAN NOT CLEAR BIT6 OF LKS/
4422  021076  052117  041440  042514
4423  021104  051101  041040  052111
4424  021112  020066  043117  046040
4425  021120  051513     000
4426
4427
4428
4429
4430  021123     122  051505  052105  EM66:   .ASCIZ  /RESET DID NOT CLEAR BIT6 OF LKS/
4431  021130  042040  042111  047040
4432  021136  052117  041440  042514
4433  021144  051101  041040  052111
4434  021152  020066  043117  046040
4435  021160  051513     000
4436
4437
4438
4439
4440
4441  021163     102  052111  020067  EM67:   .ASCIZ  /BIT7 OF LKS NOT SET AFTER RESET/
4442  021170  043117  046040  051513
4443  021176  047040  052117  051440
4444  021204  052105  040440  052106
4445  021212  051105  051040  051505
4446  021220  052105     000
4447
4448
4449
4450
4451
4452  021223     103  047101  047040  EM70:   .ASCIZ  /CAN NOT CLEAR BIT7 OF LKS/
4453  021230  052117  041440  042514
4454  021236  051101  041040  052111
4455  021244  020067  043117  046040
4456  021252  051513     000
4457
4458
4459
4460
4461  021255     102  052111  020067  EM71:   .ASCIZ  /BIT7 OF LKS DOES NOT SET/
4462  021262  043117  046040  051513
4463  021270  042040  042517  020123
4464  021276  047516  020124  042523
4465  021304  000124
```

```
4466
4467
4468
4469
4470  021306  051127  052111  047111  EM72:   .ASCIZ  /WRITING TO ONE INTERNAL ADDRESS MODIFIED ANOTHER/
4471  021314  020107  047524  047440
4472  021322  042516  044440  052116
4473  021330  051105  040516  020114
4474  021336  042101  051104  051505
4475  021344  020123  047515  044504
4476  021352  044506  042105  040440
4477  021360  047516  044124  051105
4478  021366     000
4479
4480
4481
4482
4483
4484
4485
4486
4487  021367     123  052514  020061  EM74:   .ASCIZ  /SLU1 XMIT INTERRUPTS WHEN DISABLED/
4488  021374  046530  052111  044440
4489  021402  052116  051105  052522
4490  021410  052120  020123  044127
4491  021416  047105  042040  051511
4492  021424  041101  042514  000104
4493
4494
4495
4496
4497
4498  021432  046123  030525  054040  EM75:   .ASCIZ  /SLU1 XMIT DID NOT INTERRUPT/
4499  021440  044515  020124  044504
4500  021446  020104  047516  020124
4501  021454  047111  042524  051122
4502  021462  050125  000124
4503
4504
4505
4506
4507  021466  046123  030525  054040  EM76:   .ASCIZ  /SLU1 XMIT INTERRUPT AT PRIORITY 7/
4508  021474  044515  020124  047111
4509  021502  042524  051122  050125
4510  021510  020124  052101  050040
4511  021516  044522  051117  052111
4512  021524  020131  000067
4513
4514
4515
4516
4517
4518  021530  046123  030525  054040  EM77:   .ASCIZ  /SLU1 XMIT INTERRUPTS WITH ENABLE CLEAR/
4519  021536  044515  020124  047111
4520  021544  042524  051122  050125
4521  021552  051524  053440  052111
```

F7


```
4522  021560  020110  047105  041101
4523  021566  042514  041440  042514
4524  021574  051101     000
4525
4526
4527
4528
4529
4530
4531  021577     123  052514  020061  EM100:  .ASCIZ  /SLU1 XMIT DID NOT INTERRUPT/
4532  021604  046530  052111  042040
4533  021612  042111  047040  052117
4534  021620  044440  052116  051105
4535  021626  052522  052120     000
4536
4537
4538
4539
4540  021633     123  052514  020061  EM101:  .ASCIZ  /SLU1 XMIT RE-INTERRUPTED/
4541  021640  046530  052111  051040
4542  021646  026505  047111  042524
4543  021654  051122  050125  042524
4544  021662  000104
4545
4546
4547
4548
4549  021664  047514  042101  047111  EM102:  .ASCIZ  /LOADING SLU1 TBUF DID NOT CLEAR INTERRUPT/
4550  021672  020107  046123  030525
4551  021700  052040  052502  020106
4552  021706  044504  020104  047516
4553  021714  020124  046103  040505
4554  021722  020122  047111  042524
4555  021730  051122  050125  000124
4556
4557
4558
4559
4560
4561
4562  021736  046123  030525  051040  EM103:  .ASCIZ  /SLU1 RCVR INTERRUPTS WITH ENABLE CLEAR/
4563  021744  053103  020122  047111
4564  021752  042524  051122  050125
4565  021760  051524  053440  052111
4566  021766  020110  047105  041101
4567  021774  042514  041440  042514
4568  022002  051101     000
4569
4570
4571
4572
4573
4574
4575  022005     123  052514  020061  EM104:  .ASCIZ  /SLU1 RCVR DID NOT INTERRUPT/
4576  022012  041522  051126  042040
4577  022020  042111  047040  052117
```

```
4578  022026  044440  052116  051105
4579  022034  052522  052120     000
4580
4581
4582
4583
4584  022041     123  052514  020061  EM105:  .ASCIZ  /SLU1 RCVR INTERRUPTS AT PRIORITY 7/
4585  022046  041522  051126  044440
4586  022054  052116  051105  052522
4587  022062  052120  020123  052101
4588  022070  050040  044522  051117
4589  022076  052111  020131  000067
4590
4591
4592
4593
4594
4595  022104  046123  030525  051040  EM106:  .ASCIZ  /SLU1 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR/
4596  022112  053103  020122  047111
4597  022120  042524  051122  050125
4598  022126  051524  053440  052111
4599  022134  020110  047111  042524
4600  022142  051122  050125  020124
4601  022150  047105  041101  042514
4602  022156  041440  042514  051101
4603  022164     000
4604
4605
4606
4607
4608
4609
4610
4611
4612  022165     123  052514  020061  EM107:  .ASCIZ  /SLU1 RCVR DID NOT INTERRUPT/
4613  022172  041522  051126  042040
4614  022200  042111  047040  052117
4615  022206  044440  052116  051105
4616  022214  052522  052120     000
4617
4618
4619
4620
4621  022221     123  052514  020061  EM110:  .ASCIZ  /SLU1 RECEIVER RE-INTERRUPTED/
4622  022226  042522  042503  053111
4623  022234  051105  051040  026505
4624  022242  047111  042524  051122
4625  022250  050125  042524  000104
4626
4627
4628
4629
4630  022256  046123  030525  051040  EM111:  .ASCIZ  /SLU1 READING RBUF DID NOT CLEAR INTERRUPT/
4631  022264  040505  044504  043516
4632  022272  051040  052502  020106
4633  022300  044504  020104  047516
```

```
 4634  022306  020124  046103  040505
 4635  022314  020122  047111  042524
 4636  022322  051122  050125  000124
 4637
 4638
 4639
 4640
 4641
 4642
 4643  022330  042522  042523  020124  EM112:  .ASCIZ  /RESET DID NOT CLEAR SLU1 RCVR INTERRUPT/
 4644  022336  044504  020104  047516
 4645  022344  020124  046103  040505
 4646  022352  020122  046123  030525
 4647  022360  051040  053103  020122
 4648  022366  047111  042524  051122
 4649  022374  050125  000124
 4650
 4651
 4652
 4653
 4654
 4655
 4656  022400  046123  030525  023440  EM113:  .ASCIZ  /SLU1 'OR' FLAG DID NOT SET/
 4657  022406  051117  020047  046106
 4658  022414  043501  042040  042111
 4659  022422  047040  052117  051440
 4660  022430  052105     000
 4661
 4662
 4663
 4664
 4665  022433     123  052514  020061  EM114:  .ASCIZ  /SLU1 'ERROR' NOT SET WITH 'OR' FLAG/
 4666  022440  042447  051122  051117
 4667  022446  020047  047516  020124
 4668  022454  042523  020124  044527
 4669  022462  044124  023440  051117
 4670  022470  020047  046106  043501
 4671  022476     000
 4672
 4673
 4674
 4675
 4676
 4677
 4678  022477     104  052101  020101  EM115:  .ASCIZ  /DATA COMPARE ERROR/
 4679  022504  047503  050115  051101
 4680  022512  020105  051105  047522
 4681  022520  000122
 4682
 4683
 4684
 4685  022522  046123  031125  054040  EM116:  .ASCIZ  /SLU2 XMIT INTERRUPTS WHEN DISABLED/
 4686  022530  044515  020124  047111
 4687  022536  042524  051122  050125
 4688  022544  051524  053440  042510
 4689  022552  020116  044504  040523
```

I7

SEQ 008+


```
 4690  022560  046102  042105     000
 4691
 4692
 4693
 4694
 4695
 4696  022565     123  052514  020062  EM117:  .ASCIZ  /SLU2 XMIT DID NOT INTERRUPT/
 4697  022572  046530  052111  042040
 4698  022600  042111  047040  052117
 4699  022606  044440  052116  051105
 4700  022614  052522  052120     000
 4701
 4702
 4703
 4704
 4705  022621     123  052514  020062  EM120:  .ASCIZ  /SLU2 XMIT INTERRUPT AT PRIORITY 7/
 4706  022626  046530  052111  044440
 4707  022634  052116  051105  052522
 4708  022642  052120  040440  020124
 4709  022650  051120  047511  044522
 4710  022656  054524  033440     000
 4711
 4712
 4713
 4714
 4715
 4716  022663     123  052514  020062  EM121:  .ASCIZ  /SLU2 XMIT INTERRUPTS WITH ENABLE CLEAR/
 4717  022670  046530  052111  044440
 4718  022676  052116  051105  052522
 4719  022704  052120  020123  044527
 4720  022712  044124  042440  040516
 4721  022720  046102  020105  046103
 4722  022726  040505  000122
 4723
 4724
 4725
 4726
 4727
 4728
 4729  022732  046123  031125  054040  EM122:  .ASCIZ  /SLU2 XMIT DID NOT INTERRUPT/
 4730  022740  044515  020124  044504
 4731  022746  020104  047516  020124
 4732  022754  047111  042524  051122
 4733  022762  050125  000124
 4734
 4735
 4736
 4737
 4738  022766  046123  031125  054040  EM123:  .ASCIZ  /SLU2 XMIT RE-INTERRUPTED/
 4739  022774  044515  020124  042522
 4740  023002  044455  052116  051105
 4741  023010  052522  052120  042105
 4742  023016     000
 4743
 4744
 4745
```

J7


```
 4746
 4747  023017     114  040517  044504  EM124:  .ASCIZ  /LOADING SLU2 TBUF DID NOT CLEAR INTERRUPT/
 4748  023024  043516  051440  052514
 4749  023032  020062  041124  043125
 4750  023040  042040  042111  047040
 4751  023046  052117  041440  042514
 4752  023054  051101  044440  052116
 4753  023062  051105  052522  052120
 4754  023070     000
 4755
 4756
 4757
 4758
 4759
 4760
 4761
 4762  023071     123  052514  020062  EM125:  .ASCIZ  /SLU2 RCVR INTERRUPTS WITH ENABLE CLEAR/
 4763  023076  041522  051126  044440
 4764  023104  052116  051105  052522
 4765  023112  052120  020123  044527
 4766  023120  044124  042440  040516
 4767  023126  046102  020105  046103
 4768  023134  040505  000122
 4769
 4770
 4771
 4772
 4773
 4774
 4775  023140  046123  031125  051040  EM126:  .ASCIZ  /SLU2 RCVR DID NOT INTERRUPT/
 4776  023146  053103  020122  044504
 4777  023154  020104  047516  020124
 4778  023162  047111  042524  051122
 4779  023170  050125  000124
 4780
 4781
 4782
 4783
 4784  023174  046123  031125  051040  EM127:  .ASCIZ  /SLU2 RCVR INTERRUPTS AT PRIORITY 7/
 4785  023202  053103  020122  047111
 4786  023210  042524  051122  050125
 4787  023216  051524  040440  020124
 4788  023224  051120  047511  044522
 4789  023232  054524  033440     000
 4790
 4791
 4792
 4793
 4794
 4795  023237     123  052514  020062  EM130:  .ASCIZ  /SLU2 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR/
 4796  023244  041522  051126  044440
 4797  023252  052116  051105  052522
 4798  023260  052120  020123  044527
 4799  023266  044124  044440  052116
 4800  023274  051105  052522  052120
 4801  023302  042440  040516  046102
```

K7


```
4802  023310  020105  046103  040505
4803  023316  000122
4804
4805
4806
4807
4808
4809
4810
4811
4812  023320  046123  031125  051040  EM131:  .ASCIZ  /SLU2 RCVR DID NOT INTERRUPT/
4813  023326  053103  020122  044504
4814  023334  020104  047516  020124
4815  023342  047111  042524  051122
4816  023350  050125  000124
4817
4818
4819
4820
4821  023354  046123  031125  051040  EM132:  .ASCIZ  /SLU2 RECEIVER RE-INTERRUPTED/
4822  023362  041505  044505  042526
4823  023370  020122  042522  044455
4824  023376  052116  051105  052522
4825  023404  052120  042105     000
4826
4827
4828
4829
4830  023411     123  052514  020062  EM133:  .ASCIZ  /SLU2 READING RBUF DID NOT CLEAR INTERRUPT/
4831  023416  042522  042101  047111
4832  023424  020107  041122  043125
4833  023432  042040  042111  047040
4834  023440  052117  041440  042514
4835  023446  051101  044440  052116
4836  023454  051105  052522  052120
4837  023462     000
4838
4839
4840
4841
4842
4843
4844
4845  023463     122  051505  052105  EM134:  .ASCIZ  /RESET DID NOT CLEAR SLU2 RCVR INTERRUPT/
4846  023470  042040  042111  047040
4847  023476  052117  041440  042514
4848  023504  051101  051440  052514
4849  023512  020062  041522  051126
4850  023520  044440  052116  051105
4851  023526  052522  052120     000
4852
4853
4854
4855
4856
4857
```

```
 4858  023533     123  052514  020062  EM135:  .ASCIZ  /SLU2 'OR' FLAG DID NOT SET/
 4859  023540  047447  023522  043040
 4860  023546  040514  020107  044504
 4861  023554  020104  047516  020124
 4862  023562  042523  000124
 4863
 4864
 4865
 4866
 4867  023566  046123  031125  023440  EM136:  .ASCIZ  /SLU2 'ERROR' NOT SET WITH 'OR' FLAG/
 4868  023574  051105  047522  023522
 4869  023602  047040  052117  051440
 4870  023610  052105  053440  052111
 4871  023616  020110  047447  023522
 4872  023624  043040  040514  000107
 4873
 4874
 4875
 4876
 4877
 4878  023632  046123  031125  041040  EM137:  .ASCIZ  /SLU2 BREAK DID NOT TRANSMIT ALL ZEROES/
 4879  023640  042522  045501  042040
 4880  023646  042111  047040  052117
 4881  023654  052040  040522  051516
 4882  023662  044515  020124  046101
 4883  023670  020114  042532  047522
 4884  023676  051505     000
 4885
 4886
 4887
 4888
 4889
 4890
 4891  023701     102  042522  045501  EM140:  .ASCIZ  /BREAK DID NOT SET FRAMING ERROR/
 4892  023706  042040  042111  047040
 4893  023714  052117  051440  052105
 4894  023722  043040  040522  044515
 4895  023730  043516  042440  051122
 4896  023736  051117     000
 4897
 4898
 4899
 4900
 4901
 4902  023741     123  052514  020062  EM141:  .ASCIZ  /SLU2 'ERROR' NOT SET WITH 'FR' FLAG/
 4903  023746  042447  051122  051117
 4904  023754  020047  047516  020124
 4905  023762  042523  020124  044527
 4906  023770  044124  023440  051106
 4907  023776  020047  046106  043501
 4908  024004     000
 4909
 4910
 4911
 4912
 4913
```

M7


```
 4914
 4915  024005     104  052101  020101  EM142:  .ASCIZ  /DATA COMPARE ERROR WITH CABLE/
 4916  024012  047503  050115  051101
 4917  024020  020105  051105  047522
 4918  024026  020122  044527  044124
 4919  024034  041440  041101  042514
 4920  024042     000
 4921
 4922
 4923
 4924
 4925
 4926  024043     122  041524  044440  EM143:  .ASCIZ  /RTC INTERRUPT AT PRIORITY 7/
 4927  024050  052116  051105  052522
 4928  024056  052120  040440  020124
 4929  024064  051120  047511  044522
 4930  024072  054524  033440     000
 4931
 4932
 4933
 4934
 4935  024077     122  041524  044440  EM144:  .ASCIZ  /RTC INTERRUPTS WHEN DISABLED/
 4936  024104  052116  051105  052522
 4937  024112  052120  020123  044127
 4938  024120  047105  042040  051511
 4939  024126  041101  042514  000104
 4940
 4941
 4942
 4943
 4944  024134  052122  020103  047111  EM145:  .ASCIZ  /RTC INTERRUPT DID NOT OCCUR/
 4945  024142  042524  051122  050125
 4946  024150  020124  044504  020104
 4947  024156  047516  020124  041517
 4948  024164  052503  000122
 4949
 4950
 4951
 4952
 4953  024170  052122  020103  047111  EM146:  .ASCIZ  /RTC INTERRUPT DID NOT OCCUR/
 4954  024176  042524  051122  050125
 4955  024204  020124  044504  020104
 4956  024212  047516  020124  041517
 4957  024220  052503  000122
 4958
 4959
 4960
 4961
 4962  024224  052122  020103  047504  EM147:  .ASCIZ  /RTC DOUBLE INTERRUPT/
 4963  024232  041125  042514  044440
 4964  024240  052116  051105  052522
 4965  024246  052120     000
 4966
 4967
 4968
 4969  024251     122  051505  052105  EM150:  .ASCIZ  /RESET DID NOT CLEAR RTC INTERRUPT/
```

```
 4970  024256  042040  042111  047040
 4971  024264  052117  041440  042514
 4972  024272  051101  051040  041524
 4973  024300  044440  052116  051105
 4974  024306  052522  052120     000
 4975
 4976
 4977
 4978
 4979
 4980  024313     122  041524  044440  EM151:  .ASCIZ  /RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS/
 4981  024320  052116  051105  052522
 4982  024326  052120  042040  042111
 4983  024334  047040  052117  041440
 4984  024342  042514  051101  053440
 4985  024350  052111  020110  044502
 4986  024356  033524  047440  020106
 4987  024364  045514  000123
 4988
 4989
 4990
 4991
 4992
 4993
 4994
 4995  024370  046103  041517  020113  EM152:  .ASCIZ  /CLOCK REPEATABILITY ERROR/
 4996  024376  042522  042520  052101
 4997  024404  041101  046111  052111
 4998  024412  020131  051105  047522
 4999  024420  000122
 5000
 5001
 5002
 5003
 5004  024422  046123  030525  051040  EM153:  .ASCIZ  /SLU1 RECEIVER STATUS ERROR/
 5005  024430  041505  044505  042526
 5006  024436  020122  052123  052101
 5007  024444  051525  042440  051122
 5008  024452  051117     000
 5009
 5010
 5011
 5012
 5013  024455     123  052514  020062  EM154:  .ASCIZ  /SLU2 RECEIVER STATUS ERROR/
 5014  024462  042522  042503  053111
 5015  024470  051105  051440  040524
 5016  024476  052524  020123  051105
 5017  024504  047522  000122
 5018
 5019
 5020
 5021
 5022  024510  047111  047503  051122  EM155:  .ASCIZ  /INCORRECT RECEIVE COUNT SLU1/
 5023  024516  041505  020124  042522
 5024  024524  042503  053111  020105
 5025  024532  047503  047125  020124
```

```
 5026  024540  046123  030525     000
 5027
 5028
 5029
 5030
 5031  024545     123  052514  020061  EM156:  .ASCIZ  /SLU1 DATA COMPARE ERROR IN EXERCISER/
 5032  024552  040504  040524  041440
 5033  024560  046517  040520  042522
 5034  024566  042440  051122  051117
 5035  024574  044440  020116  054105
 5036  024602  051105  044503  042523
 5037  024610  000122
 5038
 5039
 5040
 5041
 5042
 5043
 5044  024612  047111  047503  051122  EM157:  .ASCIZ  /INCORRECT RECEIVE COUNT SLU2/
 5045  024620  041505  020124  042522
 5046  024626  042503  053111  020105
 5047  024634  047503  047125  020124
 5048  024642  046123  031125     000
 5049
 5050
 5051
 5052
 5053  024647     123  052514  020062  EM160:  .ASCIZ  /SLU2 DATA COMPARE ERROR IN EXERCISER/
 5054  024654  040504  040524  041440
 5055  024662  046517  040520  042522
 5056  024670  042440  051122  051117
 5057  024676  044440  020116  054105
 5058  024704  051105  044503  042523
 5059  024712  000122
 5060
 5061
 5062
 5063
 5064
 5065
 5066  024714  051124  050101  041440  EM161:  .ASCIZ  /TRAP CATCHER/
 5067  024722  052101  044103  051105
 5068  024730     000
 5069
 5070
 5071
 5072  024731     124  051505  021524  DH5:    .ASCIZ  /TEST#   ERR PC  TCSR/
 5073  024736  020040  042440  051122
 5074  024744  050040  020103  052040
 5075  024752  051503  000122
 5076
 5077
 5078
 5079  024756  042524  052123  020043  DH6:    .ASCIZ  /TEST#   ERR PC  RCSR/
 5080  024764  020040  051105  020122
 5081  024772  041520  020040  041522
```

```
 5082  025000  051123     000
 5083
 5084
 5085
 5086  025003     124  051505  021524  DH7:    .ASCIZ  /TEST#   ERR PC  RBUF/
 5087  025010  020040  042440  051122
 5088  025016  050040  020103  051040
 5089  025024  052502  000106
 5090
 5091
 5092
 5093  025030  042524  052123  020043  DH32:   .ASCIZ  /TEST#   ERR PC  TBUF/
 5094  025036  020040  051105  020122
 5095  025044  041520  020040  041124
 5096  025052  043125     000
 5097
 5098
 5099
 5100  025055     124  051505  021524  DH61:   .ASCIZ  /TEST#   ERR PC  LKS/
 5101  025062  020040  042440  051122
 5102  025070  050040  020103  046040
 5103  025076  051513     000
 5104
 5105
 5106
 5107  025101     124  051505  021524  DH72:   .ASCIZ  /TEST#   ERR PC  GDADR   BDADR   GDDAT   BDDAT/
 5108  025106  020040  042440  051122
 5109  025114  050040  020103  043440
 5110  025122  040504  051104  020040
 5111  025130  041040  040504  051104
 5112  025136  020040  043440  042104
 5113  025144  052101  020040  041040
 5114  025152  042104  052101     000
 5115
 5116
 5117
 5118
 5119
 5120
 5121
 5122  025157     124  051505  021524  DH115:  .ASCIZ  /TEST#   ERR PC  RCSR    GOOD    BAD/
 5123  025164  020040  042440  051122
 5124  025172  050040  020103  051040
 5125  025200  051503  020122  020040
 5126  025206  043440  047517  020104
 5127  025214  020040  041040  042101
 5128  025222     000
 5129
 5130
 5131
 5132
 5133
 5134
 5135  025223     124  051505  021524  DH137:  .ASCIZ  /TEST#   ERR PC  RCSR    DATA/
 5136  025230  020040  042440  051122
 5137  025236  050040  020103  051040
```

```
5138  025244  051503  020122  020040
5139  025252  042040  052101  000101
5140
5141
5142
5143
5144  025260  042524  052123  020043  DH142:  .ASCIZ  /TEST#   ERR PC  GOOD    BAD/
5145  025266  020040  051105  020122
5146  025274  041520  020040  047507
5147  025302  042117  020040  020040
5148  025310  040502  000104
5149
5150
5151
5152
5153  025314  042524  052123  020043  DH152:  .ASCIZ  /TEST#   ERR PC  LKS     CNT1    CNT2/
5154  025322  020040  051105  020122
5155  025330  041520  020040  045514
5156  025336  020123  020040  020040
5157  025344  047103  030524  020040
5158  025352  020040  047103  031124
5159  025360     000
5160
5161
5162
5163
5164
5165
5166  025361     124  051505  021524  DH155:  .ASCIZ  /TEST#   ERR PC  RCSR    TRANS   RCV/
5167  025366  020040  042440  051122
5168  025374  050040  020103  051040
5169  025402  051503  020122  020040
5170  025410  052040  040522  051516
5171  025416  020040  051040  053103
5172  025424     000
5173
5174
5175
5176
5177
5178
5179  025425     124  051505  021524  DH161:  .ASCIZ  /TEST#   ERR PC  RCSR    OLDPC   TRAP ADR/
5180  025432  020040  042440  051122
5181  025440  050040  020103  051040
5182  025446  051503  020122  020040
5183  025454  047440  042114  041520
5184  025462  020040  052040  040522
5185  025470  020120  042101  000122
5186
5187
5188
5189
5190
5191
5192  025476  005015  045103  042113  M1:     .ASCIZ  <CR><LF>/CJKDI-B/
5193  025504  026511  000102
```

```
5194
5195  025510  200                       M2:     .ASCII  <CRLF>
5196
5197          025512                            .EVEN
5198  025512  001104  001016  002764    DT5:    .WORD   $TESTN,$ERRPC,CTCSR,0
5199  025520  000000
5200
5201  025522  001104  001016  002760    DT6:    .WORD   $TESTN,$ERRPC,CRCSR,0
5202  025530  000000
5203
5204  025532  001104  001016  002762    DT7:    .WORD   $TESTN,$ERRPC,CRBUF,0
5205  025540  000000
5206
5207  025542  001104  001016  002754    DT31:   .WORD   $TESTN,$ERRPC,TCSR,0
5208  025550  000000
5209
5210  025552  001104  001016  002756    DT32:   .WORD   $TESTN,$ERRPC,TBUF,0
5211  025560  000000
5212
5213  025562  001104  001016  002750    DT36:   .WORD   $TESTN,$ERRPC,RCSR,0
5214  025570  000000
5215
5216  025572  001104  001016  002752    DT37:   .WORD   $TESTN,$ERRPC,RBUF,0
5217  025600  000000
5218
5219  025602  001104  001016  002770    DT61:   .WORD   $TESTN,$ERRPC,LKS,0
5220  025610  000000
5221
5222  025612  001104  001016  001020    DT72:   .WORD   $TESTN,$ERRPC,$GDADR,$BDADR,$GDDAT,$BDDAT,0
5223  025620  001022  001024  001026
5224  025626  000000
5225
5226
5227  025630  001104  001016  002760    DT115:  .WORD   $TESTN,$ERRPC,CRCSR,$GDDAT,$BDDAT,0
5228  025636  001024  001026  000000
5229
5230  025644  001104  001016  002750    DT137:  .WORD   $TESTN,$ERRPC,RCSR,$BDDAT,0
5231  025652  001026  000000
5232
5233  025656  001104  001016  001024    DT142:  .WORD   $TESTN,$ERRPC,$GDDAT,$BDDAT,0
5234  025664  001026  000000
5235
5236  025670  001104  001016  002750    DT157:  .WORD   $TESTN,$ERRPC,RCSR,XMTCT2,RECCT2,0
5237  025676  012232  012234  000000
5238
5239  025704  001104  001016  002750    DT161:  .WORD   $TESTN,$ERRPC,RCSR,OLDPC,BDVECT,0
5240  025712  012414  012416  000000
5241
5242  025720  000200                    BUF1:   .BLKW   200             ;SLU1 INPUT BUFFER FOR BLAST TEST
5243  026320  000200                    BUF2:   .BLKW   200             ;SLU2 INPIT BUFFER FOR BLAST TEST
5244          000001                            .END
```

F8


```
ABASE = 176500            777#    964
ACPUOP= 000000            795#    938
ADEVCT= 000000            788#    929
ADEVM = 000000            809#    965
AENV  = 000000            792#    934
AENVM = 000000            793#    935
AFATAL= 000000            785#    926
AMADR1= 000000            798#    951
AMADR2= 000000            801#    955
AMADR3= 000000            804#    958
AMADR4= 000000            807#    961
AMAMS1= 000000            796#    945
AMAMS2= 000000            799#    953
AMAMS3= 000000            802#    956
AMAMS4= 000000            805#    959
AMSGAD= 000000            790#    931
AMSGLG= 000000            791#    932
AMSGTY= 000000            784#    925
AMTYP1= 000000            797#    946
AMTYP2= 000000            800#    954
AMTYP3  000000            803#    957
AMTYP4= 000000            806#    960
APASS = 000000            787#    928
APTCSU= 000040           3341#   3374
APTENV= 000001           3116    3296   3339#   3367
APTSIZ= 000200           1589    3338#
APTSPO= 000100           3298    3340#  33( 9
ASWREG= 000000            794#    936
ATESTN= 000000            786#    927
AUNIT = 000000            789#    930
AUSWR = 000400            779#    937
AVECT1= 000300            778#    962
AVECT2= 000000            808#    963
BDVECT  012416           3082*   3088#  5239
BIT0  = 000001            762#   1606   1613   1700   1961   1962   1965   1966   1969   1970   1979   1981   1983
                         2263    2714   2721   2726   2730   2740   2744   2932
BIT00 = 000001            752#    762
BIT01 = 000002            751#    761
BIT02 = 000004            750#    760
BIT03 = 000010            749#    759
BIT04 = 000020            748#    758
BIT05 = 000040            747#    757
BIT06 = 000100            746#    756
BIT07 = 000200            745#    755
BIT08 = 000400            744#    754
BIT09 = 001000            743#    753   3127   3267
BIT1  = 000002            761#   1609   1846   2390
BIT10 = 002000            742#   2685   2738
BIT11 = 004000            741#
BIT12 = 010000            740#   3796
BIT13 = 020000            739#   2745   3111
BIT14 = 040000            738#   2693   3251
BIT15 = 100000            737#   2698   2750
BIT2  = 000004            760#   1612   2136   2796
BIT3  = 000010            759#   1607
BIT4  = 000020            758#   1604   3784   3857
```

```
BIT5  = 000040   757# 1604 1607
BIT6  = 000100   756# 1610 1755 1760 1770 1771 1777 1778 1784 1786 1799 1804 1814
                1815 1821 1822 1829 1831 1992 1997 2007 2008 2014 2015 2025 2027
                2040 2045 2055 2056 2062 2063 2074 2076 2155 2160 2170 2171 2177
                2178 2187 2189 2225 2228 2268 2281 2286 2296 2303 2310 2328 2337
                2347 2361 2366 2374 2394 2407 2412 2422 2429 2436 2456 2465 2475
                2494 2499 2507 2527 2528 2543 2548 2565 2572 2599 2610 2629 2637
                2662 2670 2808 2836 2845 2869 2888 2909 2917 2955 2956 2959 2974
                2993 2994 2995
BIT7  = 000200   755# 2761 3811 3876
BIT8  = 000400   754#
BIT9  = 001000   753#
BLAST   011474  2798 2925#
BPT   = 000003   782#
BPTVEC= 000014   769#
BUF1    025720  5242#
BUF2    026320  2957 2983 3010 5243#
CATCH   012370   821 3080#
CHECK2  012120  3007#
CKSWR = 104407   657# 3126 3250
CONT41  010546  2717 2724#
COUNT   012236  2946* 2965* 2991 3030#
CR    = 000015   677# 3430 3829 3832 3835 3838 5192
CRBUF   002762  1527# 1739 5204
CRCSR   002760  1526# 1724 1799* 1804 1814* 1815 1821* 1822 1829* 1831 3788 3861 5201
                5227
CRLF  = 000200   678# 3384 5195
CRPSW   003004  1540#
CRVECT  003002  1539# 1800 1801* 1836* 2942 3024*
CTBUF   002766  1529# 1642 1661* 1666* 1704* 1707* 2367* 2370*
CTCSR   002764  1528# 1625 1662 1667 1673 1705 1710 1755* 1760 1770* 1771 1777* 1778
                1784* 1786 2268* 2271 2281* 2286* 2296* 2301 2303* 2310* 2328* 2335 2337*
                2347* 2361* 2366* 2368 2374* 2993* 5198
CTPSW   003010  1542# 2330 2332* 2354*
CTVECT  003006  1541# 1756 1757* 1791* 2269 2270* 2280* 2288* 2299 2300* 2311* 2320* 2329
                2331* 2343* 2353* 2364 2365* 2380* 2941 3023*
DADTBL  006066  2198# 2219 2220 2226
DDISP = 177570   684#  901 1577
DH115   025157  1344 1490 1501 5122#
DH137   025223  1434 5135#
DH142   025260  1449 1495 1511 5144#
DH152   025314  5153#
DH155   025361  1506 5166#
DH161   025425  1516 5179#
DH32    025030  1091 5093#
DH5     024731   990 1008 1016 1021 1026 1031 1036 1041 1086 1096 1101 1106 1121
                1126 1131 1136 1141 1146 1151 1156 1161 1259 1264 1269 1274 1279
                1284 1289 1349 1354 1359 1364 1369 1374 1379 5072#
DH6     024756   995 1046 1051 1056 1061 1066 1071 1076 1081 1111 1166 1171 1176
                1181 1186 1191 1196 1201 1294 1299 1304 1309 1314 1319 1324 1329
                1334 1339 1384 1389 1394 1399 1404 1409 1414 1419 1424 1429 1439
                1444 5079#
DH61    025055  1206 1211 1216 1221 1226 1231 1236 1241 1246 1454 1459 1464 1469
                1474 1479 1484 5100#
DH7     025003  1000 1116 5086#
DH72    025101  1251 5107#
```

H8


```
DISPLA  001042          901#  1577*  1585*  3106*  3278*  3783*  3856*
DISPRE  000174          831#  1585   3783   3856
DSWR  = 177570          683#   900   1576
DT115   025630         1345  1491    1502   5227#
DT137   025644         1435  5230#
DT142   025656         1450  1496    1512   5233#
DT157   025670         1507  5236#
DT161   025704         1517  5239#
DT31    025542         1087  1097    1102   1107   1122   1127   1132   1137   1142   1147   1152   1157   1162
                       1350  1355    1360   1365   1370   1375   1380   5207#
DT32    025552         1092  5210#
DT36    025562         1112  1167    1172   1177   1182   1187   1192   1197   1202   1385   1390   1395   1400
                       1405  1410    1415   1420   1425   1430   1445   5213#
DT37    025572         1117  5216#
DT5     025512          991  1009    1017   1022   1027   1032   1037   1042   1260   1265   1270   1275   1280
                       1285  1290    5198#
DT6     025522          996  1047    1052   1057   1062   1067   1072   1077   1082   1295   1300   1305   1310
                       1315  1320    1325   1330   1335   1340   1440   5201#
DT61    025602         1207  1212    1217   1222   1227   1232   1237   1242   1247   1455   1460   1465   1470
                       1475  1480    1485   5219#
DT7     025532         1001  5204#
DT72    025612         1252  5222#
ECHO    015406          836  3782#   3817
EMTVEC= 000030          772#  1561*  1562*
EM100   021577         1278  4531#
EM101   021633         1283  4540#
EM102   021664         1288  4549#
EM103   021736         1293  4562#
EM104   022005         1298  4575#
EM105   022041         1303  4584#
EM106   022104         1308  4595#
EM107   022165         1313  4612#
EM11    016150         1007  3918#
EM110   022221         1318  4621#
EM111   022256         1323  4630#
EM112   022330         1328  4643#
EM113   022400         1333  4656#
EM114   022433         1338  4665#
EM115   022477         1343  4678#
EM116   022522         1348  4685#
EM117   022565         1353  4696#
EM120   022621         1358  4705#
EM121   022663         1363  4716#
EM122   022732         1368  4729#
EM123   022766         1373  4738#
EM124   023017         1378  4747#
EM125   023071         1383  4762#
EM126   023140         1388  4775#
EM127   023174         1393  4784#
EM13    016206         1015  3927#
EM130   023237         1398  4795#
EM131   023320         1403  4812#
EM132   023354         1408  4821#
EM133   023411         1413  4830#
EM134   023463         1418  4845#
EM135   023533         1423  4858#
```

EM136	023566	1428	4867#
EM137	023632	1433	4878#
EM14	016254	1020	3940#
EM140	023701	1438	4891#
EM141	023741	1443	4902#
EM142	024005	1448	4915#
EM143	024043	1453	4926#
EM144	024077	1458	4935#
EM145	024134	1463	4944#
EM146	024170	1468	4953#
EM147	024224	1473	4962#
EM15	016324	1025	3953#
EM150	024251	1478	4969#
EM151	024313	1483	4980#
EM152	024370	4995#	
EM153	024422	1489	5004#
EM154	024455	1494	5013#
EM155	024510	5022#	
EM156	024545	1500	5031#
EM157	024612	1505	5044#
EM16	016372	1030	3966#
EM160	024647	1510	5053#
EM161	024714	1515	5066#
EM17	016430	1035	3975#
EM20	016470	1040	3986#
EM21	016536	1045	3999#
EM22	016606	1050	4012#
EM23	016652	1055	4023#
EM24	016710	1060	4032#
EM25	016750	1065	4043#
EM26	017023	1070	4058#
EM27	017060	1075	4067#
EM30	017123	1080	4078#
EM31	017175	1085	4093#
EM32	017234	1090	4104#
EM33	017273	1095	4115#
EM34	017345	1100	4130#
EM35	017413	1105	4143#
EM36	017455	1110	4154#
EM37	017514	1115	4165#
EM40	017553	1120	4176#
EM41	017623	1125	4189#
EM42	017661	1130	4200#
EM43	017721	1135	4211#
EM44	017767	1140	4224#
EM45	020037	1145	4237#
EM46	020105	1150	4250#
EM47	020143	1155	4261#
EM5	016010	989	3885#
EM50	020203	1160	4272#
EM51	020251	1165	4285#
EM52	020321	1170	4298#
EM53	020365	1175	4311#
EM54	020423	1180	4322#
EM55	020463	1185	4333#
EM56	020536	1190	4348#

J8


```
EM57    020573          1195    4357#
EM6     016052           994    3896#
EM60    020636          1200    4368#
EM61    020710          1205    4381#
EM62    020741          1210    4390#
EM63    021003          1215    4401#
EM64    021041          1220    4412#
EM65    021071          1225    4421#
EM66    021123          1230    4430#
EM67    021163          1235    4441#
EM7     016111           999    3907#
EM70    021223          1240    4452#
EM71    021255          1245    4461#
EM72    021306          1250    4470#
EM74    021367          1258    4487#
EM75    021432          1263    4498#
EM76    021466          1268    4507#
EM77    021530          1273    4518#
ERRVEC= 000004           765#   1574    1575*   1586*   3256    3257*   3259*   3262*
FINIE   012170          3021#
GTSWR = 104406           656#   1600
HT    = 000011           675#   3382
ID      003660          1615    1674    1683#
IOHAND  012036          2966    2975    2991#
IOTVEC= 000020           770#   1559*   1560*
LF    = 000012           676#   3434    3829    3832    3835    3838    5192
LKS     002770          1530#   2142    2155*   2160    2170*   2171    2177*   2178    2187*   2189    2808*   2817*   2836*
                        2845*   2869*   2888*   2909*   2917*   2959*   2995*   5219
LTCIT   010762          2392    2789#
LTCRT   005626          1848    2129#
MFR     015627          3805    3832#
MOR     015641          3810    3835#
MPAR    015616          3800    3829#
MSG     015570          3799    3804    3809    3814    3820#   3871
MSTOP   015654          3815    3838#
M1      025476          1690    5192#
M2      025510          1692    5195#
OLDPC   012414          3083*   3087#   5239
OUTTST  015664           837    3855#   3882
PIRQ  = 177772           682#
PIRQVE= 000240           776#
PR0   = 000000           699#
PR1   = 000040           700#
PR2   = 000100           701#
PR3   = 000140           702#
PR4   = 000200           703#
PR5   = 000240           704#
PR6   = 000300           705#
PR7   = 000340           706#
PS    = 177776           679#    680    2217*   2938*   2961*   2999*
PSW   = 177776           680#
PWRVEC= 000024           771#   1565*   1566*   3194*   3195*   3204*   3212*   3221*   3222*
RBUF    002752          1523#   1947    2090*   2111    2120*   2124    2516    2633    2693    2698    2713    2724    2745
                        2750    2777    2978    2980    2989    5216
RCSR    002750          1522#   1931    2040*   2045    2055*   2056    2062*   2063    2074*   2076    2093    2106    2122
                        2125    2528*   2534    2543*   2548*   2563    2565*   2572*   2597    2599*   2610*   2629*   2631
```

K8


```
                    2637*  2662*  2664   2670*  2711   2716   2742   2770   2956*  3786   3859   5213   5230
                    5236   5239
RDCHR = 104410       658#  3701
RDLIN = 104411       659#
RECCT2  012234      2945*  2977*  3007   3029#  5236
REC2    011762      2950   2977#
RESVEC= 000010       766#
RPSW    002774      1536#  2591   2593*  2611*  2951*
RTCPSW  003014      1544#  2804   2806*  2847*  2862   2864*  2890*  2953*
RTCVT   003012      1543#  2156   2157*  2194*  2803   2805*  2819*  2823*  2834*  2846*  2861   2863*  2878*
                    2889*  2905   2906*  2923*  2943   2952*  3025*
RVECT   002772      1535#  2041   2042*  2081*  2529   2530*  2542*  2550*  2560   2561*  2573*  2582*  2590
                    2592*  2606*  2617*  2627   2628*  2643*  2660   2661*  2677*  2940   2950*  3022*
SLU1IT  006262      2250#
SLU2IT  007016      2261   2265   2383#
SLU2RT  004402      1660   1702   1839#
STACK = 001100       670#
START   003016       835   1546#
STKLMT= 177774       681#
SWR     001040       900#  1555   1576*  1578   1584*  1591*  1598   1604   1607   1610   2685   2738   2761
                    3111   3123   3127   3202   3214*  3251   3267   3596   3633*  3782*  3784   3855*  3857
SWREG   000176       832#  1584   1598   3596   3609   3782   3855
SW0   = 000001       734#
SW00  = 000001       724#   734
SW01  = 000002       723#   733
SW02  = 000004       722#   732
SW03  = 000010       721#   731
SW04  = 000020       720#   730
SW05  = 000040       719#   729
SW06  = 000100       718#   728
SW07  = 000200       717#   727
SW08  = 000400       716#   726
SW09  = 001000       715#   725
SW1   = 000002       733#
SW10  = 002000       714#
SW11  = 004000       713#
SW12  = 010000       712#
SW13  = 020000       711#
SW14  = 040000       710#
SW15  = 100000       709#
SW2   = 000004       732#
SW3   = 000010       731#
SW4   = 000020       730#
SW5   = 000040       729#
SW6   = 000100       728#
SW7   = 000200       727#
SW8   = 000400       726#
SW9   = 001000       725#
TBITVE= 000014       767#
TBLEND  006076      2202#  2233
TBUF    002756      1525#  1867   1884*  1889*  1911*  1914*  2092*  2121*  2500*  2503*  2533*  2562*  2596*
                    2630*  2663*  2688*  2710*  2741*  2768*  2971*  5210
TCSR    002754      1524#  1852   1885   1890   1894   1912   1917   1961*  1962   1965*  1966   1969*  1970
                    1979*  1981   1983*  1992*  1997   2007*  2008   2014*  2015   2025*  2027   2394*  2397
                    2407*  2412*  2422*  2427   2429*  2436*  2456*  2463   2465*  2475*  2494*  2499*  2501
                    2507*  2527*  2689   2714*  2721*  2726*  2730*  2740*  2744*  2955*  2974*  2994*  5207
```

L8

SEQ 0102


```
TICKER  011716            2952    2965#
TKVEC = 000060             774#
TPSW    003000            1538#   2458    2460*   2482*   2949*
TPVEC = 000064             775#
TRAPVE= 000034             773#   1563*   1564*
TRTVEC= 000014             768#
TST1    003430            1614    1621#   3065
TST10   004236            1797#
TST11   004402            1844#
TST12   004456            1863#
TST13   004516            1877#
TST14   004610            1882    1904#
TST15   004670            1909    1927#
TST16   004730            1943#
TST17   004770            1895    1959#
TST2    003476            1638#
TST20   005126            1918    1977    1982    1990#
TST21   005310            2038#
TST22   005472            2088#
TST23   005564            2118#
TST24   005626            2126    2134#
TST25   005702            2153#
TST26   006100            2210#
TST27   006262            2215    2255#
TST3    003544            1654#
TST30   006430            2294#
TST31   006550            2326#
TST32   006704            2359#
TST33   007016            2388#
TST34   007140            2420#
TST35   007260            2454#
TST36   007414            2487#
TST37   007556            2492    2525#
TST4    003714            1680    1698#
TST40   007676            2556#
TST41   010014            2588#
TST42   010146            2623#
TST43   010252            2650#
TST44   010376            2655    2683#
TST45   010464            2686    2708#
TST46   010572            2736#
TST47   010664            2739    2759#
TST5    003772            1711    1720#
TST50   010762            2762    2781    2794#
TST51   011210            2854#
TST52   011356            2859    2896#
TST53   011474            2901    2930#
TST6    004032            1735#
TST7    004072            1753#
TVECT   002776            1537#   1993    1994*   2032*   2395    2396*   2406*   2414*   2425    2426*   2437*   2446*   2457
                          2459*   2471*   2481*   2497    2498*   2517*   2939    2948*   3021*
TYPDS = 104405             655#   3054
TYPE  = 104401             651#   1689    1691    3052    3055    3114    3149    3166    3168    3171    3173    3177    3184
                          3226    3387    3500    3575    3607    3608    3611    3624    3635    3654    3705    3708    3712
TYPOC = 104402             652#   3157    3181    3610
TYPON = 104404             654#
```

M3


```
TYPOS = 104403          653#
UNIQUE  006100          2138    2195    2205#
WAIT    015576          3822#   3823    3826
WAITER  012074          2996    2999#
WAITIO  011712          2962#   2963
WDONE   005514          2093#   2096
WRPSW   012356          1758    1802    1995    2043    2158    2273    2297    2312    2333    2344    2362    2371    2399
                        2423    2438    2461    2472    2495    2504    2531    2558    2574    2594    2607    2625    2634
                        2658    2667    2801    2820    2865    2879    2903    2914    3073#
XMIT2   011726          2948    2969#
XMTCT2  012232          2944*   2969*   2972    3007    3012    3028#   5236
XXX     011110          2819    2823#
$APTHD  000500           859     865#
$ATYC   013312          3289    3291#
$ATY1   013266          3287#
$ATY3   013274          3288#   3372
$ATY4   013304          1628    1645    1669    1677    3119    3290#
$AUTOB  001034           897#   1602*   3604
$BASE   001154           964#
$BDADR  001022           892#   2239*   5222
$BDDAT  001026           894#   2241*   2784*   2980*   3015*   5222    5227    5230    5233
$CHARC  014064          3389*   3399*   3406    3432*   3437#
$CKSWR  014542          3596#   3771
$CMTAG  001000           880#   1553
$CNTLG  015277          3607    3722#
$CNTLU  015272          3624    3721#
$CPUOP  001126           938#
$CRLF   001075           917#   3114    3149    3168    3173    3177    3388    3635    3873
$DBLK   014532          3541    3575    3583#
$DEVCT  001110           929#   1687
$DEVM   001156           965#   1549*   1606*   1609*   1612*   1613    1700    1846    2136    2263    2390    2796
$DOAGN  012332          3048    3057    3063#
$DTBL   014522          3544    3579#
$ENDAD  012322           846    3059#   3134
$ENDCT  012270          1567    3050#
$ENDMG  012341          3052    3067#
$ENULL  012336          3055    3066#
$ENV    001120           934#   1596    1656    1879    1906    1974    2020    2069    2100    2182    2212    2257    2489
                        2652    2856    2898    2932    3116    3296    3320    3367
$ENVM   001121           935#   1589    3298    3369    3374
$EOP    012240          2936    3026    3041#
$EOPCT  012262          1567*   3047#   3051
$ERFLG  001003           883#   3104*   3265    3271*
$ERMAX  001015           889#   1569*   3277*
$ERROR  012420          1561    3103#
$ERRPC  001016           890#   3108*   3109*   3110    3155    5198    5201    5204    5207    5210    5213    5216    5219
                        5222    5227    5230    5233    5236    5239
$ERRTB  001160           982#   3163
$ERRTY  012602          3113    3148#
$ERTTL  001012           887#   3107*
$ESCAP  001072           915#   1568*   3130    3132    3276*
$ETABL  001120           933#
$ETEND  001160           871     966#
$FATAL  001102           926#   1546*   3324*
$FFLG   013532          3287*   3290*   3318    3327*   3335#
$FILLC  001056           908#   3392
```

N8


```
$FILLS  001055          907#
$GDADR  001020          891#  2238*  5222
$GDDAT  001024          893#  2240*  2783*  2985*  2987*  3016*  5222  5227  5233
$GET42  012312         3056#
$GTSWR  014612         3608#  3769
$HIBTS  000500          866#
$ICNT   001004          884#
$ILLUP  013076         3194   3212   3229#
$INTAG  001035          898#  3636
$ITEMB  001014          888#  3110*  3118  3152
$LF     001076          918#  3712
$LFLG   013531         3328*  3334#
$LPADR  001006          885#  1570*  3269*  3274*  3279
$LPERR  001010          886#  1571*  3129  3269  3275*
$MADR1  001132          951#
$MADR2  001136          955#
$MADR3  001142          958#
$MADR4  001146          961#
$MAIL   001100          867    871    924#
$MAMS1  001130          945#
$MAMS2  001134          953#
$MAMS3  001140          956#
$MAMS4  001144          959#
$MBADR  000502          867#
$MFLG   013530         3288*  3294   3329*  3333#
$MNEW   015315         3611   3726#
$MSGAD  001114          931#  3304*  3307
$MSGLG  001116          932#  3309*
$MSGTY  001100          925#  1547*  3302  3310*  3322  3325*
$MSWR   015304         3608   3723#
$MTYP1  001131          946#
$MTYP2  001135          954#
$MTYP3  001141          957#
$MTYP4  001145          960#
$NULL   001054          906#  3394
$OCNT   014312         3472*  3501*  3514#
$OMODE  014314         3467*  3471*  3476  3479*  3490*  3516#
$OVER   013252         3252   3270   3278#
$PASS   001106          928#  1588*  1658  1685  1881  1908  1976  2022  2071  2102  2184  2214  2259
                       2491   2654   2858  2900  2934  3044*  3045*  3053
$PASTM  000506          869#
$POWER  013104         3227   3232#
$PWRON  012736         1565   3194#  3221
$PWRMG  013072         3227#
$PWRUP  013010         3204   3212#
$QUES   001074          916#  3654   3705
$RDCHR  015024         3667#  3772
$RDLIN  015154         3697#  3773
$RTNAD  012334         3065#
$SAVR6  013102         3203*  3213   3223*  3224*  3231#
$SCOPE  013114         1559   3249#
$SETUP= 000137          649#
$STUP = 177777          649#
$SVLAD  013216         3260   3266   3272#
$SVPC = 000500          844#   849
$SWR  = 161000          663#   781#
```

B9


```
$SWREG  001122          936#  1591
$SWRMR= 000000          664#
$TESTN  001104          927#  1548*  1622*  1639*  1655*  1699*  1721*  1736*  1754*  1798*  1845*  1864*  1878*
                       1905*  1928*  1944*  1960*  1991*  2039*  2089*  2119*  2135*  2154*  2211*  2256*  2295*
                       2327*  2360*  2389*  2421*  2455*  2488*  2526*  2557*  2589*  2624*  2651*  2684*  2709*
                       2737*  2760*  2795*  2855*  2897*  2931*  3273*  5198   5201   5204   5207   5210   5213
                       5216   5219   5222   5227   5230   5233   5236   5239
$TKB    001046          903#  3413   3420   3600   3617   3671   3677
$TKS    001044          902#  3411   3418   3598   3614   3638*  3669   3675
$TMP0   001060          910#  2939*  3021
$TMP1   001062          911#  2940*  3022
$TMP2   001064          912#  2941*  3023
$TMP3   0C1066          913#  2942*  3024
$TMP4   001070          914#  2943*  3025
$TN   = 000001          662#   665#   780#
$TPB    001052          905#  3429*
$TPFLG  001057          909#  3361
$TPS    001050          904#  3427
$TRAP   015326         1563   3740#
$TRAP2  015350         3751#  3762
$TRPAD  015362         3745   3762#
$TSTM   000504          868#
$TSTNM  001002          882#  3043*  3106   3272*  3273   3278
$TTYIN  015262         3698   3699   3716   3720#
$TYPDS  014316         3529#  3767
$TYPE   013534         3315   3361#  3763
$TYPEC  013746         3391   3398   3405   3410#  3640
$TYPEX  014066         3433   3435   3438#
$TYPOC  014114         3470#  3764
$TYPON  014130         3469   3472#  3766
$TYPOS  014070         3465#  3765
$UNIT   001112          930#
$UNITM  000510          870#
$USWR   001124          937#
$VECT1  001150          962#
$VECT2  001152          963#
$XOFF = 000023          661#  3415
$XON  = 000021          660#  3422   3682
$XTSTR  013126         3254#
$OFILL  014313         3466*  3470*  3480   3515#
.     = 026720          813#   820#   824#   830#   834#   839#   844    845#   847#   849#   855    856#   858#
                        860#   879#   985#  1004#  1012#  1255#  1556   1570   1571   3187#  3206   3230   3337#
                       3583#  3720#  3843#  5197#  5242#  5243#
.$ERRT  001160          984#
.$X   = 000500          855#   860

. ABS.  026720     000

 ERRORS DETECTED:  0

 JKDIB0,JKDIB0/SOL/NL:TOC/CRF:SYM=SYSMAC.SML,JKDIB0.P11
 RUN-TIME: 10 15 1 SECONDS
 RUN-TIME RATIO: 60/26=2.3
 CORE USED:  36K  (72 PAGES)
```

C9