```
                .REM

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

PRODUCT CODE:   AC-8528I-MC

PRODUCT NAME:   CZDLDIO DL11-W/1144 MFM SLU

DATE CREATED:   JULY, 1981

MAINTAINER:     DIAGNOSTIC ENGINEERING

AUTHOR:         DAN CASALETTO

REVISED BY:     DAN MILLEVILLE
```

```
39
40
41                              HISTORY SECTION
42                              ------- -  --
43
44
45                              CZDLDDO WAS RELEASED OCT 79
46
47                              THE FOLLOWING CHANGES WERE MADE. ALL CHANGES ARE INDICATED
48                              BY ;** IN THE COMMENT FIELD:
49
50                                      1. USE THE MFPT INSTRUCTION,AND IF THE CPU IS A 11/44:
51
52                                              A. DO NOT PERFORM READER ENABLE AND RECEIVER ACTIVE
53                                                 TESTS.
54                                              B. IF ERROR FLAGS TESTS AND BREAK TESTS ARE
55                                                 ENABLED,PERFORM THESE TESTS ONLY FOR THE SLU
56                                                 AT 176500.PERFORM THESE TESTS FOR THE CONSOLE SLU
57                                                 IF BIT03 OF SWR IS ADDITIONALLY SET.
58                                              C. WHILE IN MAINTENANCE MODE DO NOT WRITE AND READ
59                                                 A ↑P OR AN ASCII 220. THIS WILL FORCE THE CONSOLE
60                                                 INTO 'CONSOLE MODE'.
61                                              D. IN THE CLOCK REPEATABILITY TEST, ALLOW FOR A
62                                                 TOLERANCE OF 2 BETWEEN CLOCK COUNTS. THIS IS TO
63                                                 ALLOW ENOUGH TOLERANCE WHEN MOS MEMORY IS USED
64                                                 AND REFRESH CYCLES ARE OCCURING.
65
66                                      2. BECAUSE 11/44  SLU INTERRUPT REQUESTS ARE DEPENDANT ON A MFM CLOCK
67                                         ENOUGH TIME MUST BE GIVEN FOR THEM TO OCCUR. THEREFORE, ALL TESTS
68                                         ASSOCIATED WITH XMIT & RECEIVE INTERRUPTS SHOULD HAVE
69                                         A MINIMUM OF 4 NOP'S ACTING AS WAIT FOR INTERRUPT.
70
71                                      3. IT WAS FOUND THAT AN ATTACHED TU58 WOULD BE ACTIVATED
72                                         BY THE DIAGNOSTIC,AND,AS A RESULT,CHARACTERS WOULD BE
73                                         SENT TO THE RECEIVER BUFFER.
74                                         THIS WOULD CAUSE SOME TESTS TO FAIL. THEREFORE TO INHIBIT
75                                         ANY COMMUNICATION TO THE TU58 FROM THE DIAGNOSTIC:
76
77                                              A.ENABLE MAINTENANCE MODE IN ALL TESTS THAT WRITE TO
78                                                THE XMITTER.
79                                              B.IN ALL TESTS THAT WITE TO XMITTER AND BEFORE LEAVING
80                                                TEST; DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
81                                                THAT MIGHT BE IN THE PROCESS OF BEING XMITTED TO
82                                                FINISH.
83                                              C.DISABLE MAINTENANCE MODE FOR PURPOSE OF ERROR PRINTING
84                                                ONLY TO THE SLU WHICH THE TERMINAL IS ATTACHED.
85
```

```
86
87
88
89
90
91
92
93
94
95
96
97
```

4. IT WAS FOUND THAT UPON POWERUP AND WITH THE TU58 ATTACHED, CHARACTERS WOULD BE SENT TO THE RECEIVER FROM THE TU58. SOME RECEIVER TESTS WOULD FAIL DUE TO THIS. THEREFORE, ON ALL TESTS THAT PERFORM RECEIVER TESTS AND FOLLOWING MAINTENANCE MODE BEING ENABLED,DELAY ENOUGH TIME TO ALLOW TO ALLOW ANY CHARACTERS THAT MIGHT BE IN THE PROCESS OF BEING RECEIVED TO FINISH.

5. ADD SOFTWARE THAT WILL IMPLEMENT AUTO INITIATION OF 11/44 T/A CONSOLE TEST VIA WRAP CABLE FROM TU58 TO CONSOLE PORTS. IT IS SELECTED BY SWR BIT02 AND IS PERFORMED ONLY AFTER ALL SLUS ARE TESTED.

CZDLDEO WAS RELEASED JUL 80

1. PROGRAM WAS CHANGED TO WORK WITH VECTORS GREATER THAN 377.

CZDLDFO WAS RELEASED OCT 80

1. IN TESTS 17 THROUGH 22 THE CLOCK DONE FLAG CAN GET SET BETWEEN THE TIME THAT IT IS CLEARED (BIC) AND THE TIME THAT INTERRUPTING IS ENABLED (BIS). THE (BIC-BIS) COMBINATION WAS CHANGED TO A (MOV) INSTRUCTION TO SET INTERRUPT ENABLE AND CLEAR DONE FLAG ALL WITH ONE INSTRUCTION.

2. IN TEST 20, 3 (NOP)'S WERE ADDED BEFORE ERROR ↑47 TO ALLOW SUFFICIENT TIME BETWEEN CLOCK FLAG AND CLOCK INTERRUPT.

3. IN TEST 23, THE COUNT DIFFERENCE ALLOWED WAS CHANGED ON 11/44'S ONLY FROM 2 TO 3.

4. TEST 45 HAD SEVERAL PROBLEMS.
   A. PROGRAM CAN KICK OUT OF DELAY LOOP PREMATURELY. A FIXED DELAY WAS ADDED.
   B. ERROR ↑113 DID NOT PRINT. A RESET WAS ADDED IMMEDIATELY BEFORE THE ERROR CALL TO BREAK THE MAINTENANCE WRAPAROUND.
   C. CLKCNT WAS NOT INITIALIZED.

CZDLDGO WAS RELEASED APR 81

1. THE LINE CLOCK SYNCHRONIZING PROBLEM, WHICH WAS CORRECTED BY PATCH ORIGINALLY IN CZDLDO, WAS REMOVED IN THE NEW REVISION. THEREFORE, TEST 20 NOW CONTAINS THE PATCH TO CORRECT THIS PROBLEM.
2. RANDOM ↑Q'S READ ON ANY INPUT MODE WILL BE IGNORED. THIS WAS A PROBLEM WITH CERTAIN SYSTEM HOOKUPS.
3. THE $SCOPE AND $ERROR ROUTINES NOW CHECK FOR BIT 0 OF THE CPU ERROR REGISTER BEING SET (POWER MONITOR BIT).

CZDLDHO WAS RELEASED JUL 81

1. IN TEST 36, THE "CLR @RBUF" INSTRUCTION WAS CHANGED TO "TST @RBUF" BECAUSE AN 11/24 DOES ONLY A WRITE, AND THE PURPOSE OF THAT INSTRUCTION IN THAT POSITION WAS TO DO A READ.

CZDLDIO WAS RELEASED SEPTEMBER 1984

1. THE TERMINAL OUTPUT TEST DID NOT CHECK FOR XON, XOFF.

1.0 GENERAL INFORMATION

1.1 ABSTRACT

THIS DIAGNOSTIC IS A LOGIC TEST TO VERIFY THE OPERATION OF THE
THE FOLLOWING MODULES:

1. DL11-W SERIAL LINE/REAL TIME CLOCK INTERFACE
2. 11/44 MULTIFUNCTION MODULE

THE PROGRAM WILL RUN WITHOUT ANY SPECIAL TEST FIXTURES BY DEFAULT.
HOWEVER, THE FOLLOWING OPTIONAL TESTING IS PROVIDED:

1. TEST TO VERIFY XMIT AND RECEIVE OF THE UARTS WITH
   A WRAP CABLE. THE UART UNDER TEST IS LOOPED BACK ON
   ITSELF. THIS IS SELECTED VIA OPERATIONAL SWITCH SETTING
   BIT 7, AND IS APPLICABLE TO THE DL11W AND 11/44 MFM.

2. AUTOMATIC INITIATION OF THE T/A CONSOLE TEST OF THE 11/44
   MFM. THE TU58 PORTS ARE LOOPED TO THE CONSOLE PORTS
   THIS IS SELECTED VIA OPERATIONAL SWITCH SETTING BIT 2
   AND IS APPLICABLE TO 11/44 MFM ONLY.
   (SEE CKKFBAO FOR T/A CONSOLE TEST EXPLANATION)

THIS DIAGNOSTIC OPERATES ON THE CONSOLE SERIAL LINE AND CLOCK INTERFACES
AS WELL AS UP TO FIFTEEN(15) ADDITIONAL IDENTICALLY CONFIGURED
SERIAL LINE INTERFACES. THE DEFAULT ADDRESSES ARE:

A. CONSOLE - 177560 SERIAL LINE
             177546 CLOCK

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

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

POWER FAILURE IS SUPPORTED FOR SYSTEMS WITH CORE MEMORY OR BATTERY
BACKUP.

NOTE: THIS DIAGNOSTIC WITH THE SWR = 000020
(CLOCK TESTS ONLY) SHOULD BE USED ON SWITCHLESS CPU'S
TO TEST KW-11L LINE CLOCK MODULES.

1.2 SYSTEM REQUIREMENTS

1.2.1 EQUIPMENT

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

```
                        OPTIONAL:
                                1.LOOP CABLE FOR UART LOOP BACK ON ITSELF
                                2.WRAP CABLE FOR 11/44 MFM T/E TESTING
                                  (SEE DWG. #7016942 WRAP AROUND CABLE
                                   AND SECTION 5.0 OF THIS DOCUMENT)


                1.2.2   STORAGE
                        -------
                        THE PROGRAM USES 5K WORDS OF MEMORY
```

1.3 ASSUMPTIONS

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

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

C.

FOR THE DL11-W:

THE PROGRAM WILL ASSUME THE ERROR FLAG BITS AND THE BREAK FUNCTION OF THE DL11-W ARE DISABLED AND WILL NOT TEST THESE FUNCTIONS UNLESS ENABLED BY BIT10 (FOR ERROR FLAGS) AND BIT8 (FOR BREAK) OF THE SWR. THE DEFAULT CHARACTER SIZE IS 8 BITS (SEE PARA 2.3.2).

FOR THE 11/44:

THE PROGRAM ASSUMES THAT THE ERROR FLAG BITS AND THE BREAK FUNCTION ARE DISABLED, AND WILL NOT TEST THESE FUNCTIONS HOWEVER, IF BIT 10 (FOR ERROR FLAGS) AND BIT 08 (FOR BREAK) OF SWR ARE SET, THEN ERROR FLAGS AND BREAK TESTS ARE PERFORMED FOR THE TU58 SLU ONLY. IF BIT03 OF SWR IS ALSO SET THEN THESE TESTS WILL BE ENABLED FOR THE CONSOLE.

READER ENABLE AND RECEIVER ACTIVE TESTS ARE NOT PERFORMED SINCE THE 11/44 MFM DOES NOT IMPLEMENT THESE FUNCTIONS.

2.0 OPERATING INSTRUCTIONS

2.1 LOADING PROCEDURE

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

2.2 STARTING PROCEDURE

LOAD THE SWITCH REGISTER WITH SETTING

NOTE: IF USING A CPU WITHOUT HARDWARE SWITCH REGISTER
SOFTWARE SWITCH REGISTER LOCATION = 176.
(FOR A 11/44 CPU USE MFM CONSOLE FOR DEPOSITING SWITCH REGISTER. TYPE ↑P TO ENTER CONSOLE)

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 HALTS THE TEST AND PRINTS "STOP" AT THE TERMINAL
                CONTINUING RESTARTS THE ECHO TEST.

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

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

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

                    @ABCDEFGHIJKLMNOPQRSTUVWXYZ[\               [UPPER CASE ALPHA]


                                        ****NOTE****

            IF THE TESTING ON TERMINIALS OTHER THAN THE CONSOLE IS DESIRED                  ;;**
C
            FOR TESTS B OR C,SEE SECTION 2.3.4. AND 2.3.5. OF THIS DOCUMENT.               ;;**
C
```

2.3 OPERATING PROCEDURE

2.3.1 OPERATIONAL SWITCH SETTINGS

THE DIAGNOSTIC WILL CHECK FOR EXISTENCE OF SWITCH REGISTER AT 177570.
IF NO SWITCH REGISTER IS AVAILABLE THE PROGRAM WILL AUTOMATICALLY USE THE CONTENTS OF LOCATION 176 AS THE SOFTWARE SWITCH REGISTER. THE USER SHOULD SET THIS LOCATION BEFORE STARTING THE PROGRAM. IF A HARDWARE SWITCH IS AVAILABLE AND A SOFTWARE SWR(LOC. 176) IS DESIRED,LOAD ALL 1'S INTO LOCATION 177570.(ALL SWITCHES UP IF PHYSICAL SWITCHES ARE AVAILABLE)

BIT15 - HALT ON ERROR
BIT14 - LOOP ON PRESENT TEST
BIT13 - INHIBIT ERROR TYPEOUT
BIT12 - UNUSED
BIT11 - UNUSED
BIT10 - ENABLE ERROR FLAGS TESTS
BIT09 - LOOP ON ERROR
BIT08 - ENABLE BREAK FUNCTION TESTS
BIT07 - ENABLE DATA TEST THROUGH LOOP-BACK CONNECTOR
BIT06 - INHIBIT RTC TESTS (ALLOW ONLY SLU TESTS)
BIT05 - ALLOW MANUAL SETTING OF "$DEVM" (DEVICE MAP)
BIT04 - INHIBIT SLU TESTS (ALLOW ONLY LINE CLOCK TESTS)
BIT03 - FOR 11/44 MFM: ENABLE BOTH 'BREAK TESTS' AND 'ERROR FLAG TESTS' FOR THE CONSOLE SLU. (THIS BIT IS VALID ONLY IF BIT10 OR BIT08 IS SET.)
BIT02 -11/44 MFM OPTION: SELECT AUTO INITIATION OF T/A CONSOLE TEST VIA WRAP CABLE

FOR DL11-W:

IF THE SOFTWARE SWITCH REGISTER IS USED(LOC. 176) THEN BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS. BECAUSE THIS DIAGNOSTIC USES THE MAINTENANCE BIT OF THE SERIAL LINE, THE CONTROL-G SHOULD BE ISSUED DURING PROGRAM TYPEOUTS AT THAT TIME THE MAINTENANCE BIT IS SURE TO BE CLEAR.

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

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

```
356                     1.      <CR>            IF NO CHANGES ARE TO BE MADE
357                     2.      6 DIGITS 0-7    TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER
358                                             VALUE ¡LAST DIGIT FOLLOWED BY <CR>.
359                     3.      ↑U              TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED
360                                             KEYING IN SWREG VALUE.
361
362                     FOR 11/44 CPU:
363
364                     SINCE THE 11/44 HAS A HARDWARE SWITCH REGISTER LOADED BY THE MFM CONSOLE
365                     THEN THE DIAGNOSTIC SHOULD ALWAYS FIND EXISTENCE OF 177570.
366                     WHEN OPERATING ON THE 11/44 CPU, DYNAMIC CHANGING OF SWREG(177570) DURING
367                     PROGRAM EXECUTION CAN BE ACCOMPLISHED BY USING
368                     THE MFM CONSOLE. TYPING ↑P<CR> ON THE CONSOLE TTY WILL ENTER THE
369                     CONSOLE. THIS IS CONSIDERED "CONSOLE MODE".
370                     TO EXAMINE SWREG TYPE ' E SW<CR> ' TO THE CONSOLE PROMPT. TO LOAD THE
371                     SWREG TYPE ' D SW DATA<CR> ' WHERE "DATA" IS AN OCTAL NUMBER.
372                     IN ORDER FOR THE DIAGNOSTIC TO TYPE TO THE TTY IT IS NECESSARY
373                     TO HAVE THE 11/44 MFM IN "PROGRAM I/O MODE". THIS CAN BE ACCOMPLISHED
374                     BY TYPING ' C<CR> ' TO THE CONSOLE PROMPT. THEREFORE,WHEN CONSOLE
375                     USE IS COMPLETED, PLACE THE MFM IN "PROGRAM I/O MODE".
376                     BECAUSE THIS DIAGNOSTIC USES THE MAINTENANCE BIT, ↑P WILL
377                     NOT BE ACKNOWLEDGED DURING THESE TESTS.THEREFORE, ISSUE ↑P
378                     DURING PROGRAM TYPEOUTS. AT THIS TIME THE MAINTENANCE BIT
379                     WILL BE CLEARED.
```

2.3.2 SETTING BITS PER CHARACTER

THIS PROGRAM DEFAULTS TO TESTING 8 BITS PER CHARACTER. IF THE SERIAL LINE IS SET FOR 5-->7 BITS PER CHARACTER, SET THE MEMORY LOCATION "$USWR" AS FOLLOWS:

CHAR. SIZE (# OF BITS)	"$USWR" CONTENTS (BINARY)	(OCTAL)
8	100000000	400
7	010000000	200
6	001000000	100
5	000100000	40

"$USWR" IS USED IN THE DATA PATH TESTS TO LIMIT THE BINARY COUNT TEST PATTERN TO THE NUMBER OF BITS SELECTED ON THE SERIAL LINE.

2.3.3 RUNNING UNDER APT

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

FOR DL11W:

THE DEFAULT EXECUTION TIMES PROVIDED($TSTM,$PASTM) ARE FOR EXECUTION WITH AN 11/34 PROCESSOR, CORE MEMORY , AND 110 BAUD.

FOR 11/44:

THE EXECUTION TIMES PROVIDED IN THE APT SCRIPT THAT FOLLOWS AND IN SECT. 2.4 ARE FOR EXECUTION WITH A 11/44 PROCESSOR, CACHE, 16K CORE MEMORY, AND 300 BAUD.

THE FOLLOWING IS A PROGRAM LOAD FILE USED BY APT:

1. E TABLE 'A' IS USED FOR APT DUMP MODE AND RUN TIME MODES.
   A. TWO SLU'S ARE TESTED.($SWREG BIT05=1 AND $DEVM=3)
      A SLU AT 177560(DEFAULT CONSOLE SLU) AND AT 176500 (BASE ADDRESS CODE).
   B. THE ERROR FLAG AND BREAK TESTS ARE SELECTED FOR THE SLU AT 176500(TU58 SLU IN MFG.)
      ($SWREG BIT 10 AND 8 =1)
   C. THE 'ENABLE DATA TEST THROUGH LOOP-BACK CONNECTOR' TEST IS ENABLED TO ALLOW TEST 44 TO EXECUTE ($SWREG BIT 7 =1).

2. E TABLE 'B' IS USED FOR APT QV. IT ACCOMPLISHES WHAT E TABLE 'A' DOES, WITHOUT THE ENABLE DATA TEST THROUGH THE LOOPBACK CONNECTOR, BUT ADDITIONALLY IT SUPPRESSES ALL TYPEOUTS TO THE TERMINAL ($ENVM=240) AND SELECTS AUTO TESTING OF T/A CONSOLE TEST VIA WRAP CABLE ($SWREG BIT02=1).

1ST PASS RUN TIME	LONGEST TEST TIME	ADDITIONAL RUN TIME
60	50	45

```
437
438                                ......          E TABLES        .......
439
440
441                                                A               B
442                             E-MODE/S-MODE      200/000         240/001
443                             ($ENVM/$ENV)
444
445                             SWITCH REGISTER 1  002640          002404
446                             ($SWREG)
447
448                             SWITCH REGISTER 2  000400          000400
449                             CPU TYPE/OPTIONS   00/0000          00/0000
450                             MEMORY MAP CODE 1  000/00000000    000/00000000
451                             MEMORY MAP CODE 2  000/00000000    000/00000000
452                             MEMORY MAP CODE 3  000/00000000    000/00000000
453                             MEMORY MAP CODE 4  000/00000000    000/00000000
454                             BUS PRIORITY/INTERRUPT 1  0000     0000
455                             BUS PRIORITY/INTERRUPT 2  0000     0000
456                             BUS ADDRESS CODE   176500          176500
457                             DEVICE MAP CODE    000003          000003
458                             ($DEVM)
459
460                             CTLR. SPECIFIC WORD 1  000000      000000
461                             CTLR. SPECIFIC WORD 2  000000      000000
```

2.3.4 RUN WITH ALTERNATE CONSOLE ADDRESS

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

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

2.3.5 TESTING ADDITIONAL SERIAL LINES

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

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

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

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

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

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

```
513                 STARTING AT LOCATION 200 AND HAVING BITS OF SWR CLEAR, THE PROGRAM WILL SELF
514                 SIZE THE NUMBER OF DEVICES (STARTING AT THE BAS ADDRESS) AND
515                 STORE A BIT MAP AT "$DEVM" (DEVICE MAP) TO INDICATE WHICH UNIT NUMBERS
516                 ARE PRESENT AND WILL BE TESTED:
517
518
519                         ------------------------------------------------
520                         ! UNIT ! UNIT ! .....! UNIT ! UNIT ! CONSOLE!
521                         !  15  !  14  ! .....!   2  !   1  !        !
522                         ------------------------------------------------
523
524                 A BIT MAP CAN BE ENTERED AT "$DEVM" PRIOR TO STARTING THE PROGRAM
525                 SETTING BITS OF THE SWR INHIBITS THE SELF-SIZING AND DEVICE MAP
526                 GENERATION,AND USES THE VALUE STORED  BY THE OPERATOR.
527
528                         EXAMPLE:
529
530                         SWR  =  000040          [BINARY  0 000 000 000 100 000]
531                         $BASE:  776500
532                         $VECT1: 300
533
534                         $DEVM:  13      [BINARY - 0 000 000 000 001 011]
535
536                         THE PROGRAM WILL TEST -
537                                 UNIT# 0 = CONSOLE
538                                 UNIT# 1 = 776500 ; 300
539                                 UNIT# 3 = 776520 ; 320
```

```
                        2.4     EXECUTION TIMES
                                --- ----------

                                FOR DL11-W: (110 BAUD)
                                        LONGEST SUBTEST TIME - 50 SECONDS
                                        PASS TIME           - 60 SECONDS
                                        ADDITIONAL DEVICES  - 55 SECONDS/DEVICE

                                FOR 11/44: (300 BAUD)
                                        LONGEST SUBTEST TIME - 50 SECONDS
                                        PASS TIME           - 60 SECONDS
                                        ADDITIONAL DEVICES  - 55 SECONDS/DEVICE
```

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

                                        "(SOME ASCII MESSAGE)"
                                        TEST#   ERR PC   RCSR    [ANY APPLICABLE 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 AND THE PC CAN BE EXAMINED BY THE OPERATOR
                                TO FIND THE ERROR INFORMATION IN THE PROGRAM LISTING.

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

                                IF THE CPU THIS DIAGNOSTIC WILL BE RUN ON DOES NOT HAVE A CPU ERROR
                                REGISTER (ADDRESS 17777766), THE FOLLOWING SECTION DOES NOT APPLY,
                                SINCE THE ROUTINES MENTIONED PLANS FOR THAT POSSIBLITY.

                                IF AN ERROR SHOULD OCCUR WHEN CHECKING THE POWER MONITOR BIT IN THE
                                SCOPE ROUTINE, THE ERROR WILL BE CALLED FROM THAT ROUTINE.  IF THE
                                BIT BECOMES SET AFTER THE SCOPE ROUTINE, AND AN ERROR OCCURS FOR ANY
                                REASON, *TWO* ERRORS WILL CALL.  THE POWER MONITOR BIT ERROR WILL CALL
                                FIRST, THEN THE ERROR TO BE CALLED WILL CALL.  IT IS A DEFINITE POSSI-
                                BILITY THAT THE ERROR WAS CAUSED BY THE POWER SUPPLY(S) THAT WERE OUT
                                OF SPECIFICATION, AND REPAIR SHOULD BE EXECUTED BEFORE RELYING ON THE
                                RESULTS OF THE FAILURE.
```

```
590
591
592                     4.0     SUBROUTINE ABSTRACTS
593                             ---------- - --
594
595                     4.1     TRAPCATCHER
596                             -----------
597                             THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0 DESIGNED TO
598                             DETECT AND ISOLATE UNEXPECTED TRAPS TO THE TRAP AND INTERRUPT VECTOR
599                             AREA OF MEMORY.
600
601                             EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS OF THE NEXT
602                             LOCATION. THE NEXT LOCATION IS LOADED WITH A BREAK POINT TRAP (000003).
603                             THUS AN ILLEGAL TRAP OR INTERRUPT CAUSES A TRAP THROUGH THE BPT VECTOR(14)
604                             WHICH POINTS TO THE "CATCH" ROUTINE.
605
606                             THE "CATCH" ROUTINE REPORTS THE PC THAT CAUSED THE ORIGINAL TRAP AND
607                             THE LOCATION OF THE TRAP VECTOR (IF UNDER APT, AN ERROR IS INDICATED TO APT).
608                             AFTER REPORTING THE ERROR THE PROGRAM HALTS. THE PROGRAM MUST BE RESTARTED
609                             AT THIS POINT.
610
611
612                     4.2     WRPSW
613                             -----
614                             THIS ROUTINE IS USED TO WRITE THE PSW BY POPPING VALUES FROM THE
615                             STACK. THIS METHOD IS USED TO BE COMPATIBLE WITH ALL 11 FAMILY
616                             PROCESSORS.
617
618
619                     4.3     SCOPE
620                             -----
621                             THIS ROUTINE CALL IS PLACED BETWEEN EACH SUBTEST. IT RECORDS THE STARTING
622                             ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED AND UPDATES THE
623                             TEST NUMBER. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE
624                             SUBTEST AT WHICH THE SCOPE LOOP IS REQUESTED.
625
626
```

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


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


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


                5.0     AUTOMATIC INITIATION OF 11/44 T/A CONSOLE TEST
                        VIA WRAP CABLE
                        ----------------------------------------------


                                PURPOSE: THE T/E(OR T/A) CONSOLE TEST CAN BE IMPLEMENTED
                                         BY MANUALLY TYPING T/E(OR T/A) ON THE KEYBOARD
                                         OF THE TERMINAL WHEN IN CONSOLE MODE. IN ORDER TO
                                         IMPLEMENT THIS TEST IN AN AUTOMATIC WAY IN MANUFACTURING
                                         WHILE UNDER APT, THE USE OF A WRAP CABLE FROM
                                         THE TU58 TO THE CONSOLE CAN BE USED ALLOWING THIS
                                         DIAGNOSTIC TO ISSUE THE 'T/A' COMMAND TO THE CONSOLE.
                                         THE DIAGNOSTIC WILL ISSUE THE APPROPRIATE SEQUENCE
                                         OF COMMANDS TO THE CONSOLE VIA THE WRAP CABLE WHEN
                                         BIT02 OF SWR =1. THE DIAGNOSTIC WILL THEN MONITOR
                                         THE EXPECTED RESPONSE FROM THE CONSOLE VIA THE WRAP
                                         CABLE AND HALT IF THERE IS AN ERROR. THE T/A TEST
                                         IS DONE ONLY AFTER ALL SLUS ARE TESTED. IF T/A IS SUCCESSFUL
                                         'END PASS' IS PRINTED AND THE DIAGNOSTIC STARTS AGAIN.

                                        FIGURE 1 SHOWS THE PROPER WRAP CABLE SETUP AND
                                        REQUIREMENTS FOR AUTOMATICALLY INITIATING T/A FROM THE
                                        THE DIAGNOSTIC.NOTICE THAT THIS ARRANGEMENT ALLOWS
                                        FOR THE TERMINAL TO MONITOR ALL COMMUNICATION FROM
                                        THE CONSOLE OUTPUT DURING EXECUTION OF THE DIAGNOSTIC
                                        IN "WRAP MODE".

                        TYPEOUT EXAMPLES
                        - - - - - - - - - -

                        1. WITH THE CONFIGURATION OF FIGURE 1 AND "WRAP MODE" SELECTED
```

THE PROGRAM MAY BE LOADED BY APT. IF 'F TABLE B' OF SECTION
2.3.3 WERE USED WITH THE EXCEPTION OF ALLOWING TYPEOUTS(#ENVM=200)
THE FOLLOWING WOULD BE SEEN ON THE LOCAL TERMINAL:

```
CZDLDIO DL11-W/1144 MFM SLU
02 DEVICES UNDER TEST  ↑P
CONSOLE
>>>T/A

CONSOLE-TESTB
END PASS ↑P
CONSOLE
>>>T/A

CONSOLE-TESTB
END PASS ↑P
CONSOLE
>>>T/A

CONSOLE-TESTB
END PASS ↑P   .....ETC.
```

DESCRIPTION: REFERING TO THE ABOVE PRINTOUTS:

A. THE DIAGNOSTIC IS LOADED WITH THE TITLE BEING PRINTED.
B. TWO SLU DEVICES(CONSOLE,TU58) ARE SPECIFIED
C. BOTH SLUS ARE TESTED COMPLETELY. AT THIS POINT
   THE THE DIAGNOSTIC ISSUES ↑P TO THE WRAP CABLE.
   THE CONSOLE ENTERS CONSOLE MODE AND
   THE ↑P TYPED ON THE TERMINAL IS THE MFM CONSOLE ECHO.
D. THE CONSOLE PERFORMS ITS OWN SELF TEST BY TYPING
   'CONSOLE' FOLLOWED BY >>>, THE CONSOLE PROMPT.
E. THE DIAGNOSTIC THEN ISSUES T/A AND THE TERMINAL
   SHOWS THE CONSOLE ECHO OF THIS.
F. THE MFM THEN PERFORMS THE T/A TEST SHOWN BY THE TERMINAL
   TYPING 'CONSOLE-TESTB'.THE DIAGNOSTIC LOOKS FOR THE
   'B' IN THIS TYPEOUT,AND WHEN FOUND , CONSIDERS
   THE TEST A SUCCESS. THE DIAGNOSTIC WILL TYPE END PASS
   INDICATING A SUCCESSFUL PASS OF THE DIAGNOSTIC.
G. THE DIAGNOSTIC CONTINUES WITH FURTHER PASSES OF THE
   DIAGNOSTIC.

2. IF 'E TABLE B' OF 2.3.3 WERE USED WITH TYPEOUTS SUPPRESSED (@ENVM=240) AS STATED, THEN THE FOLLOWING TYPEOUTS WOULD BE NOTICED:

```
↑P
CONSOLE
>>>T/A

CONSOLE-TESTB↑P
CONSOLE
>>>T/A

CONSOLE-TESTB↑P
CONSOLE
>>>T/A

CONSOLE-TESTB↑P .....ETC.
```

```

                                       FIG. 1- WRAPAROUND CONFIGURATION - AUTO INITIATION OF
                                               11/44 T/A CONSOLE TEST
                                               ------------------------   --   ----  -- ---


                        ;**************************************************************************
                        ;                          WRAP AROUND CABLE
                        ;
                        ;     TU58                                       CONSOLE TTY PORT
                        ;
                        ; INPUT       OUTPUT                         INPUT          OUTPUT
                        ;   !           !                              !               !
                        ;   !           !                   C1         !               !
                        ;   !           +-------------------+ /--------+               !
                        ;   !                              S1/ +C2                     !
                        ;   !                                  !                       !
                        ;   !                                  !                       !
                        ;   !                  C3               \                      !
                        ;   +--------------------+---+----------------------------------+
                        ;                       \ S3  !            /
                        ;                        \    !           !
                        ;                             !           !
                        ;                             !           !
                        ;                   TTY PRINTER   TTY KEYBOARD
                        ;
                        ; WHEN THE SWITCH IS IN "NORMAL" MODE CONTACTS ARE:
                        ;                               S1-C2 IS CLOSED
                        ;                               S1-C1 IS OPEN
                        ;                               S3-C3 IS OPEN
                        ; THIS ALLOWS THE USE OF THE TERMINAL WITH THE CONSOLE
                        ;
                        ; WHEN THE SWTICH IS IN "WRAP" MODE CONTACTS ARE
                        ;                               S1-C1 CLOSED
                        ;                               S1-C2 OPEN
                        ;                               S2-C3 CLOSED
                        ; THIS ALLOWS THE TU58 PORT TO COMMUNICATE WITH THE CONSOLE PORT AND FOR
                        ; THE TTY TO MONITOR THE CONSOLE OUTPUT
                        ;
                        ; THIS IS ALL DONE WITH A DPDT SWITCH
                        ;
                        ; BEFORE THIS TEST CAN BE RUN THE CONSOLE UART,TTY,TU58 MUST BE AT
                        ; THE SAME BAUD RATE. THE CONSOLE AND TU58 UART,AND THE TTY SHOULD BE
                        ;SET UP WITH THE SAME STOP BIT SETTING,
                        ; WITH PARITY INHIBITED AND WITH A WORD LENGTH OF 7 OR 8 BITS
                        ;
                        ; FOR MORE INFORMATION SEE THE MFM PRINTS AND THE WRAP AROUND
                        ; CABLE DRAWING NUMBER: 7016942
                        ;
                        ;**************************************************************************

```

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


                        ;********************** .$ERROR ***********************
                        ;
                        ;.$ERROR IS THE ERROR HANDLER
                        ;
                        ;ARGUMENTS:
                        ;
                        ;1) ADRESS --    IF NON-BLANK=ADDRESS OF USER ERROR ROUTINE
                        ;                IF BLANK AND SW13 IS USED TO INHIBIT ERROR
                        ;                TYPEOUTS, THE PC OF THE "ERROR" WILL BE TYPED.
                        ;                NOTE:   IF SW13 IS USED TO INHIBIT ERROR
                        ;                        TYPEOUTS A "CR" AND "LF" WILL ALWAYS
                        ;                        BE THE LAST THING TYPED AND IS PROVIDED
                        ;                        BY THIS ROUTINE
                        ;
                        ;2) INSTR --     IF NON-BLANK WILL BE THE FIRST INSTRUCTION
                        ;                OF THE ROUTINE
                        ;                (EXAMPLE OF USE    <<MOV R1,SAVR1>>
                        ;                NOTE:   INSTR CAN BE A MACRO I.E. <<SAVE <R1,R2,R3,R4>>>)
                        ;
                        ;3) INSTR2 --    IF NON-BLANK WILL REPLACE THE LAST INSTRUCTION (RTI).
                        ;                REFER TO ARGUMENT 2 (INSTR) FOR AN EXAMPLE OF USE.
                        ;                BE SURE TO PROVIDE AN RTI FOR EXITING THE ROUTINE.
                        ;
                        ;NOTE:  THIS ROUTINE IS CONDITIONALLY ASSEMBLED BY $SWR
                        ;               FOR SW09,SW10,SW13,&SW15
                        ;SW09=1 LOOP ON ERROR
                        ;SW10=1 BELL ON ERROR
                        ;SW13=1 INHIBIT ERROR TYPEOUTS
                        ;SW15=1 HALT ON ERROR
                        ;
                        ;ROUTINES REQUIRED:
                        ;
                        ;1)     TYPE AN ASCIZ STRING (.$TYPE) DEPENDING ON $SWR
                        ;
                        ;
                        ;******************************************************
                        ;
                        ;**********************************************************************
                        ;**********************************************************************

                        ;********************** .$ERRTYP **********************
                        ;
                        ;.$ERRTYP IS USED TO REPORT ERRORS
                        ;ITS INTENDED USE IS TO BE IN CONJUCTION WITH .$ERROR
                        ;
                        ;ARGUMENTS:
                        ;
                        ;1)     A  --    IF BLANK ALL DATA IN THE "DATA TABLE"(DT) ARE ASSUMED TO
                        ;                BE OCTAL NUMBERS AND ARE TYPED AS SUCH.
                        ;
                        ;                IF NON-BLANK THE "DT" CAN CONTAIN BOTH OCTAL AND DECIMAL
                        ;                NUMBERS WITH THE "DATA FORMAT"(DF) INDICATING HOW EACH
                        ;                NUMBER IS TO BE TYPED. THE "DF" MUST BE A BYTE DATA STRING
```

```
1272                            ;               WITH "0" FOR OCTAL AND "1" FOR DECIMAL.
1273                            ;
1274                            ;ROUTINES REQUIRED:
1275                            ;
1276                            ;1)     .$TYPOCT        TYPE AN OCTAL NUMBER
1277                            ;
1278                            ;2)     .$TYPDEC        TYPE A DECIMAL NUMBER (ONLY NEEDED IF "A" IS NON-BLANK).
1279                            ;
1280                            ;NOTES:
1281                            ;
1282                            ;1)     THIS ROUTINE PROVIDES AN AUTOMATIC "CARRIAGE RETURN-LINE FEED"
1283                            ;       FOR "EM", "DH" AND "DT".
1284                            ;
1285                            ;2)     TWO(2) SPACES ARE TYPED AFTER EACH NUMBER FOR "DT".
1286                            ;
1287                            ;EXAMPLE OF USE WITH .$ERROR
1288                            ;
1289                            ;       .$ERROR  ERRTYP         ;;$ERRTYP IS ENTRY POINT OF .$ERRTYP
1290                            ;       .$ERRTYP        X       ;;"DT" CONTAINS OCTAL AND DECIMAL NUMBERS
1291                            ;
1292                            ;*****************************************************
1293
1382
1386
1390
1394
1395
1402
1403
1409
1419
1431
1432
1433                            .MCALL  NEWTST,$$NEWTEST,.$TYPE,.$TYPOC,.$TRAP
1434                            .MCALL  .SETUP,STARS,PUSH,POP,SETUP,.EQUIV
1435                            .MCALL  .$APTHDR,.$APTBLS,.$ACT11
1436                            .MCALL  .$CMTAG,.$EOP,.$READ
1437                            .MCALL  .EQUAT
1438                            .SBTTL  BASIC DEFINITIONS
                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
104000                                  ERROR=EMT
000004                                  SCOPE=IOT
                                ;*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
177776                                  PSW=PS
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
```

BASIC DEFINITIONS

```
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER
                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7
                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
001000                          SW9=SW09
000400                          SW8=SW08
000200                          SW7=SW07
000100                          SW6=SW06
000040                          SW5=SW05
000020                          SW4=SW04
000010                          SW3=SW03
000004                          SW2=SW02
000002                          SW1=SW01
000001                          SW0=SW00
                        ;*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
                001000                          BIT9=BIT09
                000400                          BIT8=BIT08
                000200                          BIT7=BIT07
                000100                          BIT6=BIT06
                000040                          BIT5=BIT05
                000020                          BIT4=BIT04
                000010                          BIT3=BIT03
                000004                          BIT2=BIT02
                000002                          BIT1=BIT01
                000001                          BIT0=BIT00
                                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
                000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
                000014                  TBITVEC=14              ;;"T" BIT
                000014                  TRTVEC= 14              ;;TRACE TRAP
                000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
                000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                000024                  PWRVEC= 24              ;;POWER FAIL
                000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
                000034                  TRAPVEC=34              ;;"TRAP" TRAP
                000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
                000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
                000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
1439            000007                          MFPT=7
1440            176500                          ABASE=  176500
1441            000300                          AVECT1= 300
1442            000400                          AUSWR=  400
1443            000001                          $TN=    1
1444            161000                          $SWR=   161000
1445            000003                          BPT=    000003          ;THIS IS THE COMMAND FOR A TRAP
1446                                                                    ;   THROUGH 14 (BPT TRAP)
1447
1448            000000                          .=0
1449                                    ;;**********************************************************************
1450                                            ;*ALL UNUSED LOCATIONS FROM 4-776 CONTAIN A ".+2,BPT"
1451                                            ;*SEQUENCE TO CATCH ILLEGAL TRAPS & INTERRUPPTS
1452                                            ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1453
1461
1462            000014                          .=14                    ;THE BPT TRAP VECTOR POINTS TO THE
1463 000014    014520                          .WORD   CATCH           ;   ILLEGAL TRAP HANDLER "CATCH"
1464 000016    000340                          .WORD   340
1465
1466            000042                          .=      42
1467 000042    000000                          .WORD   0
1468
1469
1471
1472
1473            000174                          .=      174
1474 000174    000000                  DISPREG:.WORD   0
1475 000176    000000                  SWREG:  .WORD   0
1476
```

```
1477            000200                          .=200
1478 000200     000137  003046                  JMP     START           ;DO INTERFACE TEST
1479 000204     000137  017600                  JMP     ECHO            ;DO ECHO TEST
1480 000210     000137  020020                  JMP     OUTTST          ;DO OUTPUT TEST TO TERMINAL
```

BASIC DEFINITIONS

```
                        .=      500
                .SBTTL  ACT11 HOOKS
                ;;**************************************************************
                ;HOOKS REQUIRED BY ACT11
000500                  $SVPC=.                 ;SAVE PC
000046                  .=46
000046  014376          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
000052                  .=52
000052  000000          .WORD   0               ;;2)SET LOC.52 TO ZERO
000500                  .=$SVPC                 ;; RESTORE PC
```

```
1485                                    .SBTTL  APT PARAMETER BLOCK
                                        ;;*****************************************************************
                                        ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                        ;;*****************************************************************
                 000500                         .$X=.   ;;SAVE CURRENT LOCATION
                 000024                         .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
          000024 000200                         200     ;;FOR APT START UP
                 000044                         .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
          000044 000500                         $APTHDR ;;POINT TO APT HEADER BLOCK
                 000500                         .=.$X   ;;RESET LOCATION COUNTER
                                        ;;*****************************************************************
                                        ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                        ;INTERFACE SPEC.
                 000500                 $APTHD:
          000500 000000                 $MXBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
          000502 001066                 $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
          000504 000050                 $TSTM:  .WORD   50      ;;RUN TIM OF LONGEST TEST
          000506 000060                 $PASTM: .WORD   60      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
          000510 000055                 $UNITM: .WORD   55      ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
          000512 000030                         .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
1487                                    .SBTTL  COMMON TAGS
                                        ;;*****************************************************************
                                        ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                        ;*USED IN THE PROGRAM.
        001000                                  .=1000
001000                                  $CMTAG:                         ;;START OF COMMON TAGS
001000  000000                                  .WORD   0
001002     000                          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001003     000                          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001004  000000                          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001006  000000                          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001010  000000                          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001012  000000                          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001014     000                          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001015     001                          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001016  000000                          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001020  000000                          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001022  000000                          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001024  000000                          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001026  000000                          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001030  000000                                  .WORD   0               ;;RESERVED--NOT TO BE USED
001032  000000                                  .WORD   0
001034     000                          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001035     000                          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001036  000000                                  .WORD   0
001040  177570                          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001042  177570                          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001044  177560                          $TKS:   177560                  ;;TTY KBD STATUS
001046  177562                          $TKB:   177562                  ;;TTY KBD BUFFER
001050  177564                          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001052  177566                          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001054     000                          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001055     002                          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001056     012                          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001057     000                          $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001060  000000                          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001062     077                          $QUES:  .ASCII  /?/             ;;QUESTION MARK
001063     015                          $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001064     012     000                  $LF:    .ASCIZ  <12>            ;;LINE FEED
                                        ;;*****************************************************************
                                        .SBTTL  APT MAILBOX-ETABLE
                                        ;;*****************************************************************
                                        .EVEN
001066                                  $MAIL:                          ;;APT MAILBOX
001066  000000                          $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001070  000000                          $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001072  000000                          $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001074  000000                          $PASS:  .WORD   APASS           ;;PASS COUNT
001076  000000                          $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001100  000000                          $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001102  000000                          $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
001104  000000                          $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
001106                                  $ETABLE:                        ;;APT ENVIRONMENT TABLE
001106     000                          $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
001107     000                          $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
001110  000000                          $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
001112  000400                          $USWR:  .WORD   AUSWR           ;;USER SWITCHES
```

```
001114  000000                  $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
                                ;*                      BITS 15-11=CPU TYPE
                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                              11/70=06,PDQ=07,Q=10
                                ;*                      BIT 10=REAL TIME CLOCK
                                ;*                      BIT  9=FLOATING POINT PROCESSOR
                                ;*                      BIT  8=MEMORY MANAGEMENT
001116     000                  $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
001117     000                  $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                              900 NSEC CORE=001
                                ;*                              300 NSEC BIPOLAR=002
                                ;*                              500 NSEC MOS=003
001120  000000                  $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                                ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001122     000                  $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
001123     000                  $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
001124  000000                  $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
001126     000                  $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
001127     000                  $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
001130  000000                  $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
001132     000                  $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
001133     000                  $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
001134  000000                  $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
001136  000300                  $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001140  000000                  $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001142  176500                  $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001144  000000                  $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
001146                          $ETEND:
                                .MEXIT
```

```
                                .SBTTL  ERROR POINTER TABLE
                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT
     001146                     $ERRTB:
1488 001146                     .$ERRTB:
1489 001146  020134                     EM1                     ;"CAN NOT ACCESS TCSR"
1490 001150  024315                     DH1                     ;"TEST# ERR PC  TCSR"
1491 001152  025076                     DT1                     ;$TESTN,$ERRPC,TCSR
1492 001154  000000                     0
1493
1494 001156  020160                     EM2                     ;"CAN NOT ACCESS TBUF"
1495 001160  024342                     DH2                     ;"TEST# ERR PC  TBUF"
1496 001162  025106                     DT2                     ;$TESTN,$ERRPC,TBUF
1497 001164  000000                     0
1498
1499 001166  020204                     EM3                     ;"TCSR DONE NOT CLEARED WITH TBUF FULL"
1500 001170  024315                     DH1                     ;"TEST# ERR PC  TCSR"
1501 001172  025076                     DT1                     ;$TESTN,$ERRPC,TCSR
1502 001174  000000                     0
1503
1504 001176  020251                     EM4                     ;"TCSR DONE NOT SET"
1505 001200  024315                     DH1                     ;"TEST# ERR PC  TCSR"
1506 001202  025076                     DT1                     ;$TESTN,$ERRPC,TCSR
1507 001204  000000                     0
1508
1509 001206  020273                     EM5                     ;TCSR DONE NOT SET WITH RESET
1510 001210  024315                     DH1                     ;"TEST# ERR PC  TCSR"
1511 001212  025076                     DT1                     ;$TESTN,$ERRPC,TCSR
1512 001214  000000                     0
1513
1514 001216  020330                     EM6                     ;"CAN NOT ACCESS RCSR"
1515 001220  024367                     DH6                     ;"TEST# ERR PC  RCSR"
1516 001222  025116                     DT6                     ;$TESTN,$ERRPC,RCSR
1517 001224  000000                     0
```

ERROR POINTER TABLE

```
1518 001226  020354                EM7     ;"CAN NOT ACCESS RBUF"
1519 001230  024414                DM7     ;"TEST# ERR PC  RBUF"
1520 001232  025126                DT7     ;$TESTN,$ERRPC,RBUF
1521 001234  000000                0
1522
1523 001236  020400                EM10    ;"CAN NOT ACCESS LKS"
1524 001240  024441                DM10    ;"TEST# ERR PC  LKS"
1525 001242  025136                DT10    ;$TESTN,$ERRPC,LKS
1526 001244  000000                0
1527
1528 001246  020423                EM11    ;"BIT0 OF TCSR NOT CLEAR AFTER RESET"
1529 001250  024315                DM1     ;"TEST# ERR PC  TCSR
1530 001252  025076                DT1     ;$TESTN,$ERRPC,TCSR
1531 001254  000000                0
1532
1533 001256  020466                EM12    ;"CAN NOT SET BIT0 OF TCSR"
1534 001260  024315                DM1     ;"TEST# ERR PC  TCSR"
1535 001262  025076                DT1     ;$TESTN,$ERRPC,TCSR
1536 001264  000000                0
1537
1538 001266  020517                EM13    ;"CAN NOT CLEAR BIT0 OF TCSR"
1539 001270  024315                DM1     ;"TEST# ERR PC  TCSR"
1540 001272  025076                DT1     ;$TESTN,$ERRPC,TCSR
1541 001274  000000                0
1542
1543 001276  020552                EM14    ;"RESET DID NOT CLEAR BIT0 OF TCSR"
1544 001300  024315                DM1     ;"TEST# ERR PC  TCSR"
1545 001302  025076                DT1     ;$TESTN,$ERRPC,TCSR
1546 001304  000000                0
1547
1548 001306  020613                EM15    ;"BIT2 OF TCSR NOT CLEAR AFTER RESET"
1549 001310  024315                DM1     ;"TEST# ERR PC  TCSR"
1550 001312  025076                DT1     ;$TESTN,$ERRPC,TCSR
1551 001314  000000                0
1552
1553 001316  020656                EM16    ;"CAN NOT SET BIT2 OF TCSR"
1554 001320  024315                DM1     ;"TEST# ERR PC  TCSR"
1555 001322  025076                DT1     ;$TESTN,$ERRPC,TCSR
1556 001324  000000                0
1557
1558 001326  020707                EM17    ;"CAN NOT CLEAR BIT2 OF TCSR"
1559 001330  024315                DM1     ;"TEST# ERR PC  TCSR"
1560 001332  025076                DT1     ;$TESTN,$ERRPC,TCSR
1561 001334  000000                0
```

ERROR POINTER TABLE

```
1562 001336  020742                EM20    ;"RESET DID NOT CLEAR BIT2 OF TCSR"
1563 001340  024315                DH1     ;"TEST# ERR PC  TCSR"
1564 001342  025076                DT1     ;$TESTN,$ERRPC,TCSR
1565 001344  000000                0
1566
1567 001346  021003                EM21    ;"BIT6 OF TCSR NOT CLEAR AFTER RESET2
1568 001350  024315                DH1     ;"TEST# ERR PC  TCSR"
1569 001352  025076                DT1     ;$TESTN,$ERRPC,TCSR
1570 001354  000000                0
1571
1572 001356  021046                EM22    ;"XMIT INTERRUPT WITH PRIORITY 7"
1573 001360  024315                DH1     ;"TEST# ERR PC  TCSR"
1574 001362  025076                DT1     ;$TESTN,$ERRPC,TCSR
1575 001364  000000                0
1576
1577 001366  021103                EM23    ;"CAN NOT SET BIT6 OF TCSR"
1578 001370  024315                DH1     ;"TEST# ERR PC  TCSR"
1579 001372  025076                DT1     ;$TESTN,$ERRPC,TCSR
1580 001374  000000                0
1581
1582 001376  021[illegible]         EM24    ;"CAN NOT CLEAR BIT6 OF TCSR"
1583 001400  024315                DH1     ;"TEST# ERR PC  TCSR"
1584 001402  025076                DT1     ;$TESTN,$ERRPC,TCSR
1585 001404  000000                0
1586
1587 001406  021167                EM25    ;"RESET DID NOT CLEAR BIT6 OF TCSR"
1588 001410  024315                DH1     ;"TEST# ERR PC  TCSR"
1589 001412  025076                DT1     ;$TESTN,$ERRPC,TCSR
1590 001414  000000                0
1591
1592 001416  021230                EM26    ;"BIT6 OF RCSR NOT CLEAR AFTER RESET"
1593 001420  024367                DH6     ;"TEST# ERR PC  RCSR"
1594 001422  025116                DT6     ;$TESTN,$ERRPC,RCSR
1595 001424  000000                0
1596
1597 001426  021273                EM27    ;"RCVR INTERRUPT WITH PRIORITY 7"
1598 001430  024367                DH6     ;"TEST# ERR PC  RCSR"
1599 001432  025116                DT6     ;$TESTN,$ERRPC,RCSR
1600 001434  000000                0
1601
1602 001436  021332                EM30    ;"CAN NOT SET BIT6 OF RCSR"
1603 001440  024367                DH6     ;"TEST# ERR PC  RCSR"
1604 001442  025116                DT6     ;$TESTN,$ERRPC,RCSR
1605 001444  000000                0
```

ERROR POINTER TABLE

```
1606 001446  021363                EM31            ;"CAN NOT CLEAR BIT6 OF RCSR"
1607 001450  024367                DM6             ;"TEST# ERR PC  RCSR"
1608 001452  025116                DT6             ;#TESTN,#ERRPC,RCSR
1609 001454  000000                0
1610
1611 001456  021416                EM32            ;"CAN NOT CLEAR BIT6 OF RCSR WITH RESET2
1612 001460  024367                DM6             ;"TEST# ERR PC  RCSR"
1613 001462  025116                DT6             ;#TESTN,#ERRPC,RCSR
1614 001464  000000                0
1615
1616 001466  021464                EM33            ;"BIT6 OF LKS NOT CLEAR AFTER RESET"
1617 001470  024441                DM10            ;"TEST# ERR PC  LKS"
1618 001472  025136                DT10            ;#TESTN,#ERRPC,LKS
1619 001474  000000                0
1620
1621 001476  021526                EM34            ;"LKS INTERRUPT WITH PRIORITY 7"
1622 001500  024441                DM10            ;"TEST# ERR PC  LKS"
1623 001502  025136                DT10            ;#TESTN,#ERRPC,LKS
1624 001504  000000                0
1625
1626 001506  021564                EM35            ;"CAN NOT SET BIT6 OF LKS"
1627 001510  024441                DM10            ;"TEST# ERR PC  LKS"
1628 001512  025136                DT10            ;#TESTN,#ERRPC,LKS
1629 001514  000000                0
1630
1631 001516  021614                EM36            ;"CAN NOT CLEAR BIT6 OF LKS"
1632 001520  024441                DM10            ;"TEST# ERR PC  LKS"
1633 001522  025136                DT10            ;#TESTN,#ERRPC,LKS
1634 001524  000000                0
1635
1636 001526  021646                EM37            ;"RESET DID NOT CLEAR BIT6 OF LKS"
1637 001530  024441                DM10            ;"TEST# ERR PC  LKS"
1638 001532  025136                DT10            ;#TESTN,#ERRPC,LKS
1639 001534  000000                0
1640
1641 001536  021706                EM40            ;"DUAL ADDRESSING ERROR"
1642 001540  024465                DM40            ;"TEST# ERR PC  GOOD    BAD"
1643 001542  025146                DT40            ;#TESTN,#ERRPC,#GDADR,#BDCSR
1644 001544  000000                0
1645
1646 001546  021734                EM41            ;"BIT7 OF LKS NOT SET AFTER RESET2
1647 001550  024441                DM10            ;"TEST# ERR PC  LKS"
1648 001552  025136                DT10            ;#TESTN,#ERRPC,LKS
1649 001554  000000                0
```

ERROR POINTER TABLE

```
1650 001556  021774                EM42            ;"CAN NOT CLEAR BIT7 OF LKS"
1651 001560  024441                DH10            ;"TEST# ERR PC  LKS"
1652 001562  025136                DT10            ;$TESTN,$ERRPC,LKS
1653 001564  000000                0
1654
1655 001566  022026                EM43            ;"BIT7 OF LKS DOES NOT SET"
1656 001570  024441                DH10            ;"TEST# ERR PC  LKS"
1657 001572  025136                DT10            ;$TESTN,$ERRPC,LKS
1658 001574  000000                0
1659
1660 001576  022057                EM44            ;"RTC INTERRUPT AT PRIORITY 7"
1661 001600  024441                DH10            ;"TEST# ERR PC  LKS"
1662 001602  025136                DT10            ;$TESTN,$ERRPC,LKS
1663 001604  000000                0
1664
1665 001606  022113                EM45            ;"RTC INTERRUPTS WHEN DISABLED"
1666 001610  024441                DH10            ;"TEST# ERR PC  LKS"
1667 001612  025136                DT10            ;$TESTN,$ERRPC,LKS
1668 001614  000000                0
1669
1670 001616  022150                EM46            ;"RTC INTERRUPT DID NOT OCCUR"
1671 001620  024441                DH10            ;"TEST# ERR PC  LKS"
1672 001622  025136                DT10            ;$TESTN,$ERRPC,LKS
1673 001624  000000                0
1674
1675 001626  022150                EM47            ;"RTC INTERRUPT DID NOT OCCUR"
1676 001630  024441                DH10            ;"TEST# ERR PC  LKS"
1677 001632  025136                DT10            ;$TESTN,$ERRPC,LKS
1678 001634  000000                0
1679
1680 001636  022204                EM50            ;"RTC DOUBLE INTERRUPT"
1681 001640  024441                DH10            ;"TEST# ERR PC  LKS"
1682 001642  025136                DT10            ;$TESTN,$ERRPC,LKS
1683 001644  000000                0
1684
1685 001646  022231                EM51            ;"RESET DID NOT CLEAR RTC INTERRUPT"
1686 001650  024441                DH10            ;"TEST# ERR PC  LKS"
1687 001652  025136                DT10            ;$TESTN,$ERRPC,LKS
1688 001654  000000                0
1689
1690 001656  022261                EM52            ;"RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS"
1691 001660  024441                DH10            ;"TEST# ERR PC  LKS"
1692 001662  025136                DT10            ;$TESTN,$ERRPC,LKS
1693 001664  000000                0
```

ERROR POINTER TABLE

```
1694 001666  022336                EM53
1695 001670  024521                DH53            ;"TEST# ERR PC  LKS   CNT1   CNT2"
1696 001672  025160                DT53            ;$TESTN,$ERRPC,LKS,FIRST,SECND
1697 001674  000000                0
1698
1699 001676  022362                EM54            ;"XMIT INTERRUPTS WHEN DISABLED"
1700 001700  024315                DH1             ;"TEST# ERR PC  TCSR"
1701 001702  025076                DT1             ;$TESTN,$ERRPC,TCSR
1702 001704  000000                0
1703
1704 001706  022520                EM55            ;"XMIT DID NOT INTERRUPT"
1705 001710  024315                DH1             ;"TEST# ERR PC  TCSR"
1706 001712  025076                DT1             ;$TESTN,$ERRPC,TCSR
1707 001714  000000                0
1708
1709 001716  022420                EM56            ;"XMIT INTERRUPT AT PRIORITY 7"
1710 001720  024315                DH1             ;"TEST# ERR PC  TCSR"
1711 001722  025076                DT1             ;$TESTN,$ERRPC,TCSR
1712 001724  000000                0
1713
1714 001726  022456                EM57            ;"XMIT INTERRUPTS WITH ENABLE CLEAR"
1715 001730  024315                DH1             ;"TEST# ERR PC  TCSR"
1716 001732  025076                DT1             ;$TESTN,$ERRPC,TCSR
1717 001734  000000                0
1718
1719 001736  022520                EM60            ;"XMIT DID NOT INTERRUPT"
1720 001740  024315                DH1             ;"TEST# ERR PC  TCSR"
1721 001742  025076                DT1             ;$TESTN,$ERRPC,TCSR
1722 001744  000000                0
1723
1724 001746  022547                EM61            ;"XMIT RE-INTERRUPTED"
1725 001750  024315                DH1             ;"TEST# ERR PC  TCSR"
1726 001752  025076                DT1             ;$TESTN,$ERRPC,TCSR
1727 001754  000000                0
1728
1729 001756  022573                EM62            ;"LOADING TBUF DID NOT CLEAR INTERRUPT"
1730 001760  024315                DH1             ;"TEST# ERR PC  TCSR"
1731 001762  025076                DT1             ;$TESTN,$ERRPC,TCSR
1732 001764  000000                0
1733
1734 001766  022640                EM63            ;"RCVR ACTIVE NOT SET"
1735 001770  024367                DH6             ;"TEST# ERR PC  RCSR"
1736 001772  025116                DT6             ;$TESTN,$ERRPC,RCSR
1737 001774  000000                0
```

ERROR POINTER TABLE

```
1738 001776  022664                EM64            ;"RECEIVER DONE NEVER SET"
1739 002000  024367                DM6             ;"TEST# ERR PC  RCSR"
1740 002002  025116                DT6             ;$TESTN,$ERRPC,RCSR
1741 002004  000000                0
1742
1743 002006  022710                EM65            ;"RCVR ACTIVE NOT CLEARED WITH DONE SET2
1744 002010  024367                DM6             ;"TEST# ERR PC  RCSR"
1745 002012  025116                DT6             ;$TESTN,$ERRPC,RCSR
1746 002014  000000                0
1747
1748 002016  022756                EM66            ;"RESET DID NOT CLEAR RCVR DONE"
1749 002020  024367                DM6             ;"TEST# ERR PC  RCSR"
1750 002022  025116                DT6             ;$TESTN,$ERRPC,RCSR
1751 002024  000000                0
1752
1753 002026  023014                EM67            ;"RDR ENABLE SET DID NOT CLEAR RCVR DONE"
1754 002030  024367                DM6             ;"TEST# ERR PC  RCSR"
1755 002032  025116                DT6             ;$TESTN,$ERRPC,RCSR
1756 002034  000000                0
1757
1758 002036  023057                EM70            ;"READING RBUF DID NOT CLEAR RCVR DONE"
1759 002040  024367                DM6             ;"TEST# ERR PC  RCSR"
1760 002042  025116                DT6             ;$TESTN,$ERRPC,RCSR
1761 002044  000000                0
1762
1763 002046  023124                EM71            ;"RCVR INTERRUPTS WITH ENABLE CLEAR"
1764 002050  024367                DM6             ;"TEST# ERR PC  RCSR"
1765 002052  025116                DT6             ;$TESTN,$ERRPC,RCSR
1766 002054  000000                0
1767
1768 002056  023273                EM72            ;"RCVR DID NOT INTERRUPT"
1769 002060  024367                DM6             ;"TEST# ERR PC  RCSR"
1770 002062  025116                DT6             ;$TESTN,$ERRPC,RCSR
1771 002064  000000                0
1772
1773 002066  023166                EM73            ;"RCVR INTERRUPTS AT PRIORITY 7"
1774 002070  024367                DM6             ;"TEST# ERR PC  RCSR"
1775 002072  025116                DT6             ;$TESTN,$ERRPC,RCSR
1776 002074  000000                0
1777
1778 002076  023224                EM74            ;"RCVR INTERRUPT REQUEST PASSED WITH ENABLE CLEAR"
1779 002100  024367                DM6             ;"TEST# ERR PC  RCSR"
1780 002102  025116                DT6             ;$TESTN,$ERRPC,RCSR
1781 002104  000000                0
```

ERROR POINTER TABLE

```
1782 002106  023273                EM75            ;"RCVR DID NOT INTERRUPT
1783 002110  024367                DM6             ;"TEST# ERR PC  RCSR"
1784 002112  025116                DT6             ;$TESTN,$ERRPC,RCSR
1785 002114  000000                0
1786 002116  023322                EM76            ;"RECEIVER RE-INTERRUPTED'
1787 002120  024367                DM6             ;"TEST# ERR PC  RCSR"
1788 002122  025116                DT6             ;$TESTN,$ERRPC,RCSR
1789 002124  000000                0
1790
1791 002126  023346                EM77            ;"READING RBUF DID NOT CLEAR INTERRUPT"
1792 002130  024367                DM6             ;"TEST# ERR PC  RCSR"
1793 002132  025116                DT6             ;$TESTN,$ERRPC,RCSR
1794 002134  000000                0
1795
1796 002136  023413                EM100           ;"RESET DID NOT CLEAR RCVR INTERRUPT"
1797 002140  024367                DM6             ;"TEST# ERR PC  RCSR"
1798 002142  025116                DT6             ;$TESTN,$ERRPC,RCSR
1799 002144  000000                0
1800
1801
1802 002146  023456                EM101           ;"'OR' FLAG DID NOT SET"
1803 002150  024367                DM6
1804 002152  025116                DT6             ;$TESTN,$ERRPC,RCSR
1805 002154  000000                0
1806
1807 002156  023504                EM102           ;"'ERROR' NOT SET WITH 'OR' FLAG"
1808 002160  024367                DM6             ;"TEST# ERR PC  RCSR"
1809 002162  025116                DT6             ;$TESTN,$ERRPC,RCSR
1810 002164  000000                0
1811 002166  023543                EM103           ;"BREAK DID NOT TRANSMIT ALL ZEROES"
1812 002170  024566                DM103           ;"TEST# ERR PC  RCSR  DATA"
1813 002172  025174                DT103           ;$TESTN,$ERRPC,RCSR,$BDDAT
1814 002174  000000                0
1815
1816 002176  023601                EM104           ;"BREAK DID NOT SET FRAMING ERROR"
1817 002200  024367                DM6             ;"TEST# ERR PC  RCSR"
1818 002202  025116                DT6             ;$TESTN,$ERRPC,RCSR
1819 002204  000000                0
1820
1821 002206  023636                EM105           ;"DATA COMPARE ERROR"
1822 002210  024623                DM105           ;"TEST# ERR PC  RCSR  GOOD  BAD"
1823 002212  025206                DT105           ;$TESTN,$ERRPC,RCSR,GD,BD
1824 002214  000000                0
```

ERROR POINTER TABLE

```
1825 002216  023661                    EM106                ;"LOOP-BACK DATA COMPARE ERROR"
1826 002220  024623                    DM105                ;"TEST# ERR PC  RCSR    GOOD    BAD"
1827 002222  025206                    DT105                ;#TESTN,#ERRPC,RCSR,GD,BD
1828 002224  000000                    0
1829
1830 002226  023716                    EM107                ;"TIMEOUT IN EXERCISER TEST"
1831 002230  024367                    DM6                  ;"TEST# ERR PC  RCSR"
1832 002232  025116                    DT6                  ;#TESTN,#ERRPC,RCSR
1833 002234  000000                    0
1834
1835 002236  023750                    EM110                ;"INCORRECT RECEIVE COUNT
1836 002240  024667                    DM110                ;"TEST# ERR PC  RCSR    TRANS   RCV"
1837 002242  025222                    DT110                ;#TESTN,#ERRPC,RCSR,XMTCNT,RCVCNT
1838 002244  000000                    0
1839
1840 002246  024000                    EM111                ;"DATA COMPARE ERROR IN EXERCISER"
1841 002250  024623                    DM105                ;"TEST# ERR PC  RCSR    GOOD    BAD"
1842 002252  025206                    DT105                ;#TESTN,#ERRPC,RCSR,GD,BD
1843 002254  000000                    0
1844
1845 002256  024040                    EM112                ;"TRAP CATCHER"
1846 002260  024733                    DM112                ;"TEST#   ERR PC  RCSR   OLDPC  TRAP ADR"
1847 002262  025236                    DT112                ;#TESTN,#ERRPC,RCSR,OLDPC,BDVECT
1848 002264  000000                    0
1849
1850 002266  024055                    EM113                ;"NO CLK INTERRUPT IN EXERCISER"
1851 002270  024441                    DM10                 ;"TEST#  ERR PC  LKS"
1852 002272  025136                    DT10                 ;#TESTN,#ERRPC,LKS
1853 002274  000000                    0
1854
1855 002276  024113                    EM114                ;"'ERROR' NOT SET WITH 'FR' FLAG"
1856 002300  024367                    DM6                  ;"TEST# ERR PC  RCSR"
1857 002302  025116                    DT6                  ;#TESTN,#ERRPC,RCSR
1858 002304  000000                    0
1859
1860 002306  024152                    EM115                ;RCV ACTVIE NOT CLEAR WITH INIT
1861 002310  024367                    DM6                  ;"TEST# ERR PC  RCSR"
1862 002312  025116                    DT6                  ;#TESTN,#ERRPC,RCSR
1863 002314  000000                    0
1864
1865 002316  024211                    EM116                ;RCV ACTIVE WITHOUT "START" BIT
1866 002320  024367                    DM6                  ;"TEST# ERR PC  RCSR"
1867 002322  025116                    DT6                  ;#TESTN,#ERRPC,RCSR
1868 002324  000000                    0
1869
1870 002326  024250                    EM117                ;RDR ENABLE NOT CLEAR WITH RCV ACTIVE
1871 002330  024367                    DM6                  ;"TEST# ERR PC  RCSR"
1872 002332  025116                    DT6                  ;#TESTN,#ERRPC,RCSR
1873 002334  000000                    0
1874
1875                          ;STORAGE LOCATIONS
1876 002336  000000          FIRST:   .WORD   0
1877 002340  000000          LOC1:    .WORD   0           ;TIMER LOOP COUNTER
1878 002342  000000          LOC2:    .WORD   0           ;TIMER LOOP COUNTER
1879 002344  000000          SAVEO:   .WORD   0           ;STORAGE FOR LOCATIONS
1880 002346                  SAVLOC:  .BLKW   22          ;THAT T/E USES
1881 002412                  ENDSTK:  .BLKW   10          ;JIMS SPECIAL STACK (WRAP AROUND)
```

ERROR POINTER TABLE

```
1882 002432  000000                JIMSTK: .WORD   0
1883 002434  000000                SAVEPS: .WORD   0                ;SAVE PSW AREA
1884 002436  000000                OLDSUM: .WORD   0                ;CHECKSUM STORAGE
1885 002440  000000                RCVCNT: .WORD   0
1886 002442  000000                XMTCNT: .WORD   0
1887 002444  000000                CLKCNT: .WORD   0
1888 002446  000000                STPSW:  .WORD   0
1889 002450  000000                SRPSW:  .WORD   0
1890 002452  000000                SCPSW:  .WORD   0
1891 002454                        BUF:    .BLKW   44
1892 002564     020     000        CNTLP:  .ASCIZ  <20>
1893 002566     136     120   000  PROMPT: .ASCII  /↑P/<0><CR><LF>/CONSOLE/<CR><LF>/>>>/<377>
1894 002610     124     057   101  TA:     .ASCIZ  |T/A|<CR><LF>
1895
1896                               .EVEN
1897 002616  000000                SECND:  .WORD   0
1898 002620  000000                OLDPC:  .WORD   0
1899 002622  000000                BDVECT: .WORD   0
1900
1901
1902
1903
1904
1905 002624  000000                CTSTFL: .WORD   0                ;CONSLE UNDER TEST FLAG
1906 002626  000000                TMP1:   .WORD   0                ;TEMP LOCATION FOR TABLE OFFSETS
1907 002630  000000                TMP2:   .WORD   0                ;TEMP LOCAYION FOR DEVICE COUNT
1908 002632  000000                TMP3:   .WORD   0                ;LOCATION FOR DEVICE MAP BIT TEST MASK
1909                               ;REGISTER AND VECTOR ADDRESSES FOR THE DL-11W UNDER TEST
1910
1911 002634  000000                RCSR:   .WORD   0
1912 002636  000000                RBUF:   .WORD   0
1913 002640  000000                TCSR:   .WORD   0
1914 002642  000000                TBUF:   .WORD   0
1915 002644  000000                RVECT:  .WORD   0
1916 002646  000000                RPSW:   .WORD   0
1917 002650  000000                TVECT:  .WORD   0
1918 002652  000000                TPSW:   .WORD   0
1919
1920                               ;CONSOLE REGISTER AND VECTOR ADDRESSES FOR THE DL-11W
1921
1922 002654  177560                CRCSR:  177560                   ;ADDRESS OF RECEIVER COMMAND/STATUS REGISTER
1923 002656  177562                CRBUF:  177562                   ;ADDRESS OF RECEIVER BUFFER
1924 002660  177564                CTCSR:  177564                   ;ADDRESS OF TRANSMITTER COMMAND/STATUS REGISTER
1925 002662  177566                CTBUF:  177566                   ;ADDRESS OF TRANSMITTER BUFFER
1926 002664  000060                CRVECT: 60                       ;RECEIVER INTERRUPT VECTOR
1927 002666  000062                CRPSW:  62
1928 002670  000064                CTVECT: 64                       ;TRANSMITTER INTERRUPT VECTOR
1929 002672  000066                CTPSW:  66
1930
1931                               ;REAL TIME CLOCK REGISTER AND VECTOR ADDRESSES
1932 002674  177546                LKS:    .WORD   177546
1933 002676  000100                RTCVT:  .WORD   100
1934 002700  000102                RTCPSW: .WORD   102
1935
1936 002702                        ADRTBL: .BLKW   20
1937 002742                        VCTTBL: .BLKW   20
1938 003002  000000                FLAG44: .WORD   0
```

ERROR POINTER TABLE

```
1939 003004  176500                    TURCSR: .WORD   176500
1940 003006  176502                    TURBUF: .WORD   176502
1941 003010  176504                    TUTCSR: .WORD   176504
1942 003012  176506                    TUTBUF: .WORD   176506
1943
1944
1945                                    ;SUBROUTINE TO GENERATE DEVICE ADDRESS TABLE
1946
1947 003014  012702  002702            DEVADR: MOV     #ADRTBL,R2      ;POINT R2 TO THE DEVICE ADDRESS TABLE
1948 003020  013700  001142                    MOV     $BASE,R0        ;LOAD BASE DEVICE ADDRESS IN R0
1949 003024  010001                            MOV     R0,R1           ;
1950 003026  062701  000170                    ADD     #170,R1         ;POINT R1 TO LAST DEVICE ADDRESS
1951 003032  010022                    1$:     MOV     R0,(R2)+        ;MOVE DEVICE ADDRESS TO TABLE
1952 003034  062700  000010                    ADD     #10,R0          ;POINT R0 TO NEXT DEVICE ADDRESS
1953 003040  020001                            CMP     R0,R1           ;FINISHED GENERATING TABLE?
1954 003042  003773                            BLE     1$              ;BR, IF LAST DEVICE ADDRESS NOT LOADED
1955 003044  000207                            RTS     PC
1956
1957
1958
1959 003046  005037  001070            START:  CLR     $FATAL          ;CLEAR ERROR NO.
1960 003052  005037  001066                    CLR     $MSGTYP         ;CLEAR MESSAGE TYPE
1961 003056  005037  001072                    CLR     $TESTN          ;CLEAR TEST NO.
1962 003062  005037  002624                    CLR     CTSTFL          ;CLEAR CONSOLE UNDER TEST FLAG
1963 003066  005037  001076                    CLR     $DEVCT          ;CLEAR DEVICE COUNT
1964 003072  005037  001100                    CLR     $UNIT           ;CLEAR UNIT NUMBER
1965 003076  005037  003002                    CLR     FLAG44          ;** CLEAR 11/44 CPU FLAG
1966 003102  005737  001112                    TST     $USWR           ;IS $USWR LOADED?
1967 003106  001003                            BNE     1$              ;BR IF YES
1968 003110  012737  000400  001112            MOV     #400,$USWR      ;ELSE, DEFAULT TO $USWR=400
1969 003116  012737  000006  000004    1$:     MOV     #6,@#4          ;INITIALIZE TIMEOUT VECTORS TO TRAP
1970 003124  012737  000003  000006            MOV     #3,@#6          ;  CATCHER ROUTINE
1971
1972                                    .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
     003132  012706  001000                    MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
     003136  005026                            CLR     (R6)+           ;;CLEAR MEMORY LOCATION
     003140  022706  001040                    CMP     #$WR,R6 ;;DONE?
     003144  001374                            BNE     .-6             ;;LOOP BACK IF NO
     003146  012706  001000                    MOV     #1000,SP                ;;SETUP THE STACK POINTER
                                        ;;INITIALIZE A FEW VECTORS
     003152  012737  015374  000020            MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
     003160  012737  000340  000022            MOV     #340,@#IOTVEC+2 ;;LEVEL 7
     003166  012737  014544  000030            MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
     003174  012737  000340  000032            MOV     #340,@#EMTVEC+2 ;;LEVEL 7
     003202  012737  017522  000034            MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
     003210  012737  000340  000036            MOV     #340,@#TRAPVEC+2;LEVEL 7
     003216  012737  015212  000024            MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
     003224  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;;LEVEL 7
     003232  013737  014356  014350            MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
     003240  005037  001060                    CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
     003244  112737  000001  001015            MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
     003252  012737  003252  001006            MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
     003260  012737  003260  001010            MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
     003266  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
```

INITIALIZE THE COMMON TAGS

```
003272  012737  003326  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
003300  012737  177570  001040          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
003306  012737  177570  001042          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
003314  022777  177777  175516          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
003322  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = 1
003324  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
003326  012716  003334          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
003332  000002                          RTI
003334  012737  000176  001040  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
003342  012737  000174  001042          MOV     #DISPREG,DISPLAY
003350  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
003354  005037  001074                  CLR     $PASS           ;;CLEAR PASS COUNT
003360  132737  000200  001107          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
003366  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
003370  012737  001110  001040          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
003376                          67$:

1973
1974                                    ;SIZE FOR 11/44 CPU
1975
1976 003376  013746  000010            MOV     @#10,-(SP)      ;** SAVE VECTOR
1977 003402  012737  003430  000010      MOV     #10$,@#10       ;** SET UP FOR TRAP
1978 003410  000007                    MFPT                    ;** WHAT CPU??
1979 003412  122700  000001            CMPB    #1,R0           ;** ARE WE A 11/44
1980 003416  001007                    BNE     11$             ;** NO GO RESET VECTOR
1981 003420  052737  000001  003002    BIS     #1,@#FLAG44     ;** YES SET FLAG
1982 003426  000403                    BR      11$             ;** SKIP RTI
1983 003430  012716  003436    10$:    MOV     #11$,(SP)       ;** SET STACK RETURN
1984 003434  000002                    RTI                     ;** RETURN
1985 003436  012637  000010    11$:    MOV     (SP)+,@#10      ;** RESTORE VECTOR
1986 003442  032777  000020  175370    BIT     #BIT4,@SWR      ;TEST CLOCK ONLY?
1987 003450  001404                    BEQ     INIT            ;BR IF NOT
1988 003452  005237  002624            INC     CTSTFL          ;ELSE, SET CONSOLE TEST FLAG TO ENABLE CLOCK TESTS
1989 003456  000137  004620            JMP     ID              ;  AND JUMP TO TYPE PROGRAM ID
1990 003462  132737  000001  001106  INIT:   BITB    #BIT0,$ENV      ;CHECK IF ON APT
1991 003470  001404                    BEQ     MANL            ;BR IF NOT APT
1992 003472  132737  000200  001107          BITB    #BIT7,$ENVM     ;DID APT SIZE
1993 003500  001056                    BNE     APTSZD          ;BR, IF APT SIZED
1994 003502  032777  000040  175330  MANL:   BIT     #BIT5,@SWR      ;WAS "$DEVM" MANUALLY SET?
1995 003510  001052                    BNE     APTSZD          ;IF YES, SKIP SELF-SIZING
1996
1997 003512  004737  003014    SIZE:   JSR     PC,DEVADR       ;GENERATE DEVICE ADDRESS TABLE
1998 003516  005037  002630            CLR     TMP2            ;CLR TEMP LOCATION TO KEEP DEVICE COUNT
1999 003522  005037  001144            CLR     $DEVM           ;CLEAR DEVICE MAP
2000 003526  013703  000004            MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
2001 003532  012737  003562  000004      MOV     #4$,@#4         ;SET TIMEOUT POINTER
2002 003540  013700  001142            MOV     $BASE,R0        ;LOAD BASE ADDRESS
2003 003544  062700  000160            ADD     #160,R0         ;POINT R0 TO UNIT #15 (UNIT#0=CONSOLE)
2004 003550  005710              3$:   TST     (R0)            ;CHECK FOR DEVICE EXISTANCE
2005 003552  005237  001144            INC     $DEVM           ;INDICATE DEVICE EXISTANCE IN DEVICE MAP
2006 003556  005237  002630            INC     TMP2            ;INCREMENT DEVICE COUNT
2007 003562  012706  001000      4$:   MOV     #1000,SP        ;RESET STACK POINTER
2008 003566  006337  001144            ASL     $DEVM           ;ADJUST DEVICE MAP FOR NEXT UNIT CHECK
2009 003572  162700  000010            SUB     #10,R0          ;POINT R0 TO NEXT DEVICE NUMBER
2010 003576  023700  001142            CMP     $BASE,R0        ;FINISHED SIZING?
2011 003602  003762                    BLE     3$              ;BR, IF BASE ADDRESS HAS NOT BEEN CHECKED
2012 003604  013700  002654            MOV     CRCSR,R0        ;LOAD CONSOLE DEVICE ADDRESS
```

INITIALIZE THE COMMON TAGS

```
2013 003610  012737  003630  000004          MOV     #5$,@#4          ;SET UP TIMEOUT POINTER
2014 003616  005710                          TST     (R0)             ;TEST FOR CONSOLE EXISTANCE
2015 003620  005237  001144                  INC     $DEVM            ;INDICATE CONSOLE EXISTANCE IN DEVICE MAP
2016 003624  005237  002630                  INC     TMP2             ;INCREMENT DEVICE COUNT
2017 003630  010337  000004          5$:     MOV     R3,@#4           ;RESTORE TIMEOUT VECTOR
2018
2019 003634  000415                          BR      VCTADR           ;BR TO GENERATE VECTOR ADDRESS TABLE
2020
2021 003636  005037  002630          APTSZD: CLR     TMP2             ;CLEAR TEMP LOCATION TO KEEP DEVICE CNT
2022 003642  013702  001144                  MOV     $DEVM,R2         ;MOVE DEVICE MAP TO R2
2023 003646  005702                  TSTDVM: TST     R2               ;TEST MSB OF DEVICE MAP
2024 003650  100002                          BPL     1$               ;BR, IF MSB IS ZERO
2025 003652  005237  002630                  INC     TMP2             ;INCREMENT DEVICE COUNT, IF MSB=1
2026 003656  006302                  1$:     ASL     R2               ;SHIFT NEXT BIT INTO MSB POSITION
2027 003660  001401                          BEQ     DVADT            ;BR, IF NO OTHER BITS ARE SET IN $DEVM
2028 003662  000771                          BR      TSTDVM           ;CONTINUE CHECKING $DEVM, IF MORE BITS SET
2029 003664  004737  003014          DVADT:  JSR     PC,DEVADR        ;GENERATE DEVICE ADDRESS TABLE
2030
2031                                 ;GENERATE VECTOR ADDRESS TABLE
2032
2033 003670  012702  002742          VCTADR: MOV     #VCTTBL,R2       ;GET LOCATION OF VECTOR TABLE
2034 003674  013700  001136                  MOV     @#$VECT1,R0      ;COPY BASE VECTOR
2035 003700  042700  177000                  BIC     #177000,R0       ;CLEAR UPPER BITS
2036 003704  010001                          MOV     R0,R1            ;
2037 003706  062701  000170                  ADD     #170,R1          ;POINT R1 TO LAST DEVICE VECTOR
2038 003712  010022                  1$:     MOV     R0,(R2)+         ;PUT VECTOR ADDRESS IN TABLE
2039 003714  062700  000010                  ADD     #10,R0           ;POINT R0 TO NEXT VECTOR ADDRESS
2040 003720  020001                          CMP     R0,R1            ;FINISHED GENERATING VECTOR TABLE?
2041 003722  003773                          BLE     1$               ;BR, IF LAST VECTOR IS NOT LOADED
2042
2043                                 ;MOVE DEVICE COUNT INTO DEVICE COUNT MESSAGE
2044
2045 003724  013700  002630                  MOV     TMP2,R0          ;COPY DEVICE COUNT INTO R0
2046 003730  005001                          CLR     R1               ;CLEAR AUXILARY REGISTER
2047 003732  000300                          SWAB    R0               ;PUT DEVICE COUNT IN UPPER BYTE OF R0
2048 003734  006300                          ASL     R0               ;MOVE MSB OF COUNT INTO
2049 003736  006300                          ASL     R0               ;MSB OF R0
2050 003740  006300                  SHIFT:  ASL     R0               ;PUT MSB OF COUNT INTO CARRY
2051 003742  106101                          ROLB    R1               ;MOVE MSB OF COUNT INTO R1
2052 003744  006300                          ASL     R0               ;MOVE NEXT BIT TO CARRY
2053 003746  106101                          ROLB    R1               ;MOVE INTO R1
2054 003750  006300                          ASL     R0               ;MOVE LAST BIT OF DIGIT
2055 003752  106101                          ROLB    R1               ;INTO R1
2056 003754  062701  000060                  ADD     #60,R1           ;CONVERT DIGIT TO ASCII
2057 003760  000301                          SWAB    R1               ;MOVE DIGIT TO UPPER BYTE
2058 003762  032701  000020                  BIT     #BIT4,R1         ;HAVE BOTH DIGITS BEEN MOVED TO R1?
2059 003766  001764                          BEQ     SHIFT            ;BR, IF NOT
2060 003770  010137  025046                  MOV     R1,M2A           ;MOVE DEVICE COUNT TO OUTPUT MESSAGE
2061
2062
2063 003774  052737  000002  002632  BEGIN:  BIS     #BIT1,TMP3       ;SET UP BIT MASK TO TEST $DEVM FOR DEVICES EXCEPT CONSOLE
2064 004002  005037  002626                  CLR     TMP1             ;CLEAR LOCATION TO STORE TABLE OFFSETS
2065 004006  032737  000001  001144          BIT     #BIT0,$DEVM      ;IS CONSOLE TO BE TESTED?
2066 004014  001001                          BNE     TCONS            ;BR, IF CONSOLE IS TO BE TESTED
2067 004016  000414                          BR      TSTDEV           ;BR, TO TEST OTHER DEVICES
2068 004020  005237  002624          TCONS:  INC     CTSTFL           ;INDICATE CONSOLE UNDER TEST
2069 004024  012700  002654                  MOV     #CRCSR,R0        ;SET UP CONSOLE DEVICE ADDRESSES
```

```
 2070 004030  012701  002634          MOV     #RCSR,R1        ;POINT R1 TO UUT ADDRESS TABLE
 2071 004034  012021          1$:    MOV     (R0)+,(R1)+     ;TRANSFER CONSOLE ADDRESSES
 2072 004036  022701  002652          CMP     #TPSW,R1        ;FINISHED TRANSFER?
 2073 004042  002374                  BGE     1$              ;BR, IF NOT
 2074 004044  000137  004172          JMP     TST1            ;GO TEST CONSOLE INTERFACE
 2075
 2076                         ;PREPARE ADDRESSES AND VECTORS FOR UUT
 2077 004050  033737  002632  001144 TSTDEV: BIT     TMP3,$DEVM      ;CHECK TO SEE IF DEVICE IS TO BE TESTED
 2078 004056  001010                  BNE     SETADR          ;BR, IF YES
 2079 004060  006337  002632          ASL     TMP3            ;SHIFT MASK TO CHECK NEXT $DEVM BIT
 2080 004064  062737  000002  002626  ADD     #2,TMP1         ;INCREMENT TABLE INDEX
 2081 004072  005237  001100          INC     $UNIT           ;INCREMENT UNIT NUMBER
 2082 004076  000764                  BR      TSTDEV          ;GO TEST NEXT BIT OF DEVICE MAP
 2083
 2084 004100  005237  001100 SETADR: INC     $UNIT           ;UPDATE UNIT NUMBER
 2085 004104  006337  002632          ASL     TMP3            ;UPDATE DEVICE MAP TEST MASK
 2086 004110  013702  002626          MOV     TMP1,R2         ;MOVE TABLE OFFSET TO R2
 2087 004114  062737  000002  002626  ADD     #2,TMP1         ;UPDATE TABLE OFFSET FOR NEXT DEVICE
 2088 004122  016200  00270?          MOV     ADRTBL(R2),R0   ;PUT UUT ADDRESS INTO R0
 2089 004126  012701  002634          MOV     #RCSR,R1        ;POINT R1 TO STORAGE AREA FOR UUT ADDRESSES
 2090 004132  010021          ADR:    MOV     R0,(R1)+        ;TRANSFER UUT ADDRESS
 2091 004134  062700  000002          ADD     #2,R0           ;POINT TO NEXT UUT REGISTER
 2092 004140  030027  000006          BIT     R0,#6           ;FINISHED TRANSFER?
 2093 004144  001372                  BNE     ADR             ;BR, IF NOT
 2094
 2095 004146  016200  002742          MOV     VCTTBL(R2),R0   ;PUT UUT VECTOR INTO R0
 2096 004152  010021          VECT:   MOV     R0,(R1)+        ;TRANSFER UUT VECTORS TO ACTIVE TABLE AREA
 2097 004154  062700  000002          ADD     #2,R0           ;POINT TO NEXT VECTOR
 2098 004160  030027  000006          BIT     R0,#6           ;FINISHED TRANSFER?
 2099 004164  001372                  BNE     VECT            ;BR, IF NOT
 2100 004166  000137  004172          JMP     TST1            ;GO TEST DEVICE
```

```
2101                                    .SBTTL  TEST # 1 - TEST ABILITY TO REFERENCE TCSR
                                        ;;*****************************************************************
                                        ;*TEST 1        TEST ABILITY TO REFERENCE TCSR
                                        ;;*****************************************************************
     004172  000004                     TST1:   SCOPE
2102 004174  013703  000004                     MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
2103 004200  012737  004214  000004            MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
2104 004206  005777  176426                     TST     @TCSR           ;REFERENCE THE XMIT COMMAND/STATUS REG.
2105 004212  000412                             BR      4$              ; GO TO END OF TEST
2106
2107 004214  022626                     1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
2108 004216  005737  002624                     TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
2109 004222  001002                             BNE     2$              ;IF YES, SKIP ERROR TYPEOUT
2110 004224  104001                             ERROR   .1              ;REPORT ERROR TO APT & TTY
2111 004226  000404                             BR      4$              ;BR TO END OF TEST
2112 004230                             2$:
     004230  004737  015714                     JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
     004234  000001                             1                               ;;THE ERROR NUMBER (FROM APT LIST)
2113 004236  000000                     3$:     HALT
2114 004240  010337  000004             4$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
2115
2116
2117
```

```
2118                                    .SBTTL  TEST # 2 - TEST ABILITY TO REFERENCE TBUF
                                        ;;*****************************************************************
                                        ;*TEST 2         TEST ABILITY TO REFERENCE TBUF
                                        ;;*****************************************************************
     004244  000004                     TST2:   SCOPE
2119 004246  013703  000004                     MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
2120 004252  012737  004266  000004            MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
2121 004260  005777  176356                     TST     @TBUF           ;REFERENCE THE XMIT BUFFER
2122 004264  000412                             BR      4$              ;GO TO END OF  TEST
2123
2124 004266  022626                     1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
2125 004270  005737  002624                     TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
2126 004274  001002                             BNE     2$              ;IF YES, SKIP ERROR TYPEOUT
2127 004276  104002                             cRROR   .2              ;REPORT ERROR TO APT & TTY
2128 004300  000404                             BR      4$              ;BR TO END OF TEST
2129 004302                             2$:
     004302  004737  015714                     JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
     004306  000002                             2                       ;;THE ERROR NUMBER (FROM APT LIST)
2130 004310  000000                     3$:     HALT
2131 004312  010337  000004             4$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
```

```
2132                                    .SBTTL  TEST # 3 - TEST THAT BIT2(MAINT. BIT) CAN BE SET & RESET
                                        ;;**********************************************************************
                                        ;*TEST 3         TEST THAT BIT2(MAINT. BIT) CAN BE SET & RESET
                                        ;;**********************************************************************
     004316  000004                     TST3:   SCOPE
2133 004320  000005                             RESET                   ;CLEAR EVERYTHING
2134 004322  032777  000004  176310             BIT     @#BIT2,@TCSR    ;TEST FOR BIT2 OF TCSR CLEAR
2135 004330  001411                             BEQ     3$              ;BR IF CLEAR
2136
2137 004332  005737  002624                     TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
2138 004336  001002                             BNE     1$              ;IF YES, SKIP ERROR TYPEOUT
2139 004340  104015                             ERROR   *15             ;BIT2 OF TCSR NOT CLEAR AFTER RESET
2140 004342  000404                             BR      3$
2141
2142 004344                             1$:
     004344  004737  015714                     JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
     004350  000015                             15                              ;;THE ERROR NUMBER (FROM APT LIST)
2143 004352  000000                     2$:     HALT
2144
2145 004354  052777  000004  176256     3$:     BIS     @#BIT2,@TCSR    ;SET BIT2 OF TCSR
2146 004362  032777  000004  176250             BIT     @#BIT2,@TCSR    ;TEST FOR BIT2 SET
2147 004370  001001                             BNE     4$              ;BR IF SET
2148
2149 004372  104016                             ERROR   *16             ;BIT2 OF TCSR WILL NOT SET
2150
2151 004374  042777  000004  176236     4$:     BIC     @#BIT2,@TCSR    ;CLEAR BIT2 OF TCSR
2152 004402  032777  000004  176230             BIT     @#BIT2,@TCSR    ;TEST BIT2 CLEAR
2153 004410  001411                             BEQ     7$              ;BR IF CLEAR
2154
2155 004412  005737  002624                     TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
2156 004416  001002                             BNE     5$              ;IF YES, SKIP ERROR TYPEOUT
2157 004420  104017                             ERROR   *17
2158 004422  000404                             BR      7$
2159 004424                             5$:
     004424  004737  015714                     JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
     004430  000017                             17                              ;;THE ERROR NUMBER (FROM APT LIST)
2160 004432  000000                     6$:     HALT                    ;BIT0 OF TCSR WILL NOT CLEAR
2161
2162 004434  052777  000004  176176     7$:     BIS     @#BIT2,@TCSR    ;SET BIT2 OF TCSR
2163 004442  000005                             RESET                   ;CLEAR BIT2 WITH RESET
2164 004444  032777  000004  176166             BIT     @#BIT2,@TCSR    ;TEST FOR BIT2 CLEAR
2165 004452  001404                             BEQ     10$             ;** IF CLEAR, GO TO NEXT TEST
2166
2167 004454  042777  000004  176156             BIC     @#BIT2,@TCSR    ;CLEAR BIT2, TO PRINT ERROR
2168 004462  104020                             ERROR   *20
2169                                                                    ;RESET DID NOT CLEAR BIT2 OF TCSR
2170
2171 004464  000240                     10$:    NOP                     ;**
```

```
2172                                    .SBTTL  TEST # 4 - TEST THAT TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
                                        ;;**********************************************************************
                                        ;*TEST 4         TEST THAT TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
                                        ;;**********************************************************************
     004466  000004                     TST4:   SCOPE
2173 004470  052777  000004  176142             BIS     @#BIT2,@TCSR    ;** ENABLE MAINT. WRAP
2174 004476  005077  176140                     CLR     @TBUF           ;LOAD XBUF
2175 004502  105777  176132                     TSTB    @TCSR           ;CHECK DONE
2176 004506  100021                             BPL     3$              ;BR IF CLEAR
2177                                            ;FILL SECOND BUFFER, BECAUSE REFRESH COULD CAUSE
2178                                            ;   FIRST TEST TO FAIL
2179 004510  005077  176126                     CLR     @TBUF           ;FILL DOUBLE BUFFER
2180 004514  105777  176120                     TSTB    @TCSR           ;CHECK DONE
2181 004520  100014                             BPL     3$              ;BR IF CLEAR
2182
2183 004522  005737  002624                     TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
2184 004526  001005                             BNE     1$              ;IF YES, SKIP ERROR TYPEOUT
2185 004530  042777  000004  176122             BIC     @#BIT2,@CTCSR   ;** DISABLE MAINTENANCE MODE FOR
                                                                        ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                        ;** WITH TERMINAL.
2186 004536  104003                             ERROR   +3              ;DONE NOT CLEARED WITH TBUF FULL
2187 004540  000404                             BR      3$              ;BR TO END OF TEST
2188 004542                             1$:
     004542  004737  015714                     JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
     004546  000003                             3                               ;;THE ERROR NUMBER (FROM APT LIST)
2189 004550  000000                     2$:     HALT                    ;TCSR "DONE" NOT CLEARED WITH TBUF FULL
2190 004552  005000                     3$:     CLR     R0              ;CLEAR TIMER
2191 004554  105777  176060             4$:     TSTB    @TCSR           ;CHECK FOR XMIT DONE
2192 004560  100417                             BMI     ID              ;IF DONE SETS, BR TO END OF TEST
2193 004562  005200                             INC     R0              ;INCREMENT TIMER
2194 004564  001373                             BNE     4$              ;BR IF TIMER NOT DONE
2195
2196 004566  005737  002624                     TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
2197 004572  001005                             BNE     5$
2198 004574  042777  000004  176056             BIC     @#BIT2,@CTCSR   ;** DISABLE MAINTENANCE MODE FOR
                                                                        ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                        ;** WITH TERMINAL.
2199 004602  104004                             ERROR   +4              ;TCSR "DONE" DOES NOT SET
2200 004604  000405                             BR      ID              ;BR TO END OF TEST
2201 004606                             5$:
     004606  004737  015714                     JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
     004612  000004                             4                               ;;THE ERROR NUMBER (FROM APT LIST)
2202 004614  000000                             HALT
2203 004616  000427                             BR      END7            ;** BR TO NEXT TEST,
2204                                                                    ;** AND SKIP THE TYPEOUT THAT FOLLOWS
2205                                                                    ;**   BECAUSE OF THIS FAILURE
2206
2207 004620                             ID:
2208 004620  042777  000004  176032             BIC     @#BIT2,@CTCSR   ;** DISABLE MAINTENANCE MODE FOR
                                                                        ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                        ;** WITH TERMINAL.
2209 004626  023737  000042  000046             CMP     @#42,@#46       ;UNDER ACT11?
2210 004634  001412                             BEQ     6$              ;IF YES, SKIP IDENT. TYPEOUT
2211 004636  005737  001074                     TST     $PASS           ;IS THIS THE FIRST PASS?
2212 004642  001007                             BNE     6$              ;IF NOT BR TO NEXT TEST & SKIP THE IDENTIFICATION TYPEOUTS
2213 004644  005737  001076                     TST     $DEVCT          ;IS THIS THE FIRST SUBPASS?
2214 004650  001004                             BNE     6$              ;IF NOT, BRTO NEXT TEST
```

```
2215 004652  104401                          TYPE                    ;TYPE PROGRAM INDENTIFICATION
2216 004654  025004                          M1
2217 004656  104401                          TYPE                    ;TYPE NUMBER OF DEVICES UNDER TEST
2218 004660  025044                          M2
2219 004662  032777  000020  174150  6$:     BIT     @#BIT4,@#SWR    ;CLOCK TEST ONLY?
2220 004670  001402                          BEQ     END87           ;** BR IF NOT
2221 004672  000137  005536                  JMP     TCLOCK          ;ELSE, JUMP TO TEST CLOCK
2222 004676                          END87:
2223 004676  005000                          CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     004700  012701  000002                  MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
     004704  005300                  40$:     DEC     R0              ;** TRANSMITTED TO FINISH BEFORE MAINTENANCE
     004706  001376                          BNE     40$             ;** WRAP FOR THE UART UNDER TEST IS DISABLED.
     004710  005301                          DEC     R1              ;** THIS WILL INHIBIT ANY COMMUNICATION
     004712  001374                          BNE     40$             ;** TO HARDWARE MEDIA SUCH AS TU58 THAT MIGHT
                                                                     ;** BE ATTACHED TO UART UNDER TEST.
```

```
2224                                    .SBTTL  TEST # 5 - TEST THAT TCSR "DONE" SETS WITH RESET
                                        ;;**********************************************************************
                                        ;*TEST 5         TEST THAT TCSR "DONE" SETS WITH RESET
                                        ;;**********************************************************************
     004714  000004                     TST5:   SCOPE
2225 004716  052777  000004  175714             BIS     #BIT2,@TCSR     ;** ENABLE MAINT. WRAP
2226 004724  005077  175712                     CLR     @TBUF           ;LOAD TRANSMIT BUFFER
2227 004730  105777  175704             1$:     TSTB    @TCSR           ;WAIT FOR DONE
2228 004734  100375                             BPL     1$
2229 004736  005077  175700                     CLR     @TBUF           ;LOAD SECOND BUFFER
2230 004742  000240                             NOP
2231 004744  000005                             RESET                   ;CLEAR DONE WITH RESET
2232 004746  105777  175666                     TSTB    @TCSR           ;CHECK FOR DONE SET
2233 004752  100401                             BMI     10$             ;** BR TO NEXT TEST IF DONE SET
2234
2235 004754  104005                             ERROR   +5              ;TCSR "DONE" DOES NOT SET WITH RESET
2236
2237 004756  000240                     10$:    NOP                     ;**
2238
2239
```

```
                                        .SBTTL  TEST # 6 - TEST ABILITY TO ACCESS RCSR
                                        ;;*****************************************************************
                                        ;*TEST 6         TEST ABILITY TO ACCESS RCSR
                                        ;;*****************************************************************
004760  000004                  TST6:   SCOPE
2241 004762  013703  000004            MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
2242 004766  012737  005002  000004    MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
2243 004774  005777  175634            TST     @RCSR           ;ACCESS RCSR
2244 005000  000402                    BR      2$              ;BR TO END OF TEST
2245
2246 005002  022626            1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
2247 005004  104006                    ERROR   +6              ;CAN NOT ACCESS RCSR
2248 005006  010337  000004    2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
```

```
2249                                    .SBTTL  TEST # 7 - TEST ABILITY TO ACCESS RBUF
                                        ;;*****************************************************************
                                        ;*TEST 7         TEST ABILITY TO ACCESS RBUF
                                        ;;*****************************************************************
     005012  000004                     TST7:   SCOPE
2250 005014  013703  000004                     MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
2251 005020  012737  005034  000004            MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
2252 005026  005777  175604                     TST     @RBUF           ;ACCESS RBUF
2253 005032  000402                             BR      2$              ;BR TO END OF TEST
2254
2255 005034  022626                     1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
2256 005036  104007                             ERROR   #7              ;CAN NOT ACCESS RBUF
2257 005040  010337  000004             2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
```

```
2258                                    .SBTTL  TEST # 10 - TEST THAT BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET
                                        ;;**********************************************************************
                                        ;*TEST 10        TEST THAT BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET
                                        ;;**********************************************************************
     005044  000004                     TST10:  SCOPE
2259 005046  032777  000400  173764             BIT     @BIT8,@SWR      ;IS BREAK FUNCTION ENABLED?
2260 005054  001475                             BEQ     10$             ;** BR TO NEXT TEST, IF NOT
2261 005056  032737  000001  003002             BIT     @BIT0,FLAG44    ;** IS THIS A 11/44
2262 005064  001407                             BEQ     9$              ;** NO
2263 005066  005737  002624                     TST     CTSTFL          ;** YES THIS IS 11/44. IS THIS THE CONSOLE
2264                                                                     ;** SLU
2265 005072  001404                             BEQ     9$              ;** NO
2266 005074  032777  000010  173736             BIT     @BIT03,@SWR     ;** THIS IS THE CONSOLE SLU.SHOULD THE BREAK
2267                                                                     ;** TEST BE PERFORMED
2268 005102  001462                             BEQ     10$             ;** NO
2269
2270 005104  000005                     9$:     RESET                   ;CLEAR EVERYTHING
2271 005106  032777  000001  175524             BIT     @BIT0,@TCSR     ;CHECK BIT0 OF TCSR CLEAR
2272 005114  001411                             BEQ     3$              ;BR IF CLEAR
2273 005116  005737  002624                     TST     CTSTFL
2274 005122  001002                             BNE     1$
2275 005124  104011                             ERROR   .11             ;BIT0 WAS NOT CLEAR AFTER RESET
2276 005126  000404                             BR      3$
2277 005130                             1$:
     005130  004737  015714                     JSR     PC,@ATY4                ;;ONLY REPORT A FATAL ERROR
     005134  000011                             11                      ;;THE ERROR NUMBER (FROM APT LIST)
2278 005136  000000                     2$:     HALT
2279
2280 005140  052777  000001  175472     3$:     BIS     @BIT0,@TCSR     ;SET BIT0 IN TCSR
2281 005146  032777  000001  175464             BIT     @BIT0,@TCSR     ;TEST BIT0 OF TCSR
2282 005154  001001                             BNE     4$              ;BR IF SET
2283
2284 005156  104012                             ERROR   .12             ;BIT0 OF TCSR WILL NOT SET
2285
2286 005160  042777  000001  175452     4$:     BIC     @BIT0,@TCSR     ;CLEAR BIT0 OF TCSR
2287 005166  032777  000001  175444             BIT     @BIT0,@TCSR     ;TEST BIT0 OF TCSR
2288 005174  001411                             BEQ     7$
2289 005176  005737  002624                     TST     CTSTFL
2290 005202  001002                             BNE     5$
2291 005204  104013                             ERROR   .13             ;BIT0 OF TCSR WILL NOT CLEAR
2292 005206  000404                             BR      7$
2293 005210                             5$:
     005210  004737  015714                     JSR     PC,@ATY4                ;;ONLY REPORT A FATAL ERROR
     005214  000013                             13                      ;;THE ERROR NUMBER (FROM APT LIST)
2294 005216  000000                     6$:     HALT
2295
2296 005220  052777  000001  175412     7$:     BIS     @BIT0,@TCSR     ;SET BIT0 IN TCSR
2297 005226  000005                             RESET                   ;CLEAR BIT0 WITH RESET
2298 005230  032777  000001  175402             BIT     @BIT0,@TCSR     ;TEST BIT0 CLEAR
2299 005236  001404                             BEQ     10$             ;** BR IF CLEAR
2300 005240  042777  000001  175372             BIC     @BIT0,@TCSR     ;CLEAR BIT0, TO PRINT ERROR
2301 005246  104014                             ERROR   .14             ;RESET DID NOT CLEAR BIT0 OF TCSR
2302 005250  000240                     10$:    NOP                     ;**
```

```
2303                                    .SBTTL  TEST # 11 - TEST THAT BIT6(XMIT INT EN) CAN BE SET & RESET
                                        ;;**********************************************************************
                                        ;*TEST 11        TEST THAT BIT6(XMIT INT EN) CAN BE SET & RESET
                                        ;;**********************************************************************
     005252  000004                   TST11:  SCOPE
2304 005254  000005                           RESET                   ;CLEAR EVERYTHING
2305 005256  017703  175366                   MOV     @TVECT,R3       ;SAVE XMIT VECTOR
2306 005262  012777  005312  175360           MOV     #1$,@TVECT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
2307 005270  004737  014502                   JSR     PC,UMPSW                ;SET PSW TO PRIORITY=7
2308 005274  000340                           .WORD   340
2309 005276  032777  000100  175334           BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
2310 005304  001404                           BEQ     2$              ;BR IF ZERO
2311 005306  104021                           ERROR   *21
2312                                                                   ;BIT6 IN TCSR NOT CLEAR AFTER RESET
2313 005310  000402                           BR      2$
2314
2315 005312  022626                   1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2316 005314  104022                           ERROR   *22
2317                                                                   ;XMIT INTERRUPT OCCURRED  PRIO=7
2318
2319 005316  052777  000100  175314   2$:     BIS     #BIT6,@TCSR     ;SET BIT6 OF TCSR
2320 005324  032777  000100  175306           BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
2321 005332  001001                           BNE     3$              ;BR. IF SET
2322
2323 005334  104023                           ERROR   *23
2324                                                                   ;CANNOT SET BIT6 OF TCSR
2325
2326 005336  042777  000100  175274   3$:     BIC     #BIT6,@TCSR     ;CLEAR BIT6 OF TCSR
2327 005344  032777  000100  175266           BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
2328 005352  001401                           BEQ     4$              ;BR IF CLEAR
2329 005354  104024                           ERROR   *24
2330                                                                   ;CANNOT CLEAR BIT6 OF TCSR
2331
2332 005356  052777  000100  175254   4$:     BIS     #BIT6,@TCSR     ;SET BIT6 OF TCSR
2333 005364  000005                           RESET                   ;CLEAR BIT6 WITH RESET
2334 005366  032777  000100  175244           BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
2335 005374  001401                           BEQ     5$              ;BR IF CLEAR
2336
2337 005376  104025                           ERROR   *25
2338                                                                   ;CANNOT CLEAR BIT6 OF TCSR WITH RESET
2339 005400  010377  175244           5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
```

```
2340                                    .SBTTL  TEST # 12 - TEST THAT BIT6 OF RCSR CAN BE SET & RESET
                                        ;;*****************************************************************
                                        ;*TEST 12        TEST THAT BIT6 OF RCSR CAN BE SET & RESET
                                        ;;*****************************************************************
     005404  000004                     TST12:  SCOPE
2341 005406  000005                             RESET                   ;CLEAR EVERYTHING
2342 005410  017703  175230                     MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
2343 005414  012777  005444  175222             MOV     #1$,@RVECT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
2344 005422  004737  014502                     JSR     PC,WRPSW        ;SET PSW TO PRIORITY=7
2345 005426  000340                             .WORD   340
2346 005430  032777  000100  175176             BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
2347 005436  001404                             BEQ     2$
2348 005440  104026                             ERROR   #26
2349                                                                    ;BIT6 OF RCSR NOT CLEAR AFTER RESET
2350 005442  000402                             BR      2$
2351
2352 005444  022626                     1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2353 005446  104027                             ERROR   #27
2354                                                                    ;RCVR INTERRUPT WITH PRIORITY=7
2355
2356 005450  052777  000100  175156     2$:     BIS     #BIT6,@RCSR     ;SET BIT6 OF RCSR
2357 005456  032777  000100  175150             BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
2358 005464  001001                             BNE     3$              ;BR, IF SET
2359
2360 005466  104030                             ERROR   #30
2361                                                                    ;CANNOT SET BIT6 OF RCSR
2362
2363 005470  042777  000100  175136     3$:     BIC     #BIT6,@RCSR     ;CLEAR BIT6 OF RCSR
2364 005476  032777  000100  175130             BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
2365 005504  001401                             BEQ     4$              ;BR, IF CLEAR
2366
2367 005506  104031                             ERROR   #31
2368                                                                    ;CANNOT CLEAR BIT6 OF RCSR
2369
2370 005510  052777  000100  175116     4$:     BIS     #BIT6,@RCSR     ;SET BIT6 OF RCSR
2371 005516  000005                             RESET                   ;CLEAR BIT6OF RCSR WITH RESET
2372 005520  032777  000100  175106             BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
2373 005526  001401                             BEQ     5$              ;BR, IF CLEAR
2374
2375 005530  104032                             ERROR   #32
2376                                                                    ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
2377 005532  010377  175106             5$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
2378 005536  012737  000012  001002     TCLOCK: MOV     #12,#TSTNM      ;ADJUST TEST NUMBER TO (NEXT TEST - 1)
```

```
 23/9                                   .SBTTL  TEST # 13 - TEST ABILITY TO ACCESS LKS
                                        ;;*****************************************************************
                                        ;*TEST 13       TEST ABILITY TO ACCESS LKS
                                        ;;*****************************************************************
     005544  000004                     TST13:  SCOPE
2380 005546  005737  002624                     TST     CTSTFL          ;IS CONSOLE UNDER TEST?
     005552  001420                             BEQ     TST14           ;IF NOT, SKIP THIS TEST
     005554  032777  000100  173256             BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
     005562  001014                             BNE     TST14           ;IF YES, SKIP THIS TEST
2381 005564  013703  000004                     MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
2382 005570  012737  005604  000004             MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
2383 005576  005777  175072                     TST     @LKS            ;ACCESS LKS
2384 005602  000402                             BR      2$              ;NO TIMEOUT -  BR TO END OF TEST
2385
2386 005604  022626                     1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
2387 005606  104010                             ERROR   +10             ;CAN NOT ACCESS LKS
2388
2389 005610  010337  000004             2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
2390
```

```
2391                                    .SBTTL  TEST # 14 - TEST THAT BIT6 OF LKS CAN BE SET & RESET
                                        ;;*****************************************************************
                                        ;*TEST 14        TEST THAT BIT6 OF LKS CAN BE SET & RESET
                                        ;;*****************************************************************
     005614  000004                     TST14:  SCOPE
2392 005616  005737  002624                     TST     CTSTFL          ;IS CONSOLE UNDER TEST?
     005622  001460                             BEQ     TST15           ;IF NOT, SKIP THIS TEST
     005624  032777  000100  173206             BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
     005632  001054                             BNE     TST15           ;IF YES, SKIP THIS TEST
2393 005634  000005                             RESET
2394 005636  017703  175034                     MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
2395 005642  012777  005672  175026             MOV     #1$,@RTCVT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
2396 005650  004737  014502                     JSR     PC,@#RPSW       ;SET PSW TO PRIORITY 7
2397 005654  000340                             .WORD   340
2398 005656  032777  000100  175010             BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
2399 005664  001404                             BEQ     2$
2400 005666  104033                             ERROR   +33
2401                                                                    ;BIT6 OF LKS NOT CLEAR AFTER RESET
2402 005670  000402                             BR      2$
2403
2404 005672  022626                     1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2405 005674  104034                             ERROR   +34
2406                                                                    ;LKS INTERRUPT WITH PRIORITY=7
2407
2408 005676  052777  000100  174770     2$:     BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
2409 005704  032777  000100  174762             BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
2410 005712  001001                             BNE     3$              ;BR IF SET
2411
2412 005714  104035                             ERROR   +35
2413                                                                    ;CANNOT SET BIT6 OF LKS
2414
2415 005716  042777  000100  174750     3$:     BIC     #BIT6,@LKS      ;CLEAR BIT6 OF LKS
2416 005724  032777  000100  174742             BIT     #BIT6,@LKS      ;TEST BIT6 OF LK
2417 005732  001401                             BEQ     4$
2418 005734  104036                             ERROR   +36
2419                                                                    ;CANNOT CLEAR BIT6 OF LKS
2420 005736  052777  000100  174730     4$:     BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
2421 005744  000005                             RESET                   ;CLEAR BIT6 OF LKS WITH RESET
2422 005746  032777  000100  174720             BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
2423 005754  001401                             BEQ     5$              ;BR IF CLEAR
2424
2425 005756  104037                             ERROR   +37
2426                                                                    ;CANNOT CLEAR BIT6 OF LKS WITH RESET
2427 005760  010377  174712             5$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
```

```
 2428                                           .SBTTL  TEST # 15 - TEST FOR DUAL ADDRESSING OF REGISTERS
                                                ;;*****************************************************************
                                                ;*TEST 15       TEST FOR DUAL ADDRESSING OF REGISTERS
                                                ;;*****************************************************************
      005764  000004                            TST15:  SCOPE
 2429 005766  013703  000004                            MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
 2430 005772  013704  000006                            MOV     @#6,R4          ;SAVE TIMEOUT PSW
 2431 005776  012737  006130  000004                    MOV     #4$,@#4         ;SET UP TIMEOUT VECTOR
 2432 006004  012737  000340  000006                    MOV     #340,@#6        ;KEEP PRIO=7
 2433 006012  000005                                    RESET                   ;CLEAR EVERYTHING
 2434 006014  012700  000002                            MOV     #BIT1,R0        ;SET UP BIT MASK
 2435 006020  032777  000020  173012                    BIT     #BIT4,@#SWR     ;CLOCK TEST ONLY?
 2436 006026  001404                                    BEQ     1$              ;BR IF NOT
 2437 006030  013737  002674  001020                    MOV     LKS,$GDADR      ;ELSE, MOVE GOOD LKS ADDRESS INTO $GDADR
 2438 006036  000403                                    BR      2$
 2439 006040  013737  002634  001020            1$:     MOV     RCSR,$GDADR     ;MOVE GOOD RCSR ADDRESS INTO $GDADR
 2440 006046  013737  001020  001022            2$:     MOV     $GDADR,$BDADR   ;MOVE GOOD ADDRESS INTO TEST ADDRESS LOCATION
 2441 006054  040037  001022                            BIC     R0,$BDADR       ;CREATE BAD ADDRESS BY COMPLEMENTING ONE BIT
 2442 006060  023737  001020  001022                    CMP     $GDADR,$BDADR   ;ARE ADDRESSES IDENTICAL?
 2443 006066  001002                                    BNE     3$              ;IF NOT, TEST THIS ADDRESS
 2444 006070  050037  001022                            BIS     R0,$BDADR       ;ELSE, BIT SET THIS BIT POSITION TO GENERATE BAD ADDRESS
 2445 006074  017737  172722  001024            3$:     MOV     @$BDADR,$GDDAT  ;SAVE CONTENTS OF BAD ADDRESS IF IT EXISTS
 2446 006102  052777  000100  172712                    BIS     #BIT6,@$BDADR   ;SET BIT6 USING BAD ADDRESS
 2447 006110  032777  000100  172702                    BIT     #BIT6,@$GDADR   ;CHECK TO SEE IF GOOD ADDRESS CONTAINS BIT6
 2448 006116  001011                                    BNE     6$              ;BR IF SET ---> ERROR
 2449 006120  013777  001024  172674                    MOV     $GDDAT,@$BDADR  ;RESTORE ANY MEMORY LOCATION THAT WAS ALTERED
 2450 006126  000401                                    BR      5$              ;BR TO CONTINUE TEST
 2451 006130  022626                            4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
 2452 006132  006300                            5$:     ASL     R0              ;SHIFT BIT MASK TO NEXT POSITION
 2453 006134  105700                                    TSTB    R0              ;COMPLEMENTED ALL BITS FROM 1 - 7?                      ;;**
C
 2454 006136  100343                                    BPL     2$              ;BR, IF NOT.                                            ;;**
C
 2455 006140  000401                                    BR      7$              ;BR TO NEXT TEST
 2456
 2457 006142  104040                            6$:     ERROR   +40             ;DUAL ADDRESSING ERROR
 2458                                                                           ;$BDADR = DUAL ADDRESS
 2459                                                                           ;$GDADR = GOOD ADDRESS
 2460
 2461 006144  010337  000004                    7$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
 2462 006150  010437  000006                            MOV     R4,@#6          ;RESTORE TIMEOUT PSW
```

TEST # 16 - TEST THAT BIT7 OF LKS SETS & CAN BE CLEARED

```
2463                                    .SBTTL  TEST # 16 - TEST THAT BIT7 OF LKS SETS & CAN BE CLEARED
                                        ;;**********************************************************************
                                        ;*TEST 16        TEST THAT BIT7 OF LKS SETS & CAN BE CLEARED
                                        ;;**********************************************************************
     006154  000004                     TST16:  SCOPE
2464 006156  005737  002624                     TST     CTSTFL          ;IS CONSOLE UNDER TEST?
     006162  001437                     TST17:  BEQ     TST17           ;IF NOT, SKIP THIS TEST
     006164  032777  000100  172646             BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
     006172  001033                             BNE     TST17           ;IF YES, SKIP THIS TEST
2465 006174  000005                             RESET                   ;CLEAR EVERYTHING & SET BIT7 OF LKS
2466 006176  105777  174472             1$:     TSTB    @LKS            ;TEST FOR BIT7 OF LKS
2467 006202  100401                             BMI     2$              ;BR IF SET
2468
2469 006204  104041                             ERROR   .41             ;BIT7 OF LKS DID NOT SET WITH RESET
2470
2471 006206  042777  000200  174460  2$:       BIC     #BIT7,@LKS      ;CLEAR BIT7 OF LKS
2472 006214  032777  000200  174452             BIT     #BIT7,@LKS      ;TEST BIT7 OF LKS
2473 006222  001410                             BEQ     3$
2474 006224  042777  000200  174442             BIC     #BIT7,@LKS      ;TRY ONE MORE TIME BECAUSE THE CLOCK
2475 006232  032777  000200  174434             BIT     #BIT7,@LKS      ;  MAY HAVE SET IMMEDIATELY AFTER THE FIRST CLEAR
2476 006240  001401                             BEQ     3$
2477
2478 006242  104042                             ERROR   .42             ;CAN NOT CLEAR BIT7 OF LKS
2479
2480 006244  005000                     3$:     CLR     R0              ;CLEAR TIMER
2481 006246  105777  174422             CONT:   TSTB    @LKS            ;TEST FOR BIT7 OF LKS
2482 006252  100403                             BMI     TST17           ;BR, IF SET
2483 006254  005200                             INC     R0              ;INCREMENT TIMER
2484 006256  001373                             BNE     CONT            ;CONTINUE UNTIL TIME EXPIRES
2485
2486 006260  104043                             ERROR   .43             ;BIT7 OF LKS DOES NOT SET
```

```
2487                                    .SBTTL  TEST # 17 - TEST THAT THE REAL TIME CLOCK INTERRUPTS PROPERLY
                                        ;;*****************************************************************
                                        ;*TEST 17       TEST THAT THE REAL TIME CLOCK INTERRUPTS PROPERLY
                                        ;;*****************************************************************
     006262  000004                     TST17:  SCOPE
2488 006264  005737  002624                     TST     CTSTFL          ;IS CONSOLE UNDER TEST?
     006270  001514                             BEQ     TST20           ;IF NOT, SKIP THIS TEST
     006272  032777  000100  172540             BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
     006300  001110                             BNE     TST20           ;IF YES, SKIP THIS TEST
2489 006302  004737  014502                     JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2490 006306  000340                                     .WORD   340
2491 006310  017703  174362                     MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
2492 006314  017704  174360                     MOV     @RTCPSW,R4      ;SAVE LINE CLOCK PSW VECTOR
2493 006320  012777  006400  174350             MOV     #3$,@RTCVT      ;SET RTC INTERRUPT VECTOR TO ERROR REPORT
2494 006326  012777  000340  174344             MOV     #340,@RTCPSW    ;KEEP PRIORITY AT 7
2495 006334  005077  174334                     CLR     @LKS            ;CLEAR DONE FLAG AND
2496 006340  012777  000100  174326             MOV     #BIT6,@LKS      ;SET CLOCK INTERRUPT ENABLE
2497 006346  105777  174322             1$:     TSTB    @LKS            ;WAIT FOR RTC DONE(INTERRUPT REQUEST)
2498 006352  100375                             BPL     1$
2499 006354  005077  174314                     CLR     @LKS            ;CLEAR DONE FLAG AND
2500 006360  012777  000100  174306             MOV     #BIT6,@LKS      ;SET CLOCK INTERRUPT ENABLE
2501 006366  105777  174302             2$:     TSTB    @LKS            ;WAIT FOR RTC DONE(INTERRUPT REQUEST)
2502 006372  100375                             BPL     2$
2503 006374  000240                             NOP                     ;GIVE TIME FOR ANY INTERRUPTS
2504 006376  000402                             BR      4$              ;BR, IF NO INTERRUPT OCCURS
2505
2506 006400  022626             3$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2507 006402  104044                     ERROR   +44             ;RTC INTERRUPTS AT PRIORITY 7
2508
2509 006404  005077  174264             4$:     CLR     @LKS            ;DISABLE RTC INTERRUPTS & CLEAR DONE
2510 006410  012777  006440  174260             MOV     #5$,@RTCVT      ;SET RTC INTERRUPT VECTOR FOR ERROR
2511 006416  004737  014502                     JSR     PC,WRPSW        ;CHANGE PSW TO PRIORITY 5
2512 006422  000240                                     .WORD   240
2513 006424  105777  174244             20$:    TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
2514 006430  100375                             BPL     20$
2515 006432  000240                             NOP                     ;GIVE TIME FOR ANY INTERRUPT
2516 006434  000240                             NOP                     ;GIVE TIME FOR ANY INTERRUPT
2517 006436  000402                             BR      6$              ;IF NO INTERRUPT - BR TO CONTINUE TEST
2518
2519 006440  022626             5$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2520 006442  104045                     ERROR   +45             ;RTC INTERRUPTS WITH INTERRUPTS DISABLED
2521
2522 006444  012777  006502  174224     6$:     MOV     #8$,@RTCVT      ;POINT RTC VECTOR TO END OF TEST
2523 006452  042777  000200  174214             BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
2524 006460  052777  000100  174206             BIS     #BIT6,@LKS      ;ALLOW INTERRUPTS
2525 006466  105777  174202             7$:     TSTB    @LKS            ;WAIT FOR RTC DONE
2526 006472  100375                             BPL     7$
2527 006474  000240                             NOP                     ;GIVE TIME FOR INTERRUPT
2528
2529 006476  104046                             ERROR   +46             ;RTC INTERRUPT DID NOT OCCUR
2530 006500  000401                             BR      9$              ;BRANCH OVER STACK CORRECTION
2531 006502  022626             8$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2532 006504  042777  000100  174162     9$:     BIC     #BIT6,@LKS      ;DISABLE INTERRUPTS
2533 006512  010377  174160                     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
2534 006516  010477  174156                     MOV     R4,@RTCPSW      ;RESTORE LINE CLOCK PSW VECTOR
```

```
2535                                    .SBTTL  TEST # 20 - TEST RTC FOR DOUBLE INTERRUPTS
                                        ;;**********************************************************************
                                        ;*TEST #20       TEST RTC FOR DOUBLE INTERRUPTS
                                        ;;**********************************************************************
     006522  000004                     TST20:  SCOPE
2536 006524  005737  002624                     TST     CTSTFL          ;IS CONSOLE UNDER TEST?
     006530  001473                     BEQ     TST21           ;IF NOT, SKIP THIS TEST
     006532  032777  000100  172300     BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
     006540  001067                     BNE     TST21           ;IF YES, SKIP THIS TEST
2537 006542  000005                     RESET                   ;CLEAR EVERYTHING
2538 006544  017703  174126             MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
2539 006550  017704  174124             MOV     @RTCPSW,R4      ;SAVE LINE CLOCK PSW VECTOR
2540 006554  012777  006646  174114     MOV     #3$,@RTCVT      ;SET UP RTC INTERRUPT VECTOR
2541 006562  012777  000340  174110     MOV     #340,@RTCPSW    ;DISALLOW INTERRUPTS AFTER THE INTERRUPT
2542 006570  004737  014502             JSR     PC,WRPSW        ;SET PRIORITY TO 5
2543 006574  000240                     .WORD   240
2544 006576  005077  174072             CLR     @LKS            ;CLEAR DONE FLAG AND
2545 006602  012777  000100  174064     MOV     #BIT6,@LKS      ;SET CLOCK INTERRUPT ENABLE
2546 006610  105777  174060     1$:     TSTB    @LKS            ;WAIT FOR DONE
2547 006614  100375                     BPL     1$              ;BRANCH BACK IF NOT DONE
2548 006616  005077  174052             CLR     @LKS            ;CLEAR DONE FLAG AND
2549 006622  012777  000100  174044     MOV     #BIT6,@LKS      ;SET CLOCK INTERRUPT ENABLE
2550 006630  105777  174040     2$:     TSTB    @LKS            ;WAIT FOR DONE
2551 006634  100375                     BPL     2$              ;BRANCH BACK IF NOT DONE
2552 006636  000240                     NOP                     ;GIVE TIME FOR ANY INTERRUPT
2553 006640  000240                     NOP                     ;GIVE TIME FOR ANY INTERRUPT
2554 006642  104047                     ERROR   +47             ;RTC INTERRUPT DID NOT OCCUR
2555 006644  000401                     BR      4$              ;BRANCH OVER STACK CORRECTION
2556 006646  022626             3$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2557 006650  012777  006676  174020     4$:     MOV     #5$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
2558 006656  004737  014502             JSR     PC,WRPSW        ;SET PSW TO PRIORITY 5
2559 006662  000240                     .WORD   240
2560 006664  000240                     NOP                     ;GIVE SOME TIME FOR AN INTERRUPT
2561 006666  000240                     NOP                     ;GIVE SOME TIME FOR AN INTERRUPT
2562 006670  000240                     NOP                     ;GIVE SOME TIME FOR AN INTERRUPT
2563 006672  000240                     NOP                     ;GIVE SOME TIME FOR AN INTERRUPT
2564 006674  000402                     BR      6$              ;NO INTERRUPT - BR TO END OF TEST
2565
2566 006676  022626             5$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2567 006700  104050                     ERROR   +50             ;INTERRUPT SEQUENCE DID NOT CLEAR
2568                                                             ;INTERRUPT REQUEST
2569
2570 006702  042777  000100  173764     6$:     BIC     #BIT6,@LKS      ;DISABLE CLOCK INTERRUPTS
2571 006710  010377  173762             MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
2572 006714  010477  173760             MOV     R4,@RTCPSW      ;RESTORE LINE CLOCK PSW VECTOR
```

```
2573                                    .SBTTL  TEST # 21 - TEST THAT RTC INTERRUPT CLEARS WITH RESET
                                        ;;*********************************************************************
                                        ;*TEST 21       TEST THAT RTC INTERRUPT CLEARS WITH RESET
                                        ;;*********************************************************************
     006720  000004                     TST21:  SCOPE
2574 006722  005737  002624                     TST     CTSTFL          ;IS CONSOLE UNDER TEST?
     006726  001452                             BEQ     TST22           ;IF NOT, SKIP THIS TEST
     006730  032777  000100  172102             BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
     006736  001046                             BNE     TST22           ;IF YES, SKIP THIS TEST
2575 006740  004737  014502                     JSR     PC,WRPSW        ;SET PRIORITY TO 7
2576 006744  000340                             .WORD   340
2577 006746  017703  173724                     MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
2578 006752  012777  007044  173716             MOV     #3$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
2579 006760  005077  173710                     CLR     @LKS            ;CLEAR DONE FLAG AND
2580 006764  012777  000100  173702             MOV     #BIT6,@LKS      ;SET CLOCK INTERRUPT ENABLE
2581 006772  105777  173676             1$:     TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
2582 006776  100375                             BPL     1$
2583 007000  005077  173670                     CLR     @LKS            ;CLEAR DONE FLAG AND
2584 007004  012777  000100  173662             MOV     #BIT6,@LKS      ;SET CLOCK INTERRUPT ENABLE
2585 007012  105777  173656             2$:     TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
2586 007016  100375                             BPL     2$
2587 007020  000240                             NOP                     ;GIVE TIME FOR ANY INTERRUPT
2588 007022  000005                             RESET                   ;CLEAR PENDING INTERRUPT WITH RESET
2589 007024  004737  014502                     JSR     PC,WRPSW        ;SET PRIORITY TO 5
2590 007030  000240                             .WORD   240
2591 007032  000240                             NOP                     ;GIVE TIME FOR ANY INTERRUPT
2592 007034  042777  000100  173632             BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
2593 007042  000402                             BR      4$              ;BR TO END OF TEST
2594
2595 007044  022626             3$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2596 007046  104051                     ERROR   +51             ;RESET DID NOT CLEAR INTERRUPT
2597
2598 007050  010377  173622             4$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
```

```
2599                                    .SBTTL  TEST # 22 - TEST THAT RTC INTERRUPT CLEARS BY CLEARING BIT7 OF LKS
                                        ;;**********************************************************************
                                        ;*TEST 22       TEST THAT RTC INTERRUPT CLEARS BY CLEARING BIT7 OF LKS
                                        ;;**********************************************************************
 007054  000004                 TST22:  SCOPE
2600 007056  005737  002624             TST     CTSTFL          ;IS CONSOLE UNDER TEST?
     007062  001457                     BEQ     TST23           ;IF NOT, SKIP THIS TEST
     007064  032777  000100  171746     BIT     #BIT6,@SHR      ;ARE LINE CLOCK TESTS INHIBITED?
     007072  001053                     BNE     TST23           ;IF YES, SKIP THIS TEST
2601 007074  004737  014502             JSR     PC,WRPSW        ;SET PRIORITY TO 7
2602 007100  000340                     .WORD   340
2603 007102  017703  173570             MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
2604 007106  012777  007204  173562     MOV     #3$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
2605 007114  005077  173554             CLR     @LKS            ;CLEAR DONE FLAG AND
2606 007120  012777  000100  173546     MOV     #BIT6,@LKS      ;SET CLOCK INTERRUPT ENABLE
2607 007126  105777  173542     1$:     TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
2608 007132  100375                     BPL     1$
2609 007134  005077  173534             CLR     @LKS            ;CLEAR DONE FLAG AND
2610 007140  012777  000100  173526     MOV     #BIT6,@LKS      ;SET CLOCK INTERRUPT ENABLE
2611 007146  105777  173522     2$:     TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
2612 007152  100375                     BPL     2$
2613 007154  000240                     NOP                     ;GIVE TIME FOR ANY INTERRUPT
2614 007156  042777  000200  173510     BIC     #BIT7,@LKS      ;CLEAR DONE & INTERRUPT
2615 007164  004737  014502             JSR     PC,WRPSW        ;ALLOW INTERRUPTS
2616 007170  000240                     .WORD   240
2617 007172  000240                     NOP                     ;GIVE TIME FOR ANY INTERRUPT
2618 007174  042777  000100  173472     BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
2619 007202  000402                     BR      4$              ;BR TO END OF TEST
2620
2621
2622 007204  022626             3$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2623 007206  104052                     ERROR   +52             ;CLEARING BIT7 OF LKS DID NOT CLEAR INTERRUPT
2624
2625 007210  010377  173462     4$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
2626 007214  004737  014502             JSR     PC,WRPSW        ;RESTORE PRIORITY TO 7
2627 007220  000340                     .WORD   340
```

```
2628                                        .SBTTL  TEST # 23 - TEST CLOCK REPEATABILITY
                                            ;;*****************************************************************
                                            ;*TEST 23       TEST CLOCK REPEATABILITY
                                            ;;*****************************************************************
     007222  000004                   TST23:  SCOPE
2629 007224  005737  002624                   TST     CTSTFL          ;IS CONSOLE UNDER TEST?
     007230  001467                           BEQ     TST24           ;IF NOT, SKIP THIS TEST
     007232  032777  000100  171600           BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
     007240  001063                           BNE     TST24           ;IF YES, SKIP THIS TEST
2630 007242  042777  000100  173424           BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
2631
2632 007250  005000                           CLR     R0              ;CLEAR A TIMER
2633 007252  012701  177777                   MOV     #-1,R1          ;SET A FLAG INDICATING FIRST PASS THRU THIS LOOP
2634 007256  005002                   1$:     CLR     R2              ;CLEAR CLOCK COUNTER
2635 007260  005077  173410                   CLR     @LKS            ;CLEAR DONE
2636 007264  105777  173404           2$:     TSTB    @LKS            ;SYNC ON DONE
2637 007270  100375                           BPL     2$
2638 007272  005077  173376                   CLR     @LKS            ;CLEAR DONE
2639 007276  105777  173372           3$:     TSTB    @LKS            ;IS CLOCK DONE?
2640 007302  100003                           BPL     4$              ;BR IF NOT , TO INCREMENT TIMER
2641 007304  005202                           INC     R2              ;IF DONE, INCREMENT CLOCK COUNT
2642 007306  005077  173362                   CLR     @LKS            ;CLEAR DONE
2643 007312  005200                   4$:     INC     R0              ;INCREMENT TIMER
2644 007314  001370                           BNE     3$              ;BR IF TIME REMAINS
2645 007316  005201                           INC     R1              ;INCREMENT LOOP PASS FLAG
2646 007320  001003                           BNE     CMPARE          ;BR IF TWO PASSES HAVE BEEN MADE
2647 007322  010237  002336                   MOV     R2,FIRST        ;IF NOT, STORE FIRST CLOCK COUNT
2648 007326  000753                           BR      1$              ;DO LOOP AGAIN
2649 007330  013701  002336           CMPARE: MOV     FIRST,R1        ;RECALL FIRST CLOCK COUNT
2650 007334  160201                           SUB     R2,R1           ;CALCULATE DIFFERENCE OF TWO COUNTS
2651 007336  100001                           BPL     TOLER           ;IF POSITIVE,SKIP NEGATION OF DIFFERENCE
2652 007340  005401                           NEG     R1              ;MAKE DIFFERENCE A POSITIVE NUMBER
2653 007342  032737  000001  003002   TOLER:  BIT #@BIT0,FLAG44       ;** IS THIS A 11/44
2654 007350  001403                           BEQ 6$                  ;** NO
2655 007352  020127  000002                   CMP     R1,#2           ;** YES; COMPARE DIFFERENCE WITH DESIRED
2656                                                                   ;** TOLERANCE OF 2
2657 007356  000402                           BR 7$                   ;**
2658 007360  020127  000001           6$:     CMP     R1,#1           ;COMPARE DIFFERENCE WITH DESIRED TOLERANCE
2659 007364  003403                   7$:     BLE     5$              ;BR, IF LOWER/EQUAL TO TOLERANCE
2660
2661 007366  010237  002616                   MOV     R2,SECND        ;STORE SECOND COUNT
2662 007372  104053                           ERROR   #53             ;CLOCK REPEATABILITY ERROR
2663
2664 007374  032777  000020  171436   5$:     BIT     #BIT4,@SWR      ;CLOCK TESTS ONLY?
2665 007402  001402                           BEQ     TST24           ;BR IF NOT
2666 007404  000137  014326                   JMP     @EOP            ;ELSE, JUMP TO END OF PASS ROUTINE
```

```
 2667                                   .SBTTL  TEST # 24 - TEST THAT XMIT INTERRUPTS ONLY WHEN ENABLED
                                        ;;**************************************************************
                                        ;*TEST 24        TEST THAT XMIT INTERRUPTS ONLY WHEN ENABLED
                                        ;;**************************************************************
      007410  000004                    TST24:  SCOPE
 2668 007412  042777  000100  173220            BIC     #BIT6,@TCSR     ;CLEAR TRANSMIT INTERRUPT ENABLE
 2669 007420  017703  173224                    MOV     @TVECT,R3       ;SAVE XMIT VECTOR
 2670 007424  012777  007450  173216            MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
 2671 007432  105777  173202            1$:     TSTB    @TCSR           ;WAIT FOR DONE
 2672 007436  100375                            BPL     1$
 2673 007440  004737  014502                    JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
 2674 007444  000140                            .WORD   140
 2675 007446  000402                            BR      3$
 2676
 2677 007450  022626                    2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
 2678 007452  104054                            ERROR   +54
 2679                                                                   ;XMIT INTERRUPTS WITH INTERRUPT ENABLE CLEAR
 2680 007454  012777  007504  173166    3$:     MOV     #4$,@TVECT      ;SET XMIT VECTOR TO END OF TEST
 2681 007462  052777  000100  173150            BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
 2682 007470  000240                            NOP                     ;ALLOW
 2683 007472  000240                            NOP                     ;TIME
 2684 007474  000240                            NOP                     ;FOR
 2685 007476  000240                            NOP                     ;INTERRUPT
 2686 007500  104055                            ERROR   +55             ;XMIT DID NOT INTERRUPT
 2687 007502  000401                            BR      5$              ;BRANCH OVER STACK CORRECTION
 2688 007504  022626                    4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
 2689 007506  042777  000100  173124    5$:     BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
 2690 007514  010377  173130                    MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
```

```
2691                                            .SBTTL  TEST # 25 - TEST THAT XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
                                                ;;**********************************************************************
                                                ;*TEST 25        TEST THAT XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
                                                ;;**********************************************************************
     007520  000004                             TST25:  SCOPE
2692 007522  042777  000100  173110                     BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
2693 007530  004737  014502                             JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2694 007534  000340                                     .WORD   340
2695 007536  017703  173106                             MOV     @TVECT,R3       ;SAVE XMIT VECTOR
2696 007542  012777  007576  173100                     MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
2697 007550  105777  173064             1$:     TSTB    @TCSR           ;WAIT FOR DONE
2698 007554  100375                             BPL     1$
2699 007556  052777  000100  173054                     BIS     #BIT6,@TCSR     ;ENABLE INTERRUPT
2700 007564  000240                             NOP                     ;ALLOW
2701 007566  000240                             NOP                     ;TIME
2702 007570  000240                             NOP                     ;FOR
2703 007572  000240                             NOP                     ;INTERRUPT
2704 007574  000402                             BR      3$              ;CONTINUE TEST
2705
2706 007576  022626                     2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2707 007600  104056                             ERROR   #56
2708                                                                     ;XMIT INTERRUPTS AT PRIORITY=7
2709 007602  042777  000100  173030     3$:     BIC     #BIT6,@TCSR     ;CLEAR INTERRUPT ENABLE
2710 007610  012777  007636  173032             MOV     #4$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
2711 007616  004737  014502                     JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2712 007622  000140                             .WORD   140
2713 007624  000240                             NOP                     ;ALLOW
2714 007626  000240                             NOP                     ;TIME
2715 007630  000240                             NOP                     ;FOR
2716 007632  000240                             NOP                     ;INTERRUPT
2717 007634  000402                             BR      5$              ;BR TO END OF TEST-NO INTERRUPT
2718
2719 007636  022626                     4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2720 007640  104057                             ERROR   #57
2721                                                                     ;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
2722 007642  010377  173002             5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
```

```
2723                                    .SBTTL  TEST # 26 - TEST TRANSMITTER FOR DOUBLE INTERRUPTS
                                        ;;*****************************************************************
                                        ;*TEST 26        TEST TRANSMITTER FOR DOUBLE INTERRUPTS
                                        ;;*****************************************************************
     007646  000004                     TST26:  SCOPE
2724 007650  042777  000100  172762             BIC     #BIT6,@TCSR     ;CLEAR INTERRUPT ENABLE
2725 007656  017703  172766             MOV     @TVECT,R3       ;SAVE XMIT VECTOR
2726 007662  017704  172764             MOV     @TPSW,R4        ;SAVE XMIT PSW VECTOR
2727 007666  012777  007740  172754     MOV     #2$,@TVECT      ;SET UP XMIT VECTOR
2728 007674  012777  000340  172750     MOV     #340,@TPSW      ;SET PIO 7 AFTER INTERRUPT
2729 007702  004737  014502             JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2730 007706  000140                             .WORD   140
2731 007710  105777  172724             1$:     TSTB    @TCSR           ;WAIT FOR DONE
2732 007714  100375                             BPL     1$
2733 007716  052777  000100  172714             BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
2734 007724  000240                             NOP                     ;ALLOW
2735 007726  000240                             NOP                     ;TIME
2736 007730  000240                             NOP                     ;FOR
2737 007732  000240                             NOP                     ;INTERRUPT
2738
2739 007734  104060                             ERROR   .60             ;XMIT INTERRUPT DID NOT OCCUR
2740 007736  000401                             BR      25$             ;BRANCH OVER STACK CORRECTION INTERRUPT
2741 007740  022626                     2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2742 007742  012777  007776  172700     25$:    MOV     #4$,@TVECT      ;POINT XMIT VECTOR TO ERROR
2743 007750  004737  014502             JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2744 007754  000140                             .WORD   140
2745 007756  000240                             NOP                     ;ALLOW
2746 007760  000240                             NOP                     ;TIME
2747 007762  000240                             NOP                     ;FOR
2748 007764  000240                             NOP                     ;INTERRUPT
2749 007766  042777  000100  172644             BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
2750 007774  000402                             BR      5$              ;BR TO END OF TEST
2751
2752 007776  022626                     4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2753 010000  104061                             ERROR   .61
2754                                                                     ;XMIT RE-INTERRUPTED
2755 010002  010377  172642             5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
2756 010006  010477  172640             MOV     R4,@TPSW        ;RESTORE XMIT PSW VECTOR
2757
```

```
2758                                    .SBTTL  TEST # 27 - TEST THAT XMIT INTERRUPT CLEARS WITH LOADING TBUF
                                        ;;*******************************************************************
                                        ;*TEST 27       TEST THAT XMIT INTERRUPT CLEARS WITH LOADING TBUF
                                        ;;*******************************************************************
     010012  000004                     TST27:  SCOPE
2759 010014  042777  000100  172616             BIC     @#BIT6,@TCSR    ;DISABLE INTERRUPTS
2760 010022  004737  014502                     JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
2761 010026  000340                             .WORD   340
2762 010030  017703  172614                     MOV     @TVECT,R3       ;SAVE XMIT VECTOR
2763 010034  012777  010122  172606             MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR
2764 010042  052777  000004  172570             BIS     @#BIT2,@TCSR    ;** ENABLE MAINT. WRAP
2765 010050  052777  000100  172562             BIS     @#BIT6,@TCSR    ;ENABLE INTERRUPTS
2766 010056  005077  172560                     CLR     @TBUF           ;LOAD TBUF
2767 010062  105777  172552             1$:     TSTB    @TCSR           ;WAIT FOR DONE (INTERRUPT)
2768 010066  100375                             BPL     1$
2769 010070  005077  172546                     CLR     @TBUF           ;FILL SECOND BUFFER TO RESET INT.
2770 010074  004737  014502                     JSR     PC,WRPSW        ;ALLOW INTERRUPTS
2771 010100  000140                             .WORD   140
2772 010102  000240                             NOP                     ;ALLOW
2773 010104  000240                             NOP                     ;TIME
2774 010106  000240                             NOP                     ;FOR
2775 010110  000240                             NOP                     ;INTERRUPT
2776 010112  042777  000100  172520             BIC     @#BIT6,@TCSR    ;DISABLE INTERRUPTS
2777 010120  000405                             BR      3$              ;BR TO END OF TEST
2778
2779 010122  022626                     2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2780 010124  042777  000004  172526             BIC     @#BIT2,@CTCSR   ;** DISABLE MAINTENANCE MODE FOR
                                                                        ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                        ;** WITH TERMINAL.
2781 010132  104062                             ERROR   #62
2782                                                                    ;LOADING TBUF DID NOT CLEAR INTERRUPT.
2783 010134  010377  172510             3$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
2784
2785 010140  005000                             CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     010142  012701  000002                     MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
     010146  005300                     40$:    DEC     R0              ;** TRANSMITTED TO FINISH BEFORE MAINTENANCE
     010150  001376                             BNE     40$             ;** WRAP FOR THE UART UNDER TEST IS DISABLED.
     010152  005301                             DEC     R1              ;** THIS WILL INHIBIT ANY COMMUNICATION
     010154  001374                             BNE     40$             ;** TO HARDWARE MEDIA SUCH AS TU58 THAT MIGHT
                                                                        ;** BE ATTACHED TO UART UNDER TEST.
```

```
2786                                    .SBTTL  TEST # 30 - TEST THAT RCVR ACTIVE & DONE SET & CLEAR
                                        ;;**********************************************************
                                        ;*TEST 30       TEST THAT RCVR ACTIVE & DONE SET & CLEAR
                                        ;;**********************************************************
     010156  000004                     TST30:  SCOPE
2787 010160  032737  000001  003002             BIT     @#BIT0,FLAG44   ;**
2788 010166  001067                             BNE     RCVDON          ;**
2789 010170  000005                             RESET                   ;CLEAR EVERYTHING
2790 010172  052777  000004  172440             BIS     @#BIT2,@TCSR    ;SET MAINTENANCE WRAP
2791 010200  005000                             CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     010202  012701  000002                     MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                         ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                         ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     010206  105777  172424             42$:    TSTB    @RBUF           ;** READ TO CLEAR DONE
     010212  005300                             DEC     R0              ;**
     010214  001374                             BNE     42$             ;**
     010216  005301                             DEC     R1              ;**
     010220  001372                             BNE     42$             ;**
2792 010222  005000                             CLR     R0              ;CLEAR A TIMER
2793 010224  005077  172412                     CLR     @TBUF           ;LOAD TRANSMIT BUFFER
2794 010230  032777  004000  172376     WACTV:  BIT     @#BIT11,@RCSR   ;TEST RCVR ACTIVE BIT
2795 010236  001006                             BNE     2$              ;BR IF SET
2796 010240  005200                             INC     R0              ;INCREMENT TIMER IF NOT SET
2797 010242  001372                             BNE     WACTV           ;CONTINUE WAIT IF TIME REMAINS
2798 010244  042777  000004  172406             BIC     @#BIT2,@TCSR    ;** DISABLE MAINTENANCE MODE FOR
                                                                         ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                         ;** WITH TERMINAL.
2799
2800 010252  104063                             ERROR   .63             ;RCVR ACTIVE DID NOT SET WHILE RECEIVING
2801
2802 010254                             2$:
     010254  005000                             CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     010256  012701  000002                     MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
     010262  005300                     40$:    DEC     R0              ;** TRANSMITTED TO FINISH BEFORE MAINTENANCE
     010264  001376                             BNE     40$             ;** WRAP FOR THE UART UNDER TEST IS DISABLED.
     010266  005301                             DEC     R1              ;** THIS WILL INHIBIT ANY COMMUNICATION
     010270  001374                             BNE     40$             ;** TO HARDWARE MEDIA SUCH AS TU58 THAT MIGHT
                                                                         ;** BE ATTACHED TO UART UNDER TEST.
2803 010272  000005                             RESET
2804 010274  032777  004000  172332             BIT     @#BIT11,@RCSR   ;VERIFY "INIT" CLEARS RCV ACTIVE
2805 010302  001401                             BEQ     3$
2806
2807 010304  104115                             ERROR   .115            ;INIT DID NOT CLEAR RCV ACTIVE
2808
2809 010306  005000                     3$:     CLR     R0              ;CLEAR A TIMER
2810 010310  052777  000004  172322             BIS     @#BIT2,@TCSR    ;SET MAINTENANCE WRAP
2811 010316  062700  000000             WT:     ADD     #0,R0           ;WAIT AT LEAST ONE BIT TIME
2812 010322  005200                             INC     R0
2813 010324  001374                             BNE     WT
2814 010326  033777  004000  172300             BIT     BIT11,@RCSR     ;VERIFY RCV ACTIVE STILL CLEAR
2815 010334  001404                             BEQ     RCVDON          ;BR IF CLEAR
2816
2817 010336  042777  000004  172314             BIC     @#BIT2,@TCSR    ;** DISABLE MAINTENANCE MODE FOR
                                                                         ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                         ;** WITH TERMINAL.
2818 010344  104116                             ERROR   .116            ;RCV ACTIVE WITHOUT "START" BIT
2819
```

TEST # 30 TEST THAT RCVR ACTIVE & DONE SET & CLEAR

```
2820 010346  052777  000004  172264  RCVDON: BIS     #BIT2,@TCSR     ;SET MAINTENCE WRAP
2821 010354  005000                  CLR     R0              ;CLEAR TIMER
2822 010356  005077  172260          CLR     @TBUF           ;LOAD TRANSMIT BUFFER
2823 010362  105777  172246          WDONE:  TSTB    @RCSR           ;CHECK FOR RECEIVER DONE
2824 010366  100406                  BMI     5$              ;BR, IF DONE
2825 010370  005200                  INC     R0              ;INCREMENT TIMER, IF NOT DONE
2826 010372  001373                  BNE     WDONE           ;CONTINUE WAIT IF TIME REMAINS
2827 010374  042777  000004  172256          BIC     #BIT2,@TCSR     ;** DISABLE MAINTENANCE MODE FOR
                                                                ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                ;** WITH TERMINAL.
2828 010402  104064                  ERROR   .64
2829                                                            ;RECEIVER DONE NEVER SET
2830
2831 010404  032737  000001  003002  5$:     BIT     #BIT0,FLAG44    ;** 11/44??
2832 010412  001010                  BNE     6$              ;** DO NOT EXECUTE THIS SECTION
2833 010414  032777  004000  172212          BIT     #BIT11,@RCSR    ;CHECK FOR RCVR ACTIVE CLEAR
2834 010422  001404                  BEQ     6$              ;BR, IF CLEAR
2835 010424  042777  000004  172226          BIC     #BIT2,@TCSR     ;** DISABLE MAINTENANCE MODE FOR
                                                                ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                ;** WITH TERMINAL.
2836 010432  :04065                  ERROR   .65
2837                                                            ;RCVR ACTIVE DID NOT CLEAR WITH RCVR DONE
2838
2839 010434  000005          6$:     RESET                   ;CLEAR DONE WITH RESET
2840 010436  105777  172172          TSTB    @RCSR           ;CHECK FOR DONE CLEAR
2841 010442  001404                  BEQ     7$
2842
2843 010444  042777  000004  172206          BIC     #BIT2,@TCSR     ;** DISABLE MAINTENANCE MODE FOR
                                                                ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                ;** WITH TERMINAL.
2844 010452  104066                  ERROR   .66
2845                                                            ;RESET DID NOT CLEAR RCVR DONE
2846
2847 010454                  7$:
2848 010454  042777  000004  172176          BIC     #BIT2,@TCSR     ;** DISABLE MAINTENANCE MODE FOR
                                                                ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                ;** WITH TERMINAL.
```

```
2849                                    .SBTTL  TEST # 31 - TEST THAT READING RBUF CLEARS RECEIVER DONE
                                        ;;*****************************************************************
                                        ;*TEST 31        TEST THAT READING RBUF CLEARS RECEIVER DONE
                                        ;;*****************************************************************
     010462  000004                     TST31:  SCOPE
2850 010464  000005                             RESET                           ;CLEAR EVERYTHING
2851 010466  052777  000004  172144             BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
2852 010474  005000                             CLR     R0                      ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     010476  012701  000002                     MOV     #2,R1                   ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                                 ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                                 ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     010502  105777  172130             42$:    TSTB    @RBUF                   ;** READ TO CLEAR DONE
     010506  005300                             DEC     R0                      ;**
     010510  001374                             BNE     42$                     ;**
     010512  005301                             DEC     R1                      ;**
     010514  001372                             BNE     42$                     ;**
2853 010516  005077  172120                     CLR     @TBUF                   ;LOAD TRANSMITTER
2854 010522  105777  172106             1$:     TSTB    @RCSR                   ;WAIT FOR RECEIVER DONE
2855 010526  100375                             BPL     1$
2856 010530  017700  172102                     MOV     @RBUF,R0                ;READ RECEIVE BUFFER
2857 010534  042777  000004  172116             BIC     #BIT2,@TCSR             ;** DISABLE MAINTENANCE MODE FOR
                                                                                 ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                                 ;** WITH TERMINAL.
2858 010542  105777  172066                     TSTB    @RCSR                   ;CHECK FOR RECEIVE DONE CLEAR
2859 010546  001401                             BEQ     10$                     ;** BR, IF CLEAR TO NEXT TEST
2860 010550  104070                             ERROR   .70
2861                                                                             ;READING RBUF DID NOT CLEAR RCVR DONE
2862 010552  000240                     10$:    NOP
2863
```

```
2864                                    .SBTTL  TEST # 32 - TEST THAT RDR ENABLE CLEARS RECEIVER DONE FLAG
                                        ;;*****************************************************************
                                        ;*TEST 32       TEST THAT RDR ENABLE CLEARS RECEIVER DONE FLAG
                                        ;;*****************************************************************
     010554  000004                     TST32:  SCOPE
2865 010556  032737  000001  003002             BIT     @#BIT0,FLAG44   ;** 11/44 ??
2866 010564  001044                             BNE     10$             ;** YES DO NOT EXECUTE THIS TEST
2867 010566  000005                             RESET                   ;CLEAR EVERYTHING
2868 010570  052777  000001  172036             BIS     @#BIT0,@RCSR    ;SET RDR ENABLE
2869 010576  052777  000004  172034             BIS     @#BIT2,@TCSR    ;SET MAINTENANCE WRAP
2870 010604  005000                             CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     010606  012701  000002                     MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                         ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                         ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     010612  105777  172020             42$:    TSTB    @RBUF           ;** READ TO CLEAR DONE
     010616  005300                             DEC     R0              ;**
     010620  001374                             BNE     42$             ;**
     010622  005301                             DEC     R1              ;**
     010624  001372                             BNE     42$             ;**
2871 010626  005077  172010                     CLR     @TBUF           ;LOAD TRANSMITTER
2872 010632  105777  171776             1$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
2873 010636  100375                             BPL     1$
2874 010640  032777  000001  171766             BIT     @#BIT0,@RCSR    ;VERIFY RCV ACTIVE CLEARED RDR ENABLE
2875 010646  001401                             BEQ     2$              ;BR IF CLEAR
2876
2877 010650  104117                             ERROR   .117            ;RDR ENABLE NOT CLEARED WITH RCV ACTIVE
2878
2879 010652  052777  000001  171754     2$:     BIS     @#BIT0,@RCSR    ;CLEAR DONE BY SETTING RDR ENABLE
2880 010660  105777  171750                     TSTB    @RCSR           ;CHECK FOR DONE CLEAR
2881 010664  001404                             BEQ     10$             ;** BR, IF CLEAR TO NEXT TEST
2882 010666  042777  000004  171764             BIC     @#BIT2,@TCTCSR  ;** DISABLE MAINTENANCE MODE FOR
                                                                         ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                         ;** WITH TERMINAL.
2883 010674  104067                             ERROR   .67
2884                                                                     ;SETTING RDR ENABLE DID NOT CLEAR RCVR DONE
2885 010676  000240                     10$:    NOP                     ;**
```

```
2886                                    .SBTTL  TEST # 33 - TEST THAT RCVR INTERRUPTS ONLY WHEN ENABLED
                                        ;;**********************************************************************
                                        ;*TEST 33        TEST THAT RCVR INTERRUPTS ONLY WHEN ENABLED
                                        ;;**********************************************************************
     010700  000004                     TST33:  SCOPE
2887 010702  042777  000100  171730             BIC     #BIT6,@TCSR     ;DISABLE TRANSMIT INTERRUPTS
2888 010710  042777  000100  171716             BIC     #BIT6,@RCSR     ;DISABLE RECEIVER INTERRUPTS
2889 010716  052777  000004  171714             BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
2890 010724  005000                             CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     010726  012701  000002                     MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                         ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                         ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     010732  105777  171700             42$:    TSTB    @RBUF           ;** READ TO CLEAR DONE
     010736  005300                             DEC     R0              ;**
     010740  001374                             BNE     42$             ;**
     010742  005301                             DEC     R1              ;**
     010744  001372                             BNE     42$             ;**
2891 010746  017703  171672                     MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
2892 010752  012777  011010  171664             MOV     #2$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
2893 010760  004737  014502                     JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2894 010764  000140                             .WORD   140
2895 010766  005077  171650                     CLR     @TBUF           ;SEND A CHARACTER
2896 010772  105777  171636             1$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
2897 010776  100375                             BPL     1$
2898 011000  042777  000004  171652             BIC     #BIT2,@CTCSR    ;** DISABLE MAINTENANCE MODE FOR
                                                                         ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                         ;** WITH TERMINAL.
2899 011006  000405                             BR      3$              ;CONTINUE TEST
2900
2901 011010                             2$:
2902 011010  042777  000004  171642             BIC     #BIT2,@CTCSR    ;** DISABLE MAINTENANCE MODE FOR
                                                                         ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                         ;** WITH TERMINAL.
2903 011016  022626                             CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2904 011020  104071                             ERROR   +71             ;RECEIVER INTERRPUTS WITH INT. ENABLE CLEAR
2905
2906 011022  012777  011060  171614     3$:     MOV     #4$,@RVECT      ;POINT RCV VECTOR TO END OF TEST
2907 011030  052777  000100  171576             BIS     #BIT6,@RCSR     ;ENABLE RCV INTERRUPTS
2908 011036  000240                             NOP                     ;** GIVE ANY INTERRUPTS TIME
2909 011040  000240                             NOP                     ;** GIVE ANY INTERRUPTS TIME
2910 011042  000240                             NOP                     ;** GIVE ANY INTERRUPTS TIME
2911 011044  000240                             NOP                     ;** GIVE ANY INTERRUPTS TIME
2912 011046  042777  000004  171604             BIC     #BIT2,@CTCSR    ;** DISABLE MAINTENANCE MODE FOR
                                                                         ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                         ;** WITH TERMINAL.
2913 011054  104072                             ERROR   +72             ;RCVR DID NOT INTERRUPT
2914 011056  000401                             BR      5$              ;BRANCH OVER STACK CORRECTION
2915 011060  022626                     4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2916 011062  042777  000100  171544     5$:     BIC     #BIT6,@RCSR     ;DISABLE INTERRUPTS
2917 011070  042777  000004  171562             BIC     #BIT2,@CTCSR    ;** DISABLE MAINTENANCE MODE FOR
                                                                         ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                         ;** WITH TERMINAL.
2918 011076  010377  171542                     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
```

```
2919                                        .SBTTL  TEST # 34 - TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
                                            ;;**********************************************************************
                                            ;*TEST 34       TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
                                            ;;**********************************************************************
     011102  000004                 TST34:  SCOPE
     011104  000005                         RESET                       ;CLEAR EVERYTHING
2920 011106  004737  014502                 JSR     PC,@WRPSW           ;SET PSW TO PRIORITY 7
2921 011112  000340                         .WORD   340
2922 011114  017703  171524                 MOV     @RVECT,R3           ;SAVE RECEIVE VECTOR
2923 011120  012777  011216  171516         MOV     #2$,@RVECT          ;POINT RCVR VECTOR TO ERROR REPORT
2924 011126  052777  000004  171504         BIS     #BIT2,@TCSR         ;SET MAINTENANCE WRAP
2925 011134  005000                         CLR     R0                  ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
2926 011136  012701  000002                 MOV     #2,R1               ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                         ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                         ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     011142  105777  171470         42$:    TSTB    @RBUF               ;** READ TO CLEAR DONE
     011146  005300                         DEC     R0                  ;**
     011150  001374                         BNE     42$                 ;**
     011152  005301                         DEC     R1                  ;**
     011154  001372                         BNE     42$                 ;**
2927 011156  105777  167666         6$:     TSTB    @$TPS               ;TEST FOR XMIT READY
2928 011162  100375                         BPL     6$                  ;LOOP IF NOT
2929 011164  005077  171452                 CLR     @TBUF               ;SEND A CHARACTER
2930 011170  105777  171440         1$:     TSTB    @RCSR               ;WAIT FOR RECEIVER DONE
2931 011174  100375                         BPL     1$
2932 011176  052777  000100  171430         BIS     #BIT6,@RCSR         ;ENABLE INTERRUPTS
2933 011204  000240                         NOP                         ;** GIVE TIME FOR INTERRUPT
2934 011206  000240                         NOP                         ;** GIVE TIME FOR INTERRUPT
2935 011210  000240                         NOP                         ;** GIVE TIME FOR INTERRUPT
2936 011212  000240                         NOP                         ;** GIVE TIME FOR INTERRUPT
2937 011214  000405                         BR      3$                  ;CONTINUE TEST
2938 011216                         2$:
2939 011216  042777  000004  171434         BIC     #BIT2,@CTCSR        ;** DISABLE MAINTENANCE MODE FOR
                                                                         ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                         ;** WITH TERMINAL.
2940 011224  022626                         CMP     (SP)+,(SP)+         ;RESTORE SP AFTER INTERRUPT
2941 011226  104073                         ERROR   +73                 ;RCVR INTERRUPTS AT PRIORITY 7
2942
2943 011230  042777  000100  171376  3$:     BIC     #BIT6,@RCSR         ;CLEAR INTERRUPT ENABLE
2944 011236  012777  011266  171400         MOV     #4$,@RVECT          ;POINT RCVR VECTOR TO ERROR REPORT
2945 011244  004737  014502                 JSR     PC,@WRPSW           ;SET PSW TO PRIORITY 3
2946 011250  000140                         .WORD   140
2947 011252  000240                         NOP                         ;GIVE TIME FOR ANY INTERRUPT
2948 011254  000240                         NOP                         ;GIVE TIME FOR ANY INTERRUPT
2949 011256  042777  000004  171374         BIC     #BIT2,@CTCSR        ;** DISABLE MAINTENANCE MODE FOR
                                                                         ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                         ;** WITH TERMINAL.
2950 011264  000405                         BR      5$                  ;BR TO END OF TEST, IF NO INTERRUPT
2951
2952 011266                         4$:
2953 011266  042777  000004  171364         BIC     #BIT2,@CTCSR        ;** DISABLE MAINTENANCE MODE FOR
                                                                         ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                         ;** WITH TERMINAL.
2954 011274  022626                         CMP     (SP)+,(SP)+         ;RESTORE SP AFTER INTERRUPT
2955 011276  104074                         ERROR   +74
2956                                                                     ;RCVR INTERRUPT REQUEST PASSED WITH BIT6 CLEAR
2957 011300  010377  171340         5$:     MOV     R3,@RVECT           ;RESTORE RECEIVE VECTOR
```

```
2958                                    .SBTTL  TEST # 35 - TEST RECEIVER FOR DOUBLE INTERRUPTS
                                        ;;**********************************************************
                                        ;*TEST 35        TEST RECEIVER FOR DOUBLE INTERRUPTS
                                        ;;**********************************************************
     011304  000004                     TST35:  SCOPE
2959 011306  000005                             RESET                   ;CLEAR EVERYTHING
2960 011310  017703  171330                     MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
2961 011314  017704  171326                     MOV     @RPSW,R4        ;SAVE RECEIVE PSW VECTOR
2962 011320  012777  011434  171316             MOV     #2$,@RVECT      ;POINT RCV VECTOR TO CONTINUE TEST
2963 011326  012777  000340  171312             MOV     #340,@RPSW      ;SET PRIORITY TO 7 AFTER INTERRUPT
2964 011334  004737  014502                     JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
2965 011340  000140                             .WORD   140
2966 011342  052777  000004  171270             BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
2967 011350  005000                             CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     011352  012701  000002                     MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                         ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                         ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     011356  105777  171254             42$:    TSTB    @RBUF           ;** READ TO CLEAR DONE
     011362  005300                             DEC     R0              ;**
     011364  001374                             BNE     42$             ;**
     011366  005301                             DEC     R1              ;**
     011370  001372                             BNE     42$             ;**
2968 011372  005077  171244                     CLR     @TBUF           ;SEND A CHARACTER
2969 011376  105777  171232             1$:     TSTB    @RCSR           ;WAIT FOR RCVR DONE
2970 011402  100375                             BPL     1$
2971 011404  042777  000004  171246             BIC     #BIT2,@TCSR     ;** DISABLE MAINTENANCE MODE FOR
                                                                         ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                         ;** WITH TERMINAL.
2972 011412  052777  000100  171214             BIS     #BIT6,@RCSR     ;ENABLE RCV INTERRUPTS
2973 011420  000240                             NOP                     ;** GIVE SOME TIME
2974 011422  000240                             NOP                     ;** GIVE SOME TIME
2975 011424  000240                             NOP                     ;** GIVE SOME TIME
2976 011426  000240                             NOP                     ;** GIVE SOME TIME
2977
2978 011430  104075                             ERROR   .75             ;RCVR INTERRUPT DID NOT OCCUR
2979 011432  000401                             BR      25$             ;BRANCH OVER STACK CORRECTION
2980 011434  022626                     2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2981 011436  012777  011502  171200     25$:    MOV     #3$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
2982 011444  004737  014502                     JSR     PC,WRPSW        ;RESET PSW TO PRIORITY 3
2983 011450  000140                             .WORD   140
2984 011452  000240                             NOP                     ;** GIVE SOME TIME
2985 011454  000240                             NOP                     ;** GIVE SOME TIME
2986 011456  000240                             NOP                     ;** GIVE SOME TIME
2987 011460  000240                             NOP                     ;** GIVE SOME TIME
2988 011462  042777  000100  171144             BIC     #BIT6,@RCSR     ;CLEAR INTERRUPT ENABLE
2989 011470  010377  171150                     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
2990 011474  010477  171146                     MOV     R4,@RPSW        ;RESTORE RECEIVE PSW VECTOR
2991 011500  000402                             BR      4$              ;BR TO END OF TEST
2992
2993 011502  022626                     3$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
2994 011504  104076                             ERROR   .76
2995                                                                     ;RECEIVER RE-INTERRUPTED
2996 011506  010377  171132             4$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
```

```
2997                                        .SBTTL  TEST # 36 - TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
                                            ;;*****************************************************************
                                            ;*TEST 36       TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
                                            ;;*****************************************************************
     011512  000004                         TST36:  SCOPE
2998 011514  000005                                 RESET                   ;CLEAR EVERYTHING
2999 011516  004737  014502                         JSR     PC,@WRPSW       ;SET PSW PRIORITY TO 7
3000 011522  000340                                 .WORD   340
3001 011524  017703  171114                         MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
3002 011530  012777  011646  171106                 MOV     #2$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
3003 011536  052777  000100  171070                 BIS     #BIT6,@RCSR     ;SET RCVR INTERRUPT ENABLE
3004 011544  052777  000004  171066                 BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
3005 011552  005000                                 CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     011554  012701  000002                         MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                             ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                             ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     011560  105777  171052                 42$:    TSTB    @RBUF           ;** READ TO CLEAR DONE
     011564  005300                                 DEC     R0              ;**
     011566  001374                                 BNE     42$             ;**
     011570  005301                                 DEC     R1              ;**
     011572  001372                                 BNE     42$             ;**
3006 011574  005077  171042                         CLR     @TBUF           ;SEND A CHARACTER
3007 011600  105777  171030                 1$:     TSTB    @RCSR           ;WAIT FOR DONE (INTERRUPT)
3008 011604  100375                                 BPL     1$
3009 011606  042777  000004  171044                 BIC     #BIT2,@TCSR     ;** DISABLE MAINTENANCE MODE FOR
                                                                             ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                             ;** WITH TERMINAL.
3010 011614  005777  171016                         TST     @RBUF           ;READ RBUF TO CLEAR PENDING INTERRPUT   ;DPM002
3011 011620  004737  014502                         JSR     PC,@WRPSW       ;SET PSW TO PRIORITY 3
3012 011624  000140                                 .WORD   140
3013 011626  000240                                 NOP                     ;** ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
3014 011630  000240                                 NOP                     ;** ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
3015 011632  000240                                 NOP                     ;** ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
3016 011634  000240                                 NOP                     ;** ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
3017 011636  042777  000100  170770                 BIC     #BIT6,@RCSR     ;NO INTERRUPT-CLEAR INT. ENABLE
3018 011644  000402                                 BR      3$
3019
3020 011646  022626                         2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
3021 011650  104077                                 ERROR   +77             ;READING RBUF DID NOT CLEAR INTERRUPT
3022
3023 011652  010377  170766                 3$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
```

```
3024                                    .SBTTL  TEST # 37 - TEST THAT RESET CLEARS RECEIVE INTERRUPT
                                        ;;*****************************************************************
                                        ;*TEST 37       TEST THAT RESET CLEARS RECEIVE INTERRUPT
                                        ;;*****************************************************************
     011656  000004                     TST37:  SCOPE
                                             RESET                          ;CLEAR EVERYTHING
3025 011660  000005
3026 011662  004737  014502                JSR     PC,@RPSW                ;SET PSW TO PRIORITY 7
3027 011666  000340                             .WORD   340
3028 011670  017703  170750                MOV     @RVECT,R3               ;SAVE RECEIVE VECTOR
3029 011674  012777  012004  170742        MOV     #2$,@RVECT              ;POINT RCV VECTOR TO ERROR REPORT
3030 011702  052777  000100  170724        BIS     #BIT6,@RCSR             ;SET RCV INTERRUPT ENABLE
3031 011710  052777  000004  170722        BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
3032 011716  005000                        CLR     R0                      ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     011720  012701  000002                MOV     #2,R1                   ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                             ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                             ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     011724  105777  170706            42$:    TSTB    @RBUF               ;** READ TO CLEAR DONE
     011730  005300                             DEC     R0                  ;**
     011732  001374                             BNE     42$                 ;**
     011734  005301                             DEC     R1                  ;**
     011736  001372                             BNE     42$                 ;**
3033 011740  012777  000377  170674        MOV     #377,@TBUF              ;SEND AN ALL 1'S CHARACTER
3034 011746  105777  170662            1$:     TSTB    @RCSR               ;WAIT FOR RCV DONE
3035 011752  100375                             BPL     1$
3036 011754  000005                             RESET                       ;CLEAR RCV INTERRUPT & RBUF
3037 011756  004737  014502                     JSR     PC,@RPSW            ;SET PSW TO PRIORITY 3
3038 011762  000140                             .WORD   140
3039 011764  000240                             NOP                         ;** ALLOW TIME FOR AN ERRONEOUS INTERRUPT
3040 011766  000240                             NOP                         ;** ALLOW TIME FOR AN ERRONEOUS INTERRUPT
3041 011770  000240                             NOP                         ;** ALLOW TIME FOR AN ERRONEOUS INTERRUPT
3042 011772  000240                             NOP                         ;** ALLOW TIME FOR AN ERRONEOUS INTERRUPT
3043 011774  042777  000100  170632        BIC     #BIT6,@RCSR             ;NO INTERRUPT-CLEAR INT. ENABLE
3044 012002  000402                        BR      3$                      ;CONTINUE TEST
3045
3046
3047 012004  022626                    2$:     CMP     (SP)+,(SP)+         ;RESTORE SP AFTER INTERRUPT
3048 012006  104100                            ERROR   +100                ;RESET DID NOT CLEAR RCVR INTERRUPT
3049
3050 012010  010377  170630            3$:     MOV     R3,@RVECT           ;RESTORE RECEIVE VECTOR
```

```
3051                                    .SBTTL  TEST # 40 - TEST THAT THE "OR" ERROR & "ERROR" CAN BE SET
                                        ;;**********************************************************************
                                        ;*TEST 40        TEST THAT THE "OR" ERROR & "ERROR" CAN BE SET
                                        ;;**********************************************************************
     012014  000004                     TST40:  SCOPE
3052 012016  032777  002000  167014             BIT     @#BIT10,@#SWR   ;IS THIS TEST ENABLED
3053 012024  001465                             BEQ     TST41           ;IF NOT ENABLED, BR TO NEXT TEST
3054 012026  032737  000001  003002             BIT     @#BIT0,FLAG44   ;** IS THIS A 11/44
3055 012034  001407                             BEQ     9$              ;** NO
3056 012036  005737  002624                     TST     CTSTFL          ;** YES THIS IS 11/44. IS THIS THE CONSOLE
3057                                                                    ;** SLU
3058 012042  001404                             BEQ     9$              ;** NO
3059 012044  032777  000010  166766             BIT     @#BIT03,@#SWR   ;** THIS IS THE CONSOLE SLU.SHOULD THE OVERRUN
3060                                                                    ;** ERROR TEST BE PERFORMED
3061 012052  001452                             BEQ     TST41           ;** NO
3062 012054  000005                     9$:     RESET                   ;CLEAR EVERYTHING
3063 012056  052777  000004  170554             BIS     @#BIT2,@#TCSR   ;SET MAINTENANCE WRAP
3064 012064  005000                             CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     012066  012701  000002                     MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                        ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                        ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     012072  105777  170540             42$:    TSTB    @#RBUF          ;** READ TO CLEAR DONE
     012076  005300                             DEC     R0              ;**
     012100  001374                             BNE     42$             ;**
     012102  005301                             DEC     R1              ;**
     012104  001372                             BNE     42$             ;**
3065 012106  012700  000003                     MOV     #3,R0           ;SET CHARACTER COUNT TO SEND 3 CHAR.
3066 012112  005077  170524             1$:     CLR     @#TBUF          ;LOAD TRANSMIT BUFFER
3067 012116  105777  170516             2$:     TSTB    @#TCSR          ;WAIT FOR TRANSMIT DONE
3068 012122  100375                             BPL     2$
3069 012124  005300                             DEC     R0              ;DECREMENT CHARACTER COUNT
3070 012126  001371                             BNE     1$              ;BR IF ALL CHARACTERS NOT TRANSMITTED
3071 012130  042777  000004  170522             BIC     @#BIT2,@#TCTCSR ;** DISABLE MAINTENANCE MODE FOR
                                                                        ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                        ;** WITH TERMINAL.
3072 012136  032777  040000  170472             BIT     @#BIT14,@#RBUF  ;TEST FOR "OR" ERROR FLAG
3073 012144  001001                             BNE     3$              ;BR, IF SET
3074 012146  104101                             ERROR   +101
3075                                                                    ;"OR" ERROR FLAG DID NOT SET
3076
3077 012150  032777  100000  170460  3$:        BIT     @#BIT15,@#RBUF  ;TEST "ERROR" FLAG
3078 012156  001001                             BNE     4$              ;BR, IF SET
3079 012160  104102                             ERROR   +102
3080                                                                    ;"ERROR" FLAG DID NOT SET WITH "OR" FLAG
3081 012162                             4$:
3082 012162  005000                             CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     012164  012701  000002                     MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
     012170  005300                     40$:    DEC     R0              ;** TRANSMITTED TO FINISH BEFORE MAINTENANCE
     012172  001376                             BNE     40$             ;** WRAP FOR THE UART UNDER TEST IS DISABLED.
     012174  005301                             DEC     R1              ;** THIS WILL INHIBIT ANY COMMUNICATION
     012176  001374                             BNE     40$             ;** TO HARDWARE MEDIA SUCH AS TU58 THAT MIGHT
                                                                        ;** BE ATTACHED TO UART UNDER TEST.
```

```
3083                                    .SBTTL  TEST # 41 - TEST THAT BREAK TRANSMITS ALL ZEROES
                                        ;;*****************************************************************
                                        ;*TEST 41        TEST THAT BREAK TRANSMITS ALL ZEROES
                                        ;;*****************************************************************
     012200  000004                     TST41:  SCOPE
3084 012202  032777  000400  166630             BIT     #BIT8,@SWR      ;IS BREAK FUNCTION TEST ENABLED?
3085 012210  001501                             BEQ     TST42           ;BR TO NEXT TEST, IF NOT ENABLED
3086 012212  032737  000001  003002             BIT     #BIT0,FLAG44    ;** IS THIS A 11/44
3087 012220  001407                             BEQ     9$              ;** NO
3088 012222  005737  002624                     TST     CTSTFL          ;** YES THIS IS 11/44. IS THIS THE CONSOLE
3089                                                                    ;** SLU
3090 012226  001404                             BEQ     9$              ;** NO
3091 012230  032777  000010  166602             BIT     #BIT03,@SWR     ;** THIS IS THE CONSOLE SLU.SHOULD THE BREAK
3092                                                                    ;** TEST BE PERFORMED
3093 012236  001466                             BEQ     TST42           ;** NO
3094 012240  000005                     9$:     RESET                   ;CLEAR EVERYTHING
3095 012242  052777  000004  170370             BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
3096 012250  005000                             CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     012252  012701  000002                     MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                        ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                        ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     012256  105777  170354             42$:    TSTB    @RBUF           ;** READ TO CLEAR DONE
     012262  005300                             DEC     R0              ;**
     012264  001374                             BNE     42$             ;**
     012266  005301                             DEC     R1              ;**
     012270  001372                             BNE     42$             ;**
3097 012272  012777  177777  170342             MOV     #-1,@TBUF       ;TRANSMIT ALL ONES TO RCVR
3098 012300  105777  170330             1$:     TSTB    @RCSR           ;WAIT FOR RCVR DONE
3099 012304  100375                             BPL     1$
3100 012306  005077  170324                     CLR     @RBUF           ;CLEAR DONE (LEAVING ALL ONES IN RBUF)
3101 012312  052777  000001  170320             BIS     #BIT0,@TCSR     ;TRANSMIT BREAK
3102 012320  005000                             CLR     R0              ;CLEAR A TIMER
3103 012322  117737  170306  001026     2$:     MOVB    @RCSR,#BDDAT    ;WAIT FOR RCVR DONE
3104 012330  100411                             BMI     CONT41          ;BR IF DONE
3105 012332  005200                             INC     R0              ;IF NOT, INCREMENT TIMER
3106 012334  001372                             BNE     2$              ;BR IF TIME REMAINS
3107
3108 012336  042777  000001  170274             BIC     #BIT0,@TCSR       ;CLEAR BREAK  BIT
3109 012344  042777  000004  170306             BIC     #BIT2,@CTCSR    ;** DISABLE MAINTENANCE MODE FOR
                                                                        ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                        ;** WITH TERMINAL.
3110 012352  104103                             ERROR   .+103           ;BREAK DID NOT TRANSMIT ANYTHING
3111
3112 012354  105777  170256             CONT41: TSTB    @RBUF           ;CHECK RECEIVE BUFFER FOR ZERO
3113 012360  001407                             BEQ     3$              ;BR, IF ZERO
3114 012362  042777  000001  170250             BIC     #BIT0,@TCSR       ;CLEAR  BREAK BIT
3115 012370  042777  000004  170262             BIC     #BIT2,@CTCSR    ;** DISABLE MAINTENANCE MODE FOR
                                                                        ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                        ;** WITH TERMINAL.
3116
3117 012376  104103                             ERROR   .+103           ;BREAK DID NOT TRANSMIT ALL ZEROES
3118
3119 012400  042777  000001  170232     3$:     BIC     #BIT0,@TCSR     ;CLEAR  BREAK BIT
3120 012406  042777  000004  170244             BIC     #BIT2,@CTCSR    ;** DISABLE MAINTENANCE MODE FOR
                                                                        ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                        ;** WITH TERMINAL.
```

```
3121                                    .SBTTL  TEST # 42 - TEST THAT "FR" ERROR CAN BE SET DURING BREAK
                                        ;;*****************************************************************
                                        ;*TEST 42        TEST THAT "FR" ERROR CAN BE SET DURING BREAK
                                        ;;*****************************************************************
     012414  000004                     TST42:  SCOPE
3122 012416  032777  002000  166414             BIT     @BIT10,@SWR     ;IS THE "TEST ERROR FLAGS" BIT SET
3123 012424  001464                             BEQ     TST43           ;BR TO NEXT TEST, IF NOT SET
3124 012426  032777  000400  166404             BIT     @BIT8,@SWR      ;IS BREAK FUNCTION ENABLED
3125 012434  001460                             BEQ     TST43           ;BR TO NEXT TEST, IF NOT SET
3126 012436  032737  000001  003002             BIT     @BIT0,FLAG44    ;** IS THIS A 11/44
3127 012444  001407                             BEQ     9$              ;** NO
3128 012446  005737  002624                     TST     CTSTFL          ;** YES THIS IS 11/44. IS THIS THE CONSOLE
3129                                                                     ;** SLU
3130 012452  001404                             BEQ     9$              ;** NO
3131 012454  032777  000010  166356             BIT     @BIT03,@SWR     ;** THIS IS THE CONSOLE SLU.SHOULD THE FRAME
3132                                                                     ;** ERROR TEST BE PERFORMED
3133 012462  001445                             BEQ     TST43           ;** NO
3134 012464  000005                     9$:     RESET                   ;CLEAR EVERYTHING
3135 012466  052777  000004  170144             BIS     @BIT2,@TCSR     ;SET MAINTENANCE WRAP
3136 012474  005000                             CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     012476  012701  000002                     MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                     ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                     ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     012502  105777  170130             42$:    TSTB    @RBUF           ;** READ TO CLEAR DONE
     012506  005300                             DEC     R0              ;**
     012510  001374                             BNE     42$             ;**
     012512  005301                             DEC     R1              ;**
     012514  001372                             BNE     42$             ;**
3137 012516  052777  000001  170114             BIS     @BIT0,@TCSR     ;SEND BREAK
3138 012524  005077  170112                     CLR     @TBUF           ;TRANSMIT A CHARACTER TO TIME BREAK
3139 012530  105777  170100             1$:     TSTB    @RCSR           ;WAIT FOR RCVR DONE
3140 012534  100375                             BPL     1$
3141 012536  042777  000001  170074             BIC     @BIT0,@TCSR     ;CLEAR  BREAK BIT
3142 012544  042777  000004  170106             BIC     @BIT2,@CTCSR    ;** DISABLE MAINTENANCE MODE FOR
                                                                     ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                     ;** WITH TERMINAL.
3143 012552  032777  020000  170056             BIT     @BIT13,@RBUF    ;CHECK FOR FRAMING ERROR FLAG
3144 012560  001001                             BNE     2$              ;BR, IF SET
3145
3146 012562  104104                             ERROR   .104
3147                                                                    ;BREAK DID NOT SET FRAMING ERROR
3148 012564  032777  100000  170044     2$:     BIT     @BIT15,@RBUF    ;TEST "ERROR"FLAG
3149 012572  001001                             BNE     3$              ;BR, IF SET
3150
3151 012574  104114                             ERROR   .114
3152                                                                    ;"ERROR" FLAG DID NOT SET WITH "OR" FLAG
3153 012576                             3$:
```

```
3154                                    .SBTTL  TEST # 43 - TEST DATA PATH FROM XMIT TO REC USING MAINT WRAP
                                        ;;*****************************************************************
                                        ;*TEST 43        TEST DATA PATH FROM XMIT TO REC USING MAINT WRAP
                                        ;;*****************************************************************
     012576  000004                    TST43:  SCOPE
3155
3156 012600  000005                            RESET                   ;CLEAR EVERYTHING
3157 012602  052777  000004  170030            BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
3158                                                                    ;TRANSMIT A BINARY COUNT PATTERN - UP
3159                                                                    ;TO THE BIT POSITION INDICATED BY THE
3160                                                                    ;CONTENTS OF LOCATION "$USWR"
3161 012610  005000                            CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     012612  012701  000002                    MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                        ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                        ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     012616  105777  170014            42$:    TSTB    @RBUF           ;** READ TO CLEAR DONE
     012622  005300                            DEC     R0              ;**
     012624  001374                            BNE     42$             ;**
     012626  005301                            DEC     R1              ;**
     012630  001372                            BNE     42$             ;**
3162 012632  005001                            CLR     R1              ;CLEAR REGISTER FOR TEST DATA
3163 012634  105201                    1$:     INCB    R1              ;INCREMENT THE TEST DATA
3164 012636  0"2737  000001  003002            BIT     #BIT0,FLAG44    ;** 11/44 CPU
3165 012644  001406                            BEQ     5$              ;** TEST ALL DATA
3166 012646  122701  000020                    CMPB    #20,R1          ;** CHECK  FOR CONT-P IN TEST DATA
3167 012652  001770                            BEQ     1$              ;** DO NOT XMIT ↑P
3168 012654  122701  000220                    CMPB    #220,R1         ;**
3169 012660  001765                            BEQ     1$              ;** DO NOT XMIT 220
3170 012662  010177  167754            5$:     MOV     R1,@TBUF        ;XMIT A CHARACTER
3171 012666  105777  167742            2$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
3172 012672  100375                            BPL     2$
3173 012674  017702  167736                    MOV     @RBUF,R2        ;GET RECEIVED CHARACTER
3174 012700  043701  001112                    BIC     @#$USWR,R1      ;CLEAR LOWEST UNUSED DATA BIT POSITITON IN TEST DATA
3175 012704  020102                            CMP     R1,R2           ;COMPARE DATA
3176 012706  001003                            BNE     3$              ;BR, IF NON-COMPARE
3177 012710  105701                            TSTB    R1              ;TEST XMIT DATA FOR ZERO
3178 012712  001411                            BEQ     4$              ;BR, IF FINISHED
3179 012714  000747                            BR      1$              ;CONTINUE IF NOT
3180 012716  010137  001024            3$:     MOV     R1,$GDDAT       ;STORE THE EXPECTED DATA
3181 012722  010237  001026                    MOV     R2,$BDDAT       ;STORE RECEIVED DATA
3182 012726  042777  000004  167724            BIC     #BIT2,@TCTCSR   ;** DISABLE MAINTENANCE MODE FOR
                                                                        ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                        ;** WITH TERMINAL.
3183 012734  104105                            ERROR   +105            ;DATA COMPARE DATA
3184
3185 012736                            4$:
3186 012736  042777  000004  167714            BIC     #BIT2,@TCTCSR   ;** DISABLE MAINTENANCE MODE FOR
                                                                        ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                                                        ;** WITH TERMINAL.
3187
3188
3189
```

```
3190                                    .SBTTL  TEST # 44 - TEST DATA PATHS USING LOOP-BACK CONNECTOR
                                        ;;*****************************************************************
                                        ;*TEST 44        TEST DATA PATHS USING LOOP-BACK CONNECTOR
                                        ;;*****************************************************************
     012744  000004                     TST44:  SCOPE
3191 012746  032777  000200  166064             BIT     @BIT7,@SWR      ;IS THIS TEST ENABLED
3192 012754  001442                             BEQ     TST45           ;BR, IF NOT
3193 012756  000005                             RESET                   ;CLEAR EVERYTHING
3194 012760  005001                             CLR     R1              ;CLEAR REGISTER FOR TEST DATA
3195                                                                    ;TRANSMIT A BINARY COUNT PATTERN - UP
3196                                                                    ;TO THE BIT POSITION INDICATED BY THE
3197                                                                    ;CONTENTS OF LOCATION "(USWR"
3198 012762  105201                     1$:     INCB    R1              ;INCREMENT THE TEST DATA
3199 012764  032737  000001  003002             BIT     @BIT0,FLAG44    ;11/44 CPU
3200 012772  001404                             BEQ     5$              ;TEST ALL DATA
3201 012774  023727  002634  177560             CMP     RCSR,@177560    ;IS THIS 11/44 CONSOLE TERMINAL SLU?
3202 013002  001427                             BEQ     TST45           ;YES, SKIP THIS TEST
3203 013004  010177  167632             5$:     MOV     R1,@TBUF        ;XMIT A CHARACTER
3204 013010  005000                             CLR     R0              ;CLEAR A TIMER
3205 013012  105777  167616             2$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
3206 013016  100403                             BMI     3$              ;BR IF DONE
3207 013020  005200                             INC     R0              ;INCREMENT TIMER IF NOT
3208 013022  001373                             BNE     2$              ;BR IF TIME REMAINS
3209
3210 013024  104064                             ERROR   +64             ;RECEIVER DONE NOT SET
3211
3212 013026  017702  167604             3$:     MOV     @RBUF,R2        ;GET RECEIVED CHARACTER
3213 013032  043701  001112                     BIC     @#USWR,R1       ;CLEAR LOWEST UNUSED DATA BIT POSITITON IN TEST DATA
3214 013036  020102                             CMP     R1,R2           ;COMPARE DATA
3215 013040  001003                             BNE     4$              ;BR, IF NON-COMPARE
3216 013042  105701                             TSTB    R1              ;TEST XMIT DATA FOR ZERO
3217 013044  001406                             BEQ     TST45           ;BR, IF FINISHED
3218 013046  000745                             BR      1$              ;CONTINUE IF NOT
3219 013050  010137  001024             4$:     MOV     R1,$GDDAT       ;STORE EXPECTED DATA
3220 013054  010237  001026                     MOV     R2,$BDDAT       ;STORE RECEIVED DATA
3221
3222 013060  104106                             ERROR   +106            ;DATA COMPARE ERROR USING LOOP-BACK CONNECTOR
```

```
3223                                    .SBTTL  TEST # 45 - TEST DL11-W LOGIC BY EXERCISING THE XMIT,REC, & CLOCK
                                        ;;*****************************************************************
                                        ;*TEST 45       TEST DL11-W LOGIC BY EXERCISING THE XMIT,REC, & CLOCK
                                        ;;*****************************************************************
     013062  000004                     TST45:  SCOPE
3224 013064  000005                             RESET                   ;CLEAR EVERYTHING
3225 013066  005037  002444                     CLR     @CLKCNT         ;INITIALIZE CLOCK COUNT TO ZERO
3226 013072  004737  014502                     JSR     PC,@RPSW        ;SET PRIORITY TO 7
3227 013076  000340                             .WORD   340
3228 013100  017703  167544                     MOV     @TVECT,R3       ;SAVE XMIT VECTOR
3229 013104  017704  167534                     MOV     @RVECT,R4       ;SAVE RECEIVE VECTOR
3230 013110  017705  167562                     MOV     @RTCVT,R5       ;SAVE CLOCK VECTOR
3231 013114  017737  167532  002446             MOV     @TPSW,STPSW     ;SAVE XMIT PSW VECTOR
3232 013122  017737  167520  002450             MOV     @RPSW,SRPSW     ;SAVE RECEIVE PSW VECTOR
3233 013130  017737  167544  002452             MOV     @RTCPSW,SCPSW   ;SAVE CLOCK PSW VECTOR
3234 013136  012777  013542  167504             MOV     #XMIT,@TVECT    ;POINT TRANSMIT VECTOR TO TRANSMIT ROUTINE
3235 013144  012777  000200  167500             MOV     #200,@TPSW      ;NO MULTIPLE INTERUPTS ALLOWED
3236 013152  012777  013614  167464             MOV     #RCV,@RVECT     ;POINT RECEIVE VECTOR TO RECEIVE ROUTINE
3237 013160  012777  000200  167460             MOV     #200,@RPSW      ;NO MULT INTERUPTS
3238 013166  005737  002624                     TST     CTSTFL          ;IS CONSOLE UNDER TEST?
3239 013172  001415                             BEQ     1$              ;IF NOT SKIP CLOCK SET UP
3240 013174  032777  000100  165636             BIT     #BIT6,@SWR      ;IF YES, ARE CLOCK TEST DISABLED?
3241 013202  001011                             BNE     1$              ;IF YES, SKIP CLOCK SET UP
3242 013204  012777  013630  167464             MOV     #CLK,@RTCVT     ;POINT VECTOR TO CLOCK INTERRUPT ROUTINE
3243 013212  012777  000300  167460             MOV     #300,@RTCPSW    ;NO MULT INTERUPTS
3244 013220  052777  000100  167446             BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
3245 013226  052777  000004  167404  1$:        BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
3246 013234  005000                             CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
     013236  012701  000002                     MOV     #2,R1           ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                                                        ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                                                        ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
     013242  105777  167370             42$:    TSTB    @RBUF           ;** READ TO CLEAR DONE
     013246  005300                             DEC     R0              ;**
     013250  001374                             BNE     42$             ;**
     013252  005301                             DEC     R1              ;**
     013254  001372                             BNE     42$             ;**
3247 013256  052777  000100  167354             BIS     #BIT6,@TCSR     ;ENABLE TRANSMIT INTERRUPTS
3248 013264  052777  000100  167342             BIS     #BIT6,@RCSR     ;ENABLE RECEIVE INTERRUPTS
3249 013272  005037  002442                     CLR     XMTCNT          ;CLEAR XMIT INTERRUPT COUNTER
3250 013276  005037  002440                     CLR     RCVCNT          ;CLEAR RCV INTERRUPT COUNTER
3251 013302  005001                             CLR     R1              ;CLEAR A REGISTER FOR TEST DATA USE
3252 013304  005000                             CLR     R0              ;CLEAR TIMER
3253 013306  012702  002454                     MOV     #BUF,R2         ;POINT R2 TO RECEIVE DATA STORAGE
3254 013312  005077  167324                     CLR     @TBUF           ;SEND FIRST CHARACTER
3255 013316  004737  014502                     JSR     PC,@RPSW        ;SET PSW TO PRIORITY 3
3256 013322  000140                             .WORD   140
3257                                                                    ;WAIT FOR INTERRUPTS
3258 013324  000240                 2$:         NOP                     ;STALL
3259 013326  000240                             NOP                     ;STALL
3260 013330  062700  000000                     ADD     #0,R0           ;ADD INSTRUCTIONS ARE USED TO LENGTHEN LOOP TIME
3261 013334  062700  000001                     ADD     #1,R0           ;  TO COVER THE SLOWEST BAUD RATE ON THE FASTEST CPU
3262 013340  001371                             BNE     2$
3263 013342  032777  000100  167270             BIT     #BIT6,@TCSR     ;FINISHED ENTIRE TRANSMISSION
3264 013350  001402                             BEQ     3$              ;BR, IF INTERRPUTS ARE DISABLED (FINISHED)
3265 013352  000005                             RESET                   ;CLEAR EVERYTHING
3266
3267 013354  104107                             ERROR   +107            ;TRANSMIT INTERRUPT TIMEOUT IN MAIN, DATA TEST
```

```
3268
3269 013356  023737  002442  002440  3$:     CMP     XMTCNT,RCVCNT   ;COMPARE THE NUMBER OF INTERRUPTS
3270 013364  001402                          BEQ     4$              ;BR, IF EQUAL
3271 013366  000005                          RESET                   ;CLEAR EVERYTHING
3272
3273 013370  104110                          ERROR   •110            ;RECEIVER DID NOT GET FULL TRANSMISSION
3274                                                                  ;    IF RCVCNT=0, NO DATA RECEIVED
3275                                                                  ;    IF RCVCNT=?, THEN (XMTCNT-RCVCNT)
3276                                                                  ;    EQUALS THE NO. OF INTERRUPTS LOST.
3277 013372  005737  002624          4$:     TST     CTSTFL          ;IS CONCOLE UNDER TEST?
3278 013376  001411                          BEQ     5$              ;IF NOT, SKIP CLOCK COUNT CHECK
3279 013400  032777  000100  165432          BIT     #BIT6,@SWR      ;IF YES, ARE CLOCK TESTS DISABLED?
3280 013406  001005                          BNE     5$              ;IF YES, SKIP CLOCK COUNT CHECK
3281 013410  005737  002444                  TST     CLKCNT          ;CHECK FOR AT LESST ONE CLOCK INTERRUPT
3282 013414  001002                          BNE     5$              ;BR IF INTERRUPTS OCCURRED
3283 013416  000005                          RESET                   ;CLEAR MAINTENANCE WRAPAROUND
3284 013420  104113                          ERROR   •113            ;NO CLOCK INTERRUPTS IN EXERCISER
3285
3286 013422  000005          5$:     RESET                   ;CLEAR EVERYTHING
3287 013424  012700  002454                  MOV     #BUF,R0         ;LOAD RECEIVED DATA POINTER TO R0
3288 013430  005001                          CLR     R1              ;SET UP REGISTER FOR COMPARISON
3289 013432  022001          COMP:   CMP     (R0)+,R1        ;COMPARE XMIT & RCV DATA
3290 013434  001014                          BNE     6$              ;BR, IF NOT EQUAL
3291 013436  105201          9$:     INCB    R1              ;INCREMENT COMPARE DATA
3292 013440  032737  000001  003002          BIT     #BIT0,FLAG44    ;11/44 CPU?
3293 013446  001403                          BEQ     8$              ;YES, SKIP
3294 013450  122701  000020                  CMPB    #20,R1          ;SKIP 20 DATA IF 11/44
3295 013454  001770                          BEQ     9$              ;SKIP 20
3296 013456  032701  000040          8$:     BIT     #BIT5,R1        ;FINISHED CHECKING RECEIVED DATA?
3297 013462  001763                          BEQ     COMP            ;BR, IF NOT FINISHED
3298 013464  000405                          BR      7$              ;BR TO END OF TEST
3299
3300 013466  014037  001026          6$:     MOV     -(R0),$BDDAT    ;STORE BAD DATA FOR ERROR REPORT
3301 013472  010137  001024                  MOV     R1,$GDDAT       ;STORE GOOD DATA FOR ERROR REPORT
3302 013476  104111                          ERROR   •111            ;DATA COMPARE ERROR IN EXERCISER
3303
3304 013500  010377  167144          7$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
3305 013504  010477  167134                  MOV     R4,@RVECT       ;RESTORE RECEIVE VECTOR
3306 013510  010577  167162                  MOV     R5,@RTCVT       ;RESTORE CLOCK VECTOR
3307 013514  013777  002446  167130          MOV     STPSW,@TPSW     ;RESTORE XMIT PSW VECTOR
3308 013522  013777  002450  167116          MOV     SRPSW,@RPSW     ;RESTORE RECEIVE PSW VECTOR
3309 013530  013777  002452  167142          MOV     SCPSW,@RTCPSW   ;RESTORE CLOCK PSW VECTOR
3310 013536  000137  014272                  JMP     ENDEV           ;GOTO NEXT TEST
3311
3312 013542  005237  002442          XMIT:   INC     XMTCNT          ;INCREMENT XMIT INTERRUPT COUNTER
3313 013546  105201          1$:     INCB    R1              ;INCREMENT TEST DATA
3314 013550  032737  000001  003002          BIT     #BIT0,FLAG44    ;11/44 CPU
3315 013556  001403                          BEQ     2$              ;TEST ALL DATA
3316 013560  122701  000020                  CMPB    #20,R1          ;CHECK DATA FOR ↑P
3317 013564  001770                          BEQ     1$              ;DO NOT XMIT ↑P
3318 013566  032701  000040          2$:     BIT     #BIT5,R1        ;SEND DATA PATTERN FROM 00 --> 37
3319 013572  001404                          BEQ     XCONT           ;BR, IF MORE DATA TO BE SENT
3320 013574  042777  000100  167036          BIC     #BIT6,@TCSR     ;CLEAR XMIT INTERRUPT ENABLE
3321 013602  000402                          BR      XRET            ;RETURN, WITHOUT SENDING ANY MORE DATA
3322 013604  110177  167032          XCONT:  MOVB    R1,@TBUF        ;SEND NEW CHARACTER
3323 013610  005000          XRET:   CLR     R0              ;CLEAR TIMER
3324 013612  000002                          RTI                     ;RETURN
```

```
3325
3326 013614  017722  167016          RCV:    MOV     @RBUF,(R2)+     ;STORE RECEIVED DATA
3327 013620  005237  002440                  INC     RCVCNT          ;INCREMENT RCV INTERRUPT COUNTER
3328 013624  005000                          CLR     R0              ;CLEAR TIMER
3329 013626  000002                          RTI                     ;RETURN
3330
3331 013630  005237  002444          CLK:    INC     CLKCNT          ;INCREMENT CLOCK INTERRUPT COUNT
3332 013634  000002                          RTI                     ;RETURN
```

```
3333                                    .SBTTL  TEST # 46 - TEST CONSOLE WITH WRAP AROUND
                                        ;;*****************************************************************
                                        ;*TEST 46       TEST CONSOLE WITH WRAP AROUND
                                        ;;*****************************************************************
    013636  000004                      TST46:  SCOPE
                                        ;*****************************************************************************
3334                                    ;       WRAPAROUND TESTING- AUTO INITIATION OF T/A CONSOLE TEST
3335
3336
3337            176500                          RCSR58 = 176500
3338            176502                          RBUF58 = 176502
3339            176504                          TCSR58 = 176504
3340            176506                          TBUF58 = 176506
3341            003014                          BGNADD = DEVADR
3342            025252                          ENDADD = ENDADR
                                        ;*****************************************************************************
3343                                    ;
3344                                    ; MAIN LINE TEST
3345                                    ;
3346                                    ; THIS TEST PUTS COMMANDS OUT OVER THE SERIAL LINE WHICH IS WRAPPED
3347                                    ; AROUND TO THE CONSOLE AND RECEIVES THE RESPONSES
3348                                    ;
3349                                    ; CONTROL P IS SEND TO GET THE CONSOLE'S ATTENTION AND THEN
3350                                    ; T/A(A FOR APT) IS SENT.  SINCE THE CPU IS HALTED DURING THE TESTING
3351                                    ; THE PROGRAM CAN NOT SEE THE CHARS RETURNING, THUS THE PROGRAM WILL
3352                                    ; SIT IN A LOOP WAITING FOR A "B" TO BE PRINTED BY THE CONSOLE AS A
3353                                    ; SIGNAL TO APT THAT THE TESTING IS DONE. ALSO SINCE THE TESTING
3354                                    ; INVOLES WRITTING TO THE CPU'S MEMORY A CHECKSUM IS CALCULATED AND THE
3355                                    ; MEMORY TO BE USED INSIDE THIS PROGRAMS SPACE IS SAVE BEFORE TESTING
3356                                    ; AND RESTORED AFTER TEST WITH ANOTHER CHECKSUM CALCULATION
3357                                    ;
3358                                    ;*****************************************************************************
3359
3360
3361
3362 013640  032777  000004  165172  WRAP:   BIT     #BIT2,@SWR      ;RUN THIS TEST ONLY IF
3363 013646  001451                      BEQ     10$             ;SW BIT 2 IS ON
3364
3365 013650  013737  177776  002434          MOV     PSW,SAVEPS      ;SAVE OLD PSW
3366 013656  012737  000340  177776          MOV     #340,PSW        ;PUT IN NOW PSW
3367 013664  012706  002432                  MOV     #JIMSTK,SP      ;USE MY STACK (SO NO CLOBER)
3368
3369 013670  005037  002340                  CLR     LOC1
3370 013674  012737  000100  002342          MOV     #100,LOC2       ;TIMING LOOP COUNTERS
3371
3372 013702  004537  014202                  JSR     R5,SAVETE       ;SAVE LOCATIONS T/A WRITES
3373
3374 013706  004537  014010                  JSR     R5,CHKSUM       ;CALCULATE CHECKSUM
3375 013712  010437  002436                  MOV     R4,OLDSUM       ;SAVE OLD CHECKSUM
3376
3377 013716  012700  002564                  MOV     #CNTLP,R0
3378 013722  004537  014030                  JSR     R5,PUTLIN       ;SEND OUT CONTROL P
3379
3380 013726  012700  002566                  MOV     #PROMPT,R0
3381 013732  004537  014062                  JSR     R5,GETLIN       ;GET CRLF CONSOLE>>>
3382
3383 013736  012700  002610                  MOV     #TA,R0
3384 013742  004537  014030                  JSR     R5,PUTLIN       ;SEND OUT T/A<CRLF>
3385
```

```
3386 013746  004537  014126                    JSR     R5,GETB                 ;GET THE A FROM "-TESTB"
3387
3388 013752  004537  014236                    JSR     R5,RESTTE               ;RESTORE LOCATIONS T/A WRITES
3389
3390 013756  004537  014010                    JSR     R5,CHKSUM               ;RECALCULATE CHECKSUM
3391 013762  020437  002436                    CMP     R4,OLDSUM
3392 013766  001401                           BEQ     10$
3393 013770  000000                           HALT
3394 013772  012706  001000          10$:     MOV     #1000,SP                ;RETURN THEIR SP
3395 013776  013737  002434  177776          MOV     SAVEPS,PSW              ;RETURN PSW
3396 014004  000137  014326                    JMP     $EOP                    ;BR TO END OF PASS ROUTINE
3397
3398
3399                                ;*************************************************************************
3400                                ;
3401                                ; ROUTINE TO CALCULATE CHECKSUM ON PROGRAM
3402                                ;
3403                                ; INPUT CONDITIONS
3404                                ;
3405                                ;       BGNADD = ADDRESS TO START CHECK SUM (INCLUSIVE)
3406                                ;       ENDADD = ADDRESS TO END CHECK SUM (EXCLUSIVE)
3407                                ;
3408                                ; OUTPUT CONDITIONS
3409                                ;
3410                                ;       REG4 = CHECK SUM
3411                                ;
3412                                ;*************************************************************************
3413
3414
3415 014010  012700  003014          CHKSUM:  MOV     #BGNADD,R0              ;GET STARTING ADDRESS
3416 014014  005004                           CLR     R4                      ;RESET SUM
3417 014016  062004                  1$:      ADD     (R0)+,R4                ;ADD WORD TO SUM
3418 014020  022700  025252                    CMP     #ENDADD,R0              ;CHECK FOR END
3419 014024  001374                           BNE     1$                      ;IF NOT DONE LOOP
3420 014026  000205                           RTS     R5                      ;DONE RETURN
3421
3422
3423                                ;*************************************************************************
3424                                ;
3425                                ; THIS ROUTINE OUTPUTS TO THE SERIAL LINE CHARS STARTING AT
3426                                ; THE ADDRESS IN REG0 UNTIL IT HITS A <377>
3427                                ;
3428                                ;*************************************************************************
3429
3430
3431 014030  112001                  PUTLIN:  MOVB    (R0)+,R1                ;GET DATA
3432 014032  001412                           BEQ     10$                     ;END OF DATA
3433 014034  004537  014162          1$:      JSR     R5,TIMER                ;PROVIDE FOR TIMOUT
3434 014040  032737  000200  176504          BIT     #BIT7,TCSR58            ;TEST FOR XMIT READY
3435 014046  001772                           BEQ     1$                      ;WAIT FOR XMIT READY
3436 014050  110137  176506                    MOVB    R1,TBUF58               ;OUTPUT CHAR
3437 014054  000137  014030                    JMP     PUTLIN                  ;REPETE
3438 014060  000205                  10$:     RTS     R5                      ;RETURN
3439
3440                                ;*************************************************************************
3441                                ;
3442                                ; THIS ROUTINE INPUTS A CHARS FROM THE SERIAL LINE AND COMPARES
```

```
                                        ; IT WITH THE EXPECTED VALUES POINTED TO BY REG0 UNTIL NULL<00>
                                        ;
                                        ;*************************************************************************

                                        GETLIN:         MOVB    (R0)+,R1                ;GET EXPECTED CHAR
                                                        INCB    R1
                                                        BEQ     10$                     ;IF NULL EXIT
                                                        DECB    R1
                                        1$:             JSR     R5,TIMER                ;PROVIDE FOR TIMOUT
                                                        BIT     #BIT7,RCSR58            ;TEST FOR REC READY
                                                        BEQ     1$                      ;WAIT FOR REC READY
                                                        MOVB    RBUF58,R2
                                                        BIC     #BIT7,R2                ;STRIP PARITY
                                                        CMPB    R1,R2                   ;ARE THEY THE SAME
                                                        BEQ     GETLIN                  ;SAME GET MORE
                                                        HALT                            ;NOT SAME HALT
                                        10$:            RTS     R5                      ;RETURN

                                        ;*************************************************************************
                                        ;
                                        ; THIS ROUTINE INPUTS CHARS UNTIL IT GETS THE CHAR "B"
                                        ; AND THEN RETURNS
                                        ;
                                        ;*************************************************************************

                                        GETB:           JSR     R5,TIMER                ;OUT TIMING LOOP OUT
                                                        BIT     #BIT7,RCSR58            ;TEST OFR REC READY
                                                        BEQ     GETB                    ;NOT DONE YET
                                                        MOVB    RBUF58,R2
                                                        BIC     #BIT7,R2                ;STRIP PARITY
                                                        CMPB    #102,R2                 ;CHECK FOR "B"
                                                        BNE     GETB                    ;NOT "B" REPETE
                                                        RTS     R5                      ;WAS "B" RETURN

                                        ;*************************************************************************
                                        ;
                                        ; THIS ROUTINE IS USED AS A TIME OUT FEATURE
                                        ; WHEN THE TIMING LOOPS BOTH REACH 0 THIS ROUTINE WILL
                                        ; CAUSE A HALT
                                        ;
                                        ;*************************************************************************

                                        TIMER:          DEC     LOC1
                                                        BNE     10$                     ;DECREMPNT TIMING LOOPS
                                                        DEC     LOC2
                                                        BNE     10$
                                                        HALT                            ;IF ZERO THIS ROUTINE
                                        10$:            RTS     R5                      ;EXECUTED R3 TIMES

                                        ;*************************************************************************
                                        ;
                                        ; THIS ROUTINE SAVES THE LOCATIONS WRITTEN BY THE T/A CONSOLE TEST
```

```
3500                                    ; SO THEY MAY BE RESTORED LATER
3501                                    ;
3502                                    ;*****************************************************************************
3503
3504 014202  013737  000000  002344  SAVETE:         MOV     0,SAVE0                 ;SAVE LOCATION 0
3505 014210  012702  000002                          MOV     #2,R2                   ;SET UP INDIRECT PNTER
3506 014214  012701  002346                          MOV     #SAVLOC,R1              ;SET UP STORAGE LOC. PNTER
3507 014220  011221                  1$:             MOV     (R2),(R1)+              ;GET WORD AND SAVE
3508 014222  000241                                  CLC                             ;DONT ROT IN ANY BITS
3509 014224  006102                                  ROL     R2                      ;SET NEXT ADDRESS
3510 014226  020227  025252                          CMP     R2,#ENDADD              ;SEE IF AT END
3511 014232  100772                                  BMI     1$                      ;NO REPETE
3512 014234  000205                                  RTS     R5
3513
3514
3515                                    ;*****************************************************************************
3516                                    ;
3517                                    ; THIS ROUTINE RESTORES THE SAVED LOCATIONS THAT T/A WRITES INTO
3518                                    ;
3519                                    ;*****************************************************************************
3520
3521 014236  013737  002344  000000  RESTTE:         MOV     SAVE0,0
3522 014244  012702  000002                          MOV     #2,R2                   ;SET UP INDIRECT PNTER
3523 014250  012701  002346                          MOV     #SAVLOC,R1              ;SET UP STORAGE LOC. PNTER
3524 014254  012112                  1$:             MOV     (R1)+,(R2)              ;GET WORD AND RESTORE
3525 014256  000241                                  CLC                             ;DONT ROT IN ANY BITS
3526 014260  006102                                  ROL     R2                      ;SET NEXT ADDRESS
3527 014262  020227  025252                          CMP     R2,#ENDADD              ;SEF IF AT END
3528 014266  100772                                  BMI     1$                      ;NO REPETE
3529 014270  000205                                  RTS     R5
```

```
                                                        ;END OF DEVICE PASS ROUTINE
3530
3531 014272  005037  001002          ENDEV:  CLR     #TSTNM          ;CLEAR TEST NO. COUNT FOR SCOPE ROUTINE
3532 014276  005237  001076                  INC     #DEVCT          ;INCREMENT DEVICE COUNTER
3533 014302  023737  002630  001076          CMP     TMP2,#DEVCT     ;ALL DEVICES TESTED
3534 014310  001002                          BNE     NOEOP           ;BR, IF NO
3535 014312  000137  013640                  JMP     WRAP            ;EXECUTE WRAP AROUND AFTER ALL DEVICES
3536 014316  005037  002624          NOEOP:  CLR     CTSTFL          ;CLEAR CONSOLE UNDER TEST FLAG
3537 014322  000137  004050                  JMP     TSTDEV          ;GO TEST NEXT DEVICE
```

```
                        .SBTTL  END OF PASS ROUTINE
                        ;;**********************************************************************
                        ;*INCREMENT THE PASS NUMBER ($PASS)
                        ;*IF THERES A MONITOR GO TO IT
                        ;*IF THERE ISN'T JUMP TO GOAGIN
014326                  $EOP:
014326 000004                   SCOPE
014330 005037 001002            CLR     $TSTNM          ;;ZERO THE TEST NUMBER
014334 005237 001074            INC     $PASS           ;;INCREMENT THE PASS NUMBER
014340 042737 100000 001074     BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
014346 005327                   DEC     (PC)+           ;;LOOP?
014350 000001           $EOPCT: .WORD   1
014352 003015                   BGT     $DOAGN          ;;YES
014354 012737                   MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
014356 000001           $ENDCT: .WORD   1
014360 014350                   $EOPCT
014362 104401 014416            TYPE    ,ENDMG          ;;TYPE "END PASS"
014366 013700 000042    $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
014372 001405                   BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
014374 000005                   RESET                   ;;CLEAR THE WORLD
014376 004710           $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
014400 000240                   NOP                     ;;SAVE ROOM
014402 000240                   NOP                     ;;FOR
014404 000240                   NOP                     ;;ACT11
014406                  $DOAGN:
014406 000137                   JMP     @(PC)+          ;;RETURN
014410 014432           $RTNAD: .WORD   GOAGIN
014412    377    377    000    $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                .EVEN
3540 014416    015    012    105 ENDMG:  .ASCIZ  <CR><LF>/END PASS /
```

END OF PASS ROUTINE

```
3542 014432 005046                  GOAGIN: CLR     -(SP)           ;CLEAR ANOTHER LOCATION ON STACK
3543 014434 005216                  1$:     INC     (SP)            ;INCRIMENT STACK LOCATION
3544 014436 000240                          NOP                     ;TAKE UP SOME MORE TIME
3545 014440 001375                          BNE     1$              ;BRANCH BACK UNTIL ZERO AGAIN
3546 014442 062706 000002                   ADD     #2,SP           ;CLEAN THE LOCATION OFF THE STACK
3547 014446 005037 001076                   CLR     $DEVCT          ;CLEAR DEVICE COUNT
3548 014452 022737 000001 002630            CMP     #1,TMP2         ;IS THERE ONLY ONE DEVICE UNDER TEST?
3549 014460 001004                          BNE     RSTRT           ;BR, IF NOT
3550 014462 012706 001000                   MOV     #1000,SP                ;RESET STACK POINTER
3551 014466 000137 004172                   JMP     TST1            ;GO DO ANOTHER PASS
3552
3553 014472 005037 001100           RSTRT:  CLR     $UNIT           ;CLEAR UNIT NUMBER
3554 014476 000137 003774                   JMP     BEGIN
3555
3556 014502 011646                  WRPSW:  MOV     (SP),-(SP)      ;COPY RETURN PC
3557 014504 017666 000002 000002            MOV     @2(SP),2(SP)    ;MOVE NEW PSW TO STACK
3558 014512 062716 000002                   ADD     #2,(SP)         ;ADJUST JSR RETURN OVER PRIORITY REQUESTED
3559 014516 000002                          RTI                     ;POP RETURN PC & NEW PSW
3560
3561                                ;SUBROUTINE TO REPORT UNEXPECTED OR ERRONEOUS TRAPS OR INTERRUPTS
3562
3563 014520 012600                  CATCH:  MOV     (SP)+,R0        ;GET ADDRESS OF TRAP VECTOR + 4
3564 014522 162700 000004                   SUB     #4,R0           ;ADJUST TO POINT TO TRAP ADDRESS
3565 014526 010037 002622                   MOV     R0,BDVECT       ;STORE TRAP OR INTERRUPT ADDRESS
3566 014532 016637 000002 002620            MOV     2(SP),OLDPC     ;GET PC WHERE TRAP OR INTERRUPT OCCURRED
3567 014540 104112                          ERROR   +112            ;REPORT ERROR
3568
3569 014542 000000                          HALT                    ;PROGRAM MUST BE RESTARTED AT THIS POINT
```

END OF PASS ROUTINE

```
3571
3572                                    ;;**********************************************************************
3573                                    ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3574                                    ;*SAVE THE ERROR ITEM NUMBER AND ADDRESS OF THE ERROR CALL
3575                                    ;*AND GO TO $ERRTYP ON ERROR
3576                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3577                                    ;*SW15=1         HALT ON ERROR
3578                                    ;*SW13=1         INHIBIT ERROR TYPEOUTS
3579                                    ;*SW09=1         LOOP IN ERROR
3580                                    ;*CALL
3581                                    ;*      ERROR   .N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER
3582                                    ;;**********************************************************************
3583
3584 014544                            $ERROR:
3585 014544  105237  001003            7$:     INCB    $ERFLG          ;SET THE ERROR FLAG
3586 014550  001775                            BEQ     7$              ;DON'T LET FLAG GO TO ZERO
3587 014552  013777  001002  164262            MOV     $TSTNM,@DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
3588 014560  005237  001012                    INC     $ERTTL          ;INCREMENT ERROR COUNT
3589 014564  011637  001016                    MOV     (SP),$ERRPC     ;GET ADDRESS OF ERROR INSTRUCTION
3590 014570  162737  000002  001016            SUB     #2,$ERRPC
3591 014576  117737  164214  001014            MOVB    @$ERRPC,$ITEMB  ;STRIP AND SAVE THE ERROR ITEM CODE
3592 014604  032777  020000  164226            BIT     #BIT13,@SWR     ;SKIP TYPEOUT IF SET
3593 014612  001004                            BNE     20$             ;SKIP TYPEOUTS
3594 014614  004737  014726                    JSR     PC,$ERRTYP      ;GO TO USER ERROR ROUTINE
3595 014620  104401  001063                    TYPE    ,$CRLF
3596 014624                            20$:
3597 014624  122737  000001  001106            CMPB    #APTENV,$ENV    ;RUNNING IN APT MODE
3598 014632  001007                            BNE     2$              ;NO, SKIP APT ERROR REPORT
3599 014634  113737  001014  014646            MOVB    $ITEMB,21$      ;SET ITEM NUMBER AS ERROR NUMBER
3600 014642  004737  0157'4                    JSR     PC,$ATY4        ;REPORT FATAL ERROR TO APT
3601 014646     000                    21$:    .BYTE   0
3602 014647     000                            .BYTE   0
3603 014650  000777                    22$:    BR      22$             ;APT ERROR LOOP
3604 014652  005777  164162            2$:     TST     @SWR            ;HALT ON ERROR
3605 014656  100001                            BPL     3$              ;SKIP IF CONTINUE
3606 014660  000000                            HALT                    ;HALT ON ERROR!
3607 014662  104406                    3$:     CKSWR                   ;TEST FOR CHANGE IN SOFT-SWR
3608 014664  032777  001000  164146            BIT     #BIT09,@SWR     ;LOOP ON ERROR SWITCH SET?
3609 014672  001402                            BEQ     4$              ;BR IF NO
3610 014674  013716  001010                    MOV     $LPERR,(SP)     ;FUDGE RETURN FOR LOOPING
3611 014700  005737  001060            4$:     TST     $ESCAPE         ;CHECK FOR AN ESCAPE ADDRESS
3612 014704  001402                            BEQ     5$              ;BR IF NONE
3613 014706  013716  001060                    MOV     $ESCAPE,(SP)    ;FUDGE RETURN ADDRESS FOR ESCAPE
3614 014712                            5$:
3615 014712  022737  014376  000042            CMP     #$ENDAD,@#42    ;ACT-11 AUTO-ACCEPT?
3616 014720  001001                            BNE     6$              ;BR IF NO
3617 014722  000000                            HALT                    ;YES
3618 014724                            6$:
3619 014724  000002                            RTI                     ;RETURN
3620
```

```
 3622
 3623                                   .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

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

014726                                  $ERRTYP:
014726  104401  001063                          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
014732  010046                                MOV     R0,-(SP)        ;;SAVE R0
014734  005000                                CLR     R0              ;;PICKUP THE ITEM INDEX
014736  153700  001014                        BISB    @#$ITEMB,R0
014742  001004                                BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                        ;;TYPE THE PC OF THE ERROR
014744  013746  001016                        MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                        ;;ERROR ADDRESS
014750  104402                                TYPOC                   ;;GO TYPE- OCTAL ASCII(ALL DIGITS)
014752  000434                                BR      6$              ;;GET OUT
014754  122700  000177          1$:           CMPB    #177,R0         ;IS THIS THE POWER MONITOR BIT CALL     ;DPM001
014760  001003                                BNE     100$            ;BRANCH IF NOT                          ;DPM001
014762  012700  015076                        MOV     #PFECWS,R0      ;MOVE ADDRS OF PWR MONITOR BIT ERR TO R0;DPM001
014766  000406                                BR      110$            ;BRANCH TO CALL THE ERROR               ;DPM001
014770  005300                  100$:         DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
014772  006300                                ASL     R0              ;;      WORK FOR THE ERROR TABLE
014774  006300                                ASL     R0
014776  006300                                ASL     R0
015000  062700  001146                        ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
015004  012037  015014          110$:         MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
015010  001404                                BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
015012  104401                                TYPE                    ;;TYPE THE "ERROR MESSAGE"
015014  000000                  2$:           .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
015016  104401  001063                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
015022  012037  015032          3$:           MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
015026  001404                                BEQ     5$              ;;SKIP TYPEOUT IF 0
015030  104401                                TYPE                    ;;TYPE THE "DATA HEADER"
015032  000000                  4$:           .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
015034  104401  001063                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
015040  011000                  5$:           MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
015042  001004                                BNE     7$              ;;GO TYPE THE DATA
015044  012600                  6$:           MOV     (SP)+,R0        ;;RESTORE R0
015046  104401  001063                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
015052  000207                                RTS     PC              ;;RETURN
015054                                  7$:
```

```
015054  013046                          MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
015056  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
015060  005710                          TST     (R0)            ;;IS THERE ANOTHER NUMBER?
015062  001770                          BEQ     6$              ;;BR IF NO
015064  104401  015072                  TYPE    ,8$             ;;TYPE TWO(2) SPACES
015070  000771                          BR      7$              ;;LOOP
015072     040     040     000  8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
                                        .EVEN
015076  015106  015146  015176  PFECWS: .WORD   PFECEM,PFECDH,PFECDT,PFECDF
015106     120     117     127  PFECEM: .ASCIZ  ?POWER MONITOR BIT WAS FOUND SET?
015146     124     105     123  PFECDH: .ASCIZ  ?TESTNO  ERR PC  CPUERR?
                                        .EVEN
015176  001072  001016  015674  PFECDT: .WORD   $TESTN,$ERRPC,CPSAVE,0
015206     000     000     000  PFECDF: .BYTE   0,0,0,0
3624
```

```
3626
3627                                    .SBTTL  POWER DOWN AND UP ROUTINES
3628                                    ;;**********************************************************************
3629                                            ;*POWER DOWN ROUTINE
3630                                    ;;**********************************************************************
3631 015212  012737  015356  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;SET FOR FAST UP
3632 015220  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;PRIU:7
3633 015226  010046                          MOV     R0,-(SP)        ;PUSH R0 ON STACK
3634 015230  010146                          MOV     R1,-(SP)        ;PUSH R1 ON STACK
3635 015232  010246                          MOV     R2,-(SP)        ;PUSH R2 ON STACK
3636 015234  010346                          MOV     R3,-(SP)        ;PUSH R3 ON STACK
3637 015236  010446                          MOV     R4,-(SP)        ;PUSH R4 ON STACK
3638 015240  010546                          MOV     R5,-(SP)        ;PUSH R5 ON STACK
3639 015242  017746  163572                  MOV     @SWR,-(SP)      ;PUSH @SWR ON STACK
3640 015246  010637  015362                  MOV     SP,$SAVR6       ;SAVE SP
3641 015252  012737  015264  000024          MOV     #$PWRUP,@#PWRVEC ;SET UP VECTOR
3642 015260  000000                          HALT
3643 015262  000776                          BR      .-2             ;HANG UP
3644
3645
3646                                    ;;**********************************************************************
3647                                            ;*POWER UP ROUTINE
3648                                    ;;**********************************************************************
3649 015264  012737  015356  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;SET FOR FAST DOWN
3650 015272  013706  015362                  MOV     $SAVR6,SP       ;GET SP
3651 015276  012677  163536                  MOV     (SP)+,@SWR      ;POP STACK INTO @SWR
3652 015302  012605                          MOV     (SP)+,R5
3653 015304  012604                          MOV     (SP)+,R4        ;POP STACK INTO R4
3654 015306  012603                          MOV     (SP)+,R3        ;POP STACK INTO R3
3655 015310  012602                          MOV     (SP)+,R2        ;POP STACK INTO R2
3656 015312  012601                          MOV     (SP)+,R1        ;POP STACK INTO R1
3657 015314  012600                          MOV     (SP)+,R0        ;POP STACK INTO R0
3658 015316  012737  015212  000024          MOV     #$PWRDN,@#PWRVEC ;SET UP THE POWER DOWN VECTOR
3659 015324  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;PRIO:7
3660 015332  005037  015362                  CLR     $SAVR6          ;WAIT LOOP FOR THE TTY
3661 015336  005237  015362          1$:     INC     $SAVR6          ;WAIT FOR THE INC
3662 015342  001375                          BNE     1$              ;OF WORD
3563 015344  104401                          TYPE                    ;REPORT THE POWER FAILURE
3664 015346  015364                  $PWRMG: .WORD   $POWER          ;POWER FAIL MESSAGE POINTER
3665 015350  013716  001006                  MOV     $LPADR,(SP)     ;CHOCOLATE FUDGE RETURN TO BEGINNING OF TEST
3666 015354  000002                          RTI                     ;RETURN TO THERE
3667 015356  000000                  $ILLUP: HALT                    ;THE POWER UP SEQUENCE WAS STARTED
3668 015360  000776                          BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
3669 015362  000000                  $SAVR6: 0                       ;PUT THE SP HERE
3670 015364     015     012     120  $POWER: .ASCIZ  <15><12>"POWER"
3671
```

```
3673
3674                                    .SBTTL  SCOPE HANDLER ROUTINE

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

015374                                  @SCOPE:
015374  104406                                 CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
015376  032777  040000  163434          1@:    BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
015404  001125                                 BNE     @OVER           ;;YES IF SW14=1
                                        ;*****START OF CODE FOR THE XOR TESTER*****
015406  000416                          @XTSTR: BR      6@              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
015410  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
015414  012737  015434  000004                  MOV     #5@,@#ERRVEC    ;;SET FOR TIMEOUT
015422  005737  177060                          TST     @#177060        ;;TIME OUT ON XOR?
015426  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
015432  000474                                  BR      @SVLAD          ;;GO TO THE NEXT TEST
015434  022626                          5@:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
015436  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
015442  000462                                  BR      7@              ;;LOOP ON THE PRESENT TEST
015444                                  6@:;*****END OF CODE FOR THE XOR TESTER*****
015444  022737  177777  015674          2@:    CMP     #-1,CPSAVE      ;SEE IF TIMEOUT WAS PREVIOUSLY RECORDED ;DPM001
015452  001447                                 BEQ     2001@           ;BRANCH AROUND ROUTINE IF SO            ;DPM001
015454  005227  177777                         INC     #-1             ;TEST FOR 1ST TIME THROUGH              ;DPM001
015460  001005                                 BNE     900@            ;BRANCH IF NOT                          ;DPM001
015462  013746  000004                         MOV     4,-(SP)         ;SAVE TIMEOUT VECTOR AT 4               ;DPM001
015466  012737  015542  000004                 MOV     #1999@,4        ;SET VECTOR TO LOCATION BELOW           ;DPM001
015474  013737  177766  015674  900@:    MOV     177766,CPSAVE   ;MOVE CPU ERR REG VALUE TO LOC FOR TST  ;DPM001
015502  032737  000001  015674                 BIT     #BIT00,CPSAVE   ;SEE IF THE POWER MONITOR BIT IS ON     ;DPM001
015510  001423                                 BEQ     2000@           ;BRANCH TO CONTINUE ROUTINE IF CLEAR    ;DPM001
015512  042737  000001  177766                 BIC     #BIT00,177766   ;CLEAR THE BIT FOUND TO BE SET        ;DPM001
015520  017746  163314                         MOV     @SWR,-(SP)      ;SAVE SWR VALUE                         ;DPM001
015524  042777  001000  163306                 BIC     #BIT09,@SWR     ;DON'T ALLOW LOOP ON ERROR ON THIS ERROR;DPM001
015532  104177                                 EMT     177             ;CALL SPECIAL POWER FAIL BIT ERROR CALL ;DPM001
015534  012677  163300                         MOV     (SP)+,@SWR      ;RESTORE SWR TO ORIGINAL VALUE          ;DPM001
015540  000407                                 BR      2000@           ;BRANCH OVER TIMEOUT SECTION            ;DPM001
015542  022626                          1999@:  CMP     (SP)+,(SP)+     ;CLEAN UP THE STACK AFTER TIMEOUT        ;DPM001
015544  012737  177777  015674                 MOV     #-1,CPSAVE      ;MOVE -1 TO CPSAVE - NO CPU ERR REG     ;DPM001
015552  005237  015562                         INC     2000@+2         ;INCREMENT 1ST TIME THROUGH LOCATION    ;DPM001
015556  000403                                 BR      910@            ;BRANCH OVER 1ST TIME THROUGH TEST      ;DPM001
015560  005227  177777                 2000@:  INC     #-1             ;TEST FOR 1ST TIME THROUGH              ;DPM001
015564  001002                                 BNE     2001@           ;BRANCH IF NOT                          ;DPM001
015566  012637  000004                 910@:   MOV     (SP)+,4         ;RESTORE LOCATION 4                     ;DPM001
015572  105737  001003                 2001@:  TSTB    @ERFLG          ;;HAS AN ERROR OCCURRED?
015576  001412                                 BEQ     @SVLAD          ;;BR IF NO
015600  032777  001000  1o3232                 BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
015606  001404                                 BEQ     4@              ;;BR IF NO
015610  013737  001010  001006          7@:    MOV     @LPERR,@LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
015616  000420                                 BR      @OVER
```

```
        015620  105037  001003          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
        015624  105237  001002          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
        015630  113737  001002  001072          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER"IN APT MAILBOX
        015636  011637  001006                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
        015642  011637  001010                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
        015646  005037  001060                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
        015652  112737  000001  001015          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
        015660  013777  001002  163154  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
        015666  013716  001006                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
        015672  000002                          RTI                     ;;FIXES PS
        015674  000000                  CPSAVE: .WORD   0               ;LOCATION TO SAVE CPU ERR REG CONTENTS  ;DPM00!
  3675
```

```
3677
3678                                    ;;*****************************************************************
3679                                    .SBTTL  APT COMMUNICATIONS ROUTINE
3680                                    ;;*****************************************************************
3681
3682 015676  112737  000001  016142  $ATY1:  MOVB    #1,$FFLG        ;TO REPORT FATAL ERROR
3683 015704  112737  000001  016140  $ATY3:  MOVB    #1,$MFLG        ;TO TYPE A MESSAGE
3684 015712  000403                        BR      $ATYC
3685 015714  112737  000001  016142  $ATY4:  MOVB    #1,$FFLG        ;TO ONLY REPORT FATAL ERROR
3686 015722                        $ATYC:
3687 015722  010046                        MOV     R0,-(SP)        ;PUSH R0 ON STACK
3688 015724  010146                        MOV     R1,-(SP)        ;PUSH R1 ON STACK
3689 015726  105737  016140                TSTB    $MFLG           ;SHOULD TYPE A MESSAGE?
3690 015732  001450                        BEQ     5$              ;IF NOT: BR
3691 015734  122737  000001  001106        CMPB    #APTENV,$ENV    ;OPERATING UNDER APT?
3692 015742  001031                        BNE     3$              ;IF NOT: BR
3693 015744  132737  000100  001107        BITB    #APTSPOOL,$ENVM ;SHOULD SPOOL MESSAGE?
3694 015752  001425                        BEQ     3$              ;IF NOT: BR
3695 015754  017600  000004                MOV     @4(SP),R0       ;GET MESSAGE ADDRESS
3696 015760  062766  000002  000004        ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
3697 015766  005737  001066          1$:   TST     $MSGTYPE                ;SEE IF DONE W/ LAST XMISSION?
3698 015772  001375                        BNE     1$              ;IF NOT: WAIT
3699 015774  010037  001102                MOV     R0,$MSGAD       ;PUT ADDRESS IN MAILBOX
3700 016000  105720                  2$:   TSTB    (R0)+           ;FIND END OF MESSAGE
3701 016002  001376                        BNE     2$
3702 016004  163700  001102                SUB     $MSGAD,R0       ;SUB START OF MESSAGE
3703 016010  006200                        ASR     R0              ;GET MESSAGE LENGTH IN WORDS
3704 016012  010037  001104                MOV     R0,$MSGLGT      ;PUT LENGTH IN MAILBOX
3705 016016  012737  000004  001066        MOV     #4,$MSGTYPE     ;TELL APT TO TAKE MESSAGE
3706 016024  000413                        BR      5$
3707 016026  017637  000004  016052  3$:   MOV     @4(SP),4$       ;PUT MSG ADDR IN JSR LINKAGE
3708 016034  062766  000002  000004        ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
3709 016042  013746  177776                MOV     177776,-(SP)    ;PUSH 177776 ON STACK
3710 016046  004737  016144                JSR     PC,$TYPE        ;CALL TYPE MACRO
3711 016052  000000                  4$:   .WORD   0
3712 016054                        5$:
3713 016054  105737  016142          10$:  TSTB    $FFLG           ;SHOULD REPORT FATAL ERROR?
3714 016060  001413                        BEQ     12$             ;IF NOT: BR
3715 016062  005737  001106                TST     $ENV            ;RUNNING UNDER APT?
3716 016066  001410                        BEQ     12$             ;IF NOT: BR
3717 016070  005737  001066          11$:  TST     $MSGTYPE        ;FINISHED LAST MESSAGE?
3718 016074  001375                        BNE     11$             ;IF NOT: WAIT
3719 016076  017637  000004  001070        MOV     @4(SP),$FATAL   ;GET ERROR #
3720 016104  005237  001066                INC     $MSGTYPE        ;TELL APT TO TAKE ERROR
3721 016110  062766  000002  000004  12$:   ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
3722 016116  105037  016142                CLRB    $FFLG           ;CLEAR FATAL FLAG
3723 016122  105037  016141                CLRB    $LFLG           ;CLEAR LOG FLAG
3724 016126  105037  016140                CLRB    $MFLG           ;CLEAR MESSAGE FLAG
3725 016132  012601                        MOV     (SP)+,R1        ;POP STACK INTO R1
3726 016134  012600                        MOV     (SP)+,R0        ;POP STACK INTO R1
3727 016136  000207                        RTS     PC              ;RETURN
```

```
3728 016140  000                $MFLG:  .BYTE   0
3729 016141  000                $LFLG:  .BYTE   0               ;LOG FLAG
3730 016142  000                $FFLG:  .BYTE   0               ;FATAL FLAG
3731
3732                                    .EVEN
3733          000200            APTSIZE=200
3734          000001            APTENV=001
3735          000100            APTSPOOL=100
3736          000040            APTCSUP=040
3737
```

```
3739
3740                                    .SBTTL  TYPE ROUTINE
                                        ;;*****************************************************************
                                        ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                        ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                        ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                        ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                        ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                        ;*
                                        ;*CALL:
                                        ;*1) USING A TRAP INSTRUCTION
                                        ;*       TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                        ;*OR
                                        ;*       TYPE
                                        ;*       MESADR
                                        ;*
016144  105737  001057          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
016150  100002                          BPL     1$              ;;BR IF YES
016152  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
016154  000430                          BR      3$              ;;LEAVE
016156  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
016160  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
016164  122737  000001  001106          CMPB    @APTENV,$ENV    ;;RUNNING IN APT MODE
016172  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
016174  132737  000100  001107          BITB    @APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
016202  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
016204  010037  016214                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
016210  004737  015704                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
016214  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
016216  132737  000040  001107  62$:    BITB    @APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
016224  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
016226  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
016230  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
016232  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
016234  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
016236  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
016242  000002                          RTI                     ;;RETURN
016244  122716  000011          4$:     CMPB    @#11,(SP)       ;;BRANCH IF <HT>
016250  001430                          BEQ     8$
016252  122716  000200                  CMPB    @#CRLF,(SP)     ;;BRANCH IF NOT <CRLF>
016256  001006                          BNE     5$
016260  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
016262  104401                          TYPE                    ;;TYPE A CR AND LF
016264  001063                          $CRLF
016266  105037  016504                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
016272  000755                          BR      2$              ;;GET NEXT CHARACTER
016274  004737  016356          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
016300  123726  001056          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
016304  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
016306  013746  001054                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
016312  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
016316  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
016320  004737  016356                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
016324  105337  016504                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
016330  000770                          BR      7$              ;;LOOP
                                        ;HORIZONTAL TAB PROCESSOR
```

```
016332  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
016336  004737  016356          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
016342  132737  000007  016504          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
016350  001372                          BNE     9$              ;;TAB STOP
016352  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
016354  000724                          BR      2$              ;;GET NEXT CHARACTER
016356                          $TYPEC:
016356  105777  162462                  TSTB    @#TKS           ;;CHAR IN KYBD BUFFER?                  ;MJD001
016362  100022                          BPL     10$             ;;BR IF NOT                             ;MJD001
016364  017746  162456                  MOV     @#TKB,-(SP)     ;;GET CHAR                              ;MJD001
016370  042716  177600                  BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS                 ;MJD001
016374  122716  000023                  CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF                         ;MJD001
016400  001012                          BNE     102$            ;;BR IF NOT                            ;MJD001
016402                          101$:                                                                   ;MJD001
016402  105777  162436                  TSTB    @#TKS           ;;WAIT FOR CHAR                         ;MJD001
016406  100375                          BPL     101$                                                    ;MJD001
016410  117716  162432                  MOVB    @#TKB,(SP)      ;;GET CHAR                              ;MJD001
016414  042716  177600                  BIC     #177600,(SP)    ;;STRIP IT                              ;MJD001
016420  122716  000021                  CMPB    #$XON,(SP)      ;;WAS IT XON?                           ;MJD001
016424  001366                          BNE     101$            ;;BR IF NOT                            ;MJD001
016426                          102$:                                                                   ;MJD001
016426  005726                          TST     (SP)+           ;;FIX STACK                             ;MJD001
016430                          10$:                                                                    ;MJD001
016430  105777  162414                  TSTB    @#TPS           ;;WAIT UNTIL PRINTER IS READY
016434  100375                          BPL     10$                                                     ;MJD001
016436  126627  000002  000021          CMPB    2(SP),#$XON     ;;IS CHARACTER A RANDOM XON?            ;RAN001
016444  001420                          BEQ     $TYPEX          ;;BRANCH IF YES                         ;RAN001
016446  116677  000002  162376          MOVB    2(SP),@#TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
016454  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
016462  001003                          BNE     1$              ;;BRANCH IF NO
016464  105037  016504                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
016470  000406                          BR      $TYPEX          ;;EXIT
016472  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
016500  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
016502  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
016504  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
016506  000207                  $TYPEX: RTS     PC
3741                            .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
                                ;;**************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
```

BINARY TO OCTAL (ASCII) AND TYPE

```
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT
016510  017646  000000                  $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
016514  116637  000001  016733                  MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
016522  112637  016735                          MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
016526  062716  000002                          ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
016532  000406                          BR      $TYPON
016534  112737  000001  016733          $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
016542  112737  000006  016735                  MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
016550  112737  000005  016732          $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
016556  010346                          MOV     R3,-(SP)        ;;SAVE R3
016560  010446                          MOV     R4,-(SP)        ;;SAVE R4
016562  010546                          MOV     R5,-(SP)        ;;SAVE R5
016564  113704  016735                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
016570  005404                          NEG     R4
016572  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
016576  110437  016734                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
016602  113704  016733                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
016606  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
016612  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
016614  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
016616  000404                          BR      3$              ;;GO DO MSB
016620  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
016622  006105                          ROL     R5
016624  006105                          ROL     R5
016626  010503                          MOV     R5,R3
016630  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
016632  105337  016734                  DECB    $OMODE          ;;TYPE THIS DIGIT?
016636  100016                          BPL     7$              ;;BR IF NO
016640  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
016644  001002                          BNE     4$              ;;TEST FOR 0
016646  005704                          TST     R4              ;;SUPPRESS THIS 0?
016650  001403                          BEQ     5$              ;;BR IF YES
016652  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
016654  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
016660  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
016664  110337  016730                  MOVB    R3,8$           ;;SAVE FOR TYPING
016670  104401  016730                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
016674  105337  016732          7$:     DECB    $OCNT           ;;COUNT BY 1
016700  003347                          BGT     2$              ;;BR IF MORE TO DO
016702  002402                          BLT     6$              ;;BR IF DONE
016704  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
016706  000744                          BR      2$              ;;GO DO THE LAST DIGIT
016710  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
016712  012604                          MOV     (SP)+,R4        ;;RESTORE R4
016714  012603                          MOV     (SP)+,R3        ;;RESTORE R3
016716  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
016724  012616                          MOV     (SP)+,(SP)
016726  000002                          RTI                     ;;RETURN
016730     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
016731     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
016732     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
016733     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
016734  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
3742
```

```
3744
3745                                    .SBTTL  TTY INPUT ROUTINE
                                        ;;*****************************************************************
                                        .ENABL  LSB
                                        ;;*****************************************************************
                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                        ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;*WHEN OPERATING IN TTY FLAG MODE.
016736  022737  000176  001040  $CKSWR: CMP     @SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
016744  001074                          BNE     15$             ;;BRANCH IF NO
016746  105777  162072                  TSTB    @$TKS           ;;CHAR THERE?
016752  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
016754  117746  162066                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
016760  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
016764  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
016770  001062                          BNE     15$             ;;NO, RETURN TO USER
016772  123727  001034  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
017000  001456                          BEQ     15$             ;;BRANCH IF YES
017002  104401  017473                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
017006  104401  017500          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
017012  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
017016  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
017020  104401  017511                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
017024  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
017026  005046                          CLR     -(SP)           ;;THE NEW SWR
017030  105777  162010          7$:     TSTB    @$TKS           ;;CHAR THERE?
017034  100375                          BPL     7$              ;;IF NOT TRY AGAIN
017036  117746  162004                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
017042  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
017046  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
017052  001005                          BNE     10$             ;;BRANCH IF NOT
017054  104401  017466                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
017060  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
017064  000757                          BR      19$             ;;LET'S TRY IT AGAIN
017066  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
017072  001022                          BNE     16$             ;;BRANCH IF NO
017074  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
017100  001403                          BEQ     11$             ;;BRANCH IF YES
017102  016677  000002  161730          MOV     2(SP),@SWR      ;;SAVE NEW SWR
017110  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
017114  104401  001063          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
017120  123727  001035  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
017126  001003                          BNE     15$             ;;BRANCH IF NOT
017130  012777  000100  161706          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
017136  000002                  15$:    RTI                     ;;RETURN
017140  004737  016356          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
017144  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
017150  002420                          BLT     18$             ;;BRANCH IF YES
017152  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
017156  003015                          BGT     18$             ;;BRANCH IF YES
017160  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
017164  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
017170  001403                          BEQ     17$             ;;BRANCH IF YES
017172  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
017174  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
017176  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
```

TTY INPUT ROUTINE

```
017200  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
017204  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
017210  000707                          BR      7$              ;;GET THE NEXT ONE
017212  104401  001062          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
017216  000720                          BR      20$             ;;SIMULATE CONTROL-U
                                .DSABL  LSB
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                ;
017220  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
017222  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
017230  105777  161610          1$:     TSTB    @#TKS           ;;WAIT FOR
017234  100375                          BPL     1$              ;;A CHARACTER
017236  117766  161604  000004          MOVB    @#TKB,4(SP)     ;;READ THE TTY
017244  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
017252  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
017260  001013                          BNE     3$              ;;BRANCH IF NO
017262  105777  161556          2$:     TSTB    @#TKS           ;;WAIT FOR A CHARACTER
017266  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
017270  117746  161552                  MOVB    @#TKB,-(SP)     ;;GET CHARACTER
017274  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
017300  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
017304  001366                          BNE     2$              ;;IF NOT DISCARD IT
017306  000750                          BR      1$              ;;YES, RESUME
017310  026627  000004  000021  3$:     CMP     4(SP),#@#XON    ;;IS IT A RANDOM XON?                   ;RAN001
017316  001744                          BEQ     1$              ;;BRANCH IF YES                         ;RAN001
017320  026627  000004  000140          CMP     4(SP),#140      ;;IS IT UPPER CASE?
017326  002407                          BLT     4$              ;;BRANCH IF YES
017330  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
017336  003003                          BGT     4$              ;;BRANCH IF YES
017340  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
017346  000002                  4$:     RTI                     ;;GO BACK TO USER
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                ;*CALL:
                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
017350  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
017352  012703  017456          1$:     MOV     #@#TTYIN,R3     ;;GET ADDRESS
017356  022703  017466          2$:     CMP     #@#TTYIN+8.,R3  ;;BUFFER FULL?
017362  101405                          BLOS    4$              ;;BR IF YES
017364  104407                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
017366  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
017370  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
017374  001003                          BNE     3$              ;;SKIP IF NOT
017376  104401  001062          4$:     TYPE    ,$QUES          ;;TYPE A '?'
017402  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
017404  111337  017454          3$:     MOVB    (R3),@#9        ;;ECHO THE CHARACTER
017410  104401  017454                  TYPE    ,@#9
017414  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
017420  001356                          BNE     2$              ;;LOOP IF NOT RETURN
017422  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
```

TTY INPUT ROUTINE

```
017426  104401  001064                  TYPE    ,$LF            ;;TYPE A LINE FEED
017432  012603                          MOV     (SP)+,R3        ;;RESTORE R3
017434  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
017436  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
017444  012766  017456  000004          MOV     #$TTYIN,4(SP)
017452  000002                          RTI                     ;;RETURN
017454     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
017455     000                          .BYTE   0               ;;TERMINATOR
017456                          $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
017466     136     125     015  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
017473     136     107     015  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
017500     015     012     123  $MSWR:  .ASCIZ  <15><12>/SWR = /
017511     040     040     116  $MNEW:  .ASCIZ  /  NEW = /
  3746
```

```
                                .SBTTL  TRAP DECODER
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                ;*GO TO THAT ROUTINE.
017522  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
017524  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
017530  005740                          TST     -(R0)           ;;BACKUP BY 2
017532  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
017534  006300                          ASL     R0              ;;POSITION FOR INDEXING
017536  016000  017556                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
017542  000200                          RTS     R0              ;;GO TO ROUTINE
                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
017544  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
017546  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
017554  000002                          RTI                     ;;RESTORE THE PSW
                                .SBTTL  TRAP TABLE
                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                ;*BY THE "TRAP" INSTRUCTION.
                                ;       ROUTINE
                                ;       -------
017556  017544                  $TRPAD: .WORD   $TRAP2
017560  016144                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
017562  016534                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
017564  016510                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
017566  016550                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
017570  017006                          $GTSWR  ;;CALL=GTSWR    TRAP+5(104405)  GET SOFT-SWR SETTING
017572  016736                          $CKSWR  ;;CALL=CKSWR    TRAP+6(104406)  TEST FOR CHANGE IN SOFT-SWR
017574  017220                          $RDCHR  ;;CALL=RDCHR    TRAP+7(104407)  TTY TYPEIN CHARACTER ROUTINE
017576  017350                          $RDLIN  ;;CALL=RDLIN    TRAP+10(104410) TTY TYPEIN STRING ROUTINE
```

ECHO TEST

```
                                        .SBTTL  ECHO TEST
                                ;;**************************************************************
                                ;*THIS ROUTINE WILL ECHO ANY CHARACTER TYPED
                                ;*AT THE CONSOLE
                                ;*THE TEST IS HALTED BY TYPING A CONTROL-C
                                ;*TEST CAN BE RESTARTED AFTER HALTING BY JUST CONTINUING
                                ;;**************************************************************
017600                          ECHO:
017600  000005                          RESET                   ;CLEAR EVERYTHING
017602  112777  000052  163052          MOVB    #'*,@CTBUF      ;TRANSMIT PROMPT "*"
017610  105777  163040          2$:     TSTB    @CRCSR          ;WAIT FOR INPUT
017614  100375                          BPL     2$
017616  117777  163034  163036          MOVB    @CRBUF,@CTBUF   ;ECHO INPUT
017624  017700  163026                  MOV     @CRBUF,R0       ;STORE INPUT
017630  100023                          BPL     5$              ;BR IF "ERROR"NOT SET
017632  052701  010000                  BIS     #BIT12,R1       ;SET PARITY ERROR TEST MASK
017636  030100                          BIT     R1,R0           ;CHECK  FOR PARITY ERROR FLAG
017640  001403                          BEQ     3$              ;BR IF NOT SET
017642  004737  017724                  JSR     PC,MSG          ;REPORT PARITY ERROR
017646  017752                          MPAR
017650  006301                  3$:     ASL     R1              ;SHIFT MASK TO TEST "FR" FLAG
017652  030100                          BIT     R1,R0           ;TEST FOR FRAMING ERROR FLAG
017654  001403                          BEQ     4$              ;BR IF NOT SET
017656  004737  017724                  JSR     PC,MSG          ;REPSORT FRAMING ERROR
017662  017763                          MFR
017664  006301                  4$:     ASL     R1              ;SHIFT MASK TO TEST "OR" FLAG
017666  030100                          BIT     R1,R0           ;TEST FOR OVERFLOW ERROR
017670  001403                          BEQ     5$              ;BR IF NOT SET
017672  004737  017724                  JSR     PC,MSG          ;REPORT OVERFLOW ERROR
017676  017775                          MOR
017700  042700  000200          5$:     BIC     #BIT7,R0        ;CLEAR ANY PARITY BIT
017704  022700  000003                  CMP     #3,R0           ;WAS INPUT CONTROL-C
017710  001337                          BNE     2$              ;BR IF NOT
017712  004737  017724                  JSR     PC,MSG          ;REPORT PROGRAM STOP
017716  020010                          MSTOP
017720  000000                          HALT                    ;END OF TEST HALT
017722  000726                          BR      ECHO            ;AFTER END OF TEST HALT
                                                                ;   PRESS CONTINUE TO RESTART ECHO TEST
017724  013600                  MSG:    MOV     @(SP)+,R0       ;PICK UP MESSAGE POINTER
017726  062746  000002                  ADD     #2,-(SP)        ;ADJUST RETURN PC
017732  105777  162722          WAIT:   TSTB    @CTCSR          ;WAIT FOR XMIT DONE
017736  100375                          BPL     WAIT
017740  112077  162716                  MOVB    (R0)+,@CTBUF    ;SEND CHARACTER
017744  105710                          TSTB    (R0)            ;IS THIS END OF MESSAGE?
017746  001371                          BNE     WAIT            ;BR IF NOT
017750  000207                          RTS     PC              ;RETURN

017752      015     012     120  MPAR:   .ASCIZ  <CR><LF>/PARITY/
017763      015     012     106  MFR:    .ASCIZ  <CR><LF>/FRAMING/
017775      015     012     117  MOR:    .ASCIZ  <CR><LF>/OVERFLOW/
020010      015     012     123  MSTOP:  .ASCIZ  <CR><LF>/STOP/
```

```
3806
3807
3808                                    .EVEN
3809
3810                                .SBTTL  TERMINAL OUTPUT TEST
3811
3812                                ;;**********************************************************************
3813                                ;*THIS ROUTINE WILL OUTPUT ALL WRITABLE CHARACTERS FOR THE
3814                                ;*THE OCTAL CODE 040 --> 377
3815                                ;*32 CHARACTERS ARE PRINTED ON EACH LINE
3816                                ;*THE PATTERN IS REPEATED EVERY THREE LINES
3817                                ;*
3818                                ;;**********************************************************************
3819
3820 020020  012701  000040        OUTTST: MOV     #40,R1          ;LOAD FIRST WRITABLE CHARACTER
3821 020024  012700  000040        1$:     MOV     #40,R0          ;LOAD CHAR COUNT PER LINE
3822 020030  105777  162624        2$:     TSTB    @CTCSR          ;WAIT FOR DONE
3823 020034  100375                        BPL     2$
3824 020036  010177  162620                MOV     R1,@CTBUF       ;TRANSMIT A CHARACTER
3825 020042  105201                        INCB    R1              ;INCREMENT CHARACTER CODE
3826 020044  005300                        DEC     R0              ;DECREMENT CHAR COUNT
3827 020046  001370                        BNE     2$              ;BR IF LINE NOT COMPLETE
3828 020050  004737  017724                JSR     PC,MSG          ;ISSUE CR,LINE FEED
3829
3830 020054  001063                        $CRLF
3831 020056  105777  162572                TSTB    @CRCSR          ;ANY CHARACTER RECEIVED?
3832 020062  100404                        BMI     4$              ;BR IF YES
3833 020064  032701  000200        3$:     BIT     #BIT7,R1        ;FINISHED ONE PASS OF WRITABLE CHARACTERS?
3834 020070  001353                        BNE     OUTTST          ;BR IF YES
3835 020072  000754                        BR      1$              ;IF NOT WRITE NEXT LINE
3836
3837 020074  117737  162556  020132  4$:     MOVB    @CRBUF,BUFF     ;SAVE WHAT EVER IT WAS
3838 020102  122737  000021  020132        CMPB    #$XON,BUFF      ;WAS IT AN XON?
3839 020110  001765                        BEQ     3$              ;BR IF YES, CONTINUE PRINTING...
3840 020112  122737  000023  020132        CMPB    #$XOFF,BUFF     ;WAS IT AN XOFF?
3841 020120  001765                        BEQ     4$              ;BR IF YES, WAIT FOR XON...
3842 020122  005077  162530                CLR     @CRBUF          ;CLEAR RECEIVER
3843 020126  000000                        HALT                    ;STOP TEST
3844 020130  000733                        BR      OUTTST          ;RESTART TEST IF CONTINUED
3845
3846 020132  000000                BUFF:   .WORD   0
```

```
3848
3849
3850 020134  103  101  116  EM1:   .ASCIZ  /CAN NOT ACCESS TCSR/
3851 020160  103  101  116  EM2:   .ASCIZ  /CAN NOT ACCESS TBUF/
3852 020204  124  103  123  EM3:   .ASCIZ  /TCSR DONE NOT CLEARED WITH TBUF FULL/
3853 020251  124  103  123  EM4:   .ASCIZ  /TCSR DONE NOT SET/
3854 020273  124  103  123  EM5:   .ASCIZ  /TCSR DONE NOT SET WITH RESET/
3855 020330  103  101  116  EM6:   .ASCIZ  /CAN NOT ACCESS RCSR/
3856 020354  103  101  116  EM7:   .ASCIZ  /CAN NOT ACCESS RBUF/
3857 020400  103  101  116  EM10:  .ASCIZ  /CAN NOT ACCESS LKS/
3858 020423  102  111  124  EM11:  .ASCIZ  /BIT0 OF TCSR NOT CLEAR AFTER RESET/
3859 020466  103  101  116  EM12:  .ASCIZ  /CAN NOT SET BIT0 OF TCSR/
3860 020517  103  101  116  EM13:  .ASCIZ  /CAN NOT CLEAR BIT0 OF TCSR/
3861 020552  122  105  123  EM14:  .ASCIZ  /RESET DID NOT CLEAR BIT0 OF TCSR/
3862 020613  102  111  124  EM15:  .ASCIZ  /BIT2 OF TCSR NOT CLEAR AFTER RESET/
3863 020656  103  101  116  EM16:  .ASCIZ  /CAN NOT SET BIT2 OF TCSR/
3864 020707  103  101  116  EM17:  .ASCIZ  /CAN NOT CLEAR BIT2 OF TCSR/
3865 020742  122  105  123  EM20:  .ASCIZ  /RESET DID NOT CLEAR BIT2 OF TCSR/
3866 021003  102  111  124  EM21:  .ASCIZ  /BIT6 OF TCSR NOT CLEAR AFTER RESET/
3867 021046  130  115  111  EM22:  .ASCIZ  /XMIT INTERRUPT AT PRIORITY 7/
3868 021103  103  101  116  EM23:  .ASCIZ  /CAN NOT SET BIT6 OF TCSR/
3869 021134  103  101  116  EM24:  .ASCIZ  /CAN NOT CLEAR BIT6 OF TCSR/
3870 021167  122  105  123  EM25:  .ASCIZ  /RESET DID NOT CLEAR BIT6 OF TCSR/
3871 021230  102  111  124  EM26:  .ASCIZ  /BIT6 OF RCSR NOT CLEAR AFTER RESET/
3872 021273  122  103  126  EM27:  .ASCIZ  /RCVR INTERRUPT WITH PRIORITY 7/
3873 021332  103  101  116  EM30:  .ASCIZ  /CAN NOT SET BIT6 OF RCSR/
3874 021363  103  101  116  EM31:  .ASCIZ  /CAN NOT CLEAR BIT6 OF RCSR/
3875 021416  103  101  116  EM32:  .ASCIZ  /CAN NOT CLEAR BIT6 OF RCSR WITH RESET/
3876 021464  102  111  124  EM33:  .ASCIZ  /BIT6 OF LKS NOT CLEAR AFTER RESET/
3877 021526  114  113  123  EM34:  .ASCIZ  /LKS INTERRUPT WITH PRIORITY 7/
3878 021564  103  101  116  EM35:  .ASCIZ  /CAN NOT SET BIT6 OF LKS/
3879 021614  103  101  116  EM36:  .ASCIZ  /CAN NOT CLEAR BIT6 OF LKS/
3880 021646  122  105  123  EM37:  .ASCIZ  /RESET DID NOT CLEAR BIT6 OF LKS/
3881 021706  104  125  101  EM40:  .ASCIZ  /DUAL ADDRESSING ERROR/
3882 021734  102  111  124  EM41:  .ASCIZ  /BIT6 OF LKS NOT SET AFTER RESET/
3883 021774  103  101  116  EM42:  .ASCIZ  /CAN NOT CLEAR BIT7 OF LKS/
3884 022026  102  111  124  EM43:  .ASCIZ  /BIT7 OF LKS DOES NOT SET/
3885 022057  122  124  103  EM44:  .ASCIZ  /RTC INTERRUPT AT PRIORITY 7/
3886 022113  122  124  103  EM45:  .ASCIZ  /RTC INTERRUPTS WHEN DISABLED/
3887 022150       EM47:
3888 022150  122  124  103  EM46:  .ASCIZ  /RTC INTERRUPT DID NOT OCCUR/
3889 022204  122  124  103  EM50:  .ASCIZ  /RTC DOUBLE INTERRUPT/
3890 022231  122  105  123  EM51:  .ASCIZ  /RESET DID NOT INTERRUPT/
3891 022261  122  124  103  EM52:  .ASCIZ  /RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS/
3892 022336  103  114  117  EM53:  .ASCIZ  /CLOCK REPEATABILITY/
3893 022362  130  115  111  EM54:  .ASCIZ  /XMIT INTERRUPTS WHEN DISABLED/
3894 022420  130  115  111  EM56:  .ASCIZ  /XMIT INTERRUPTS AT PRIORITY 7/
3895 022456  130  115  111  EM57:  .ASCIZ  /XMIT INTERRUPTS WITH ENABLE CLEAR/
3896 022520       EM55:
3897 022520  130  115  111  EM60:  .ASCIZ  /XMIT DID NOT INTERRUPT/
3898 022547  130  115  111  EM61:  .ASCIZ  /XMIT RE-INTERRUPTED/
3899 022573  114  117  101  EM62:  .ASCIZ  /LOADING TBUF DID NOT CLEAR INTERRUPT/
3900 022640  122  103  126  EM63:  .ASCIZ  /RCVR ACTIVE NOT SET/
3901 022664  122  103  126  EM64:  .ASCIZ  /RCVR DONE NEVER SET/
3902 022710  122  103  126  EM65:  .ASCIZ  /RCVR ACTIVE NOT CLEARED WITH DONE SET/
3903 022756  122  105  123  EM66:  .ASCIZ  /RESET DID NOT CLEAR RCVR DONE/
3904 023014  122  104  122  EM67:  .ASCIZ  /RDR ENABLE DID NOT CLEAR RCVR DONE/
```

```
 3905 023057      122      105      101  EM70:   .ASCIZ  /READING RBUF DID NOT CLEAR RCVR DONE/
 3906 023124      122      103      126  EM71:   .ASCIZ  /RCVR INTERRUPTS WITH ENABLE CLEAR/
 3907 023166      122      103      126  EM73:   .ASCIZ  /RCVR INTERRUPTS AT PRIORITY 7/
 3908 023224      122      103      126  EM74:   .ASCIZ  /RCVR INT RQST PASSED WITH ENABLE CLEAR/
 3909 023273                              EM72:
 3910 023273      122      103      126  EM75:   .ASCIZ  /RCVR DID NOT INTERRUPT/
 3911 023322      122      103      126  EM76:   .ASCIZ  /RCVR RE-INTERRUPTED/
 3912 023346      122      105      101  EM77:   .ASCIZ  /READING RBUF DID NOT CLEAR INTERRUPT/
 3913 023413      122      105      123  EM100:  .ASCIZ  /RESET DID NOT CLEAR RCVR INTERRUPT/
 3914 023456      047      117      122  EM101:  .ASCIZ  /'OR' FLAG DID NOT SET/
 3915 023504      047      105      122  EM102:  .ASCIZ  /'ERROR' NOT SET WITH 'OR' FLAG/
 3916 023543      102      122      105  EM103:  .ASCIZ  /BREAK DID NOT XMIT ALL ZEROES/
 3917 023601      102      122      105  EM104:  .ASCIZ  /BREAK DID NOT SET 'FR' ERROR/
 3918 023636      104      101      124  EM105:  .ASCIZ  /DATA COMPARE ERROR/
 3919 023661      114      117      117  EM106:  .ASCIZ  /LOOP-BACK DATA COMPARE ERROR/
 3920 023716      124      111      115  EM107:  .ASCIZ  /TIMEOUT IN EXERCISER TEST/
 3921 023750      111      116      103  EM110:  .ASCIZ  /INCORRECT RECEIVE COUNT/
 3922 024000      104      101      124  EM111:  .ASCIZ  /DATA COMPARE ERROR IN EXERCISER/
 3923 024040      124      122      101  EM112:  .ASCIZ  /TRAP CATCHER/
 3924 024055      116      117      040  EM113:  .ASCIZ  /NO CLK INTERRUPT IN EXERCISER/
 3925 024113      042      105      122  EM114:  .ASCIZ  /"ERROR" NOT SET WITH "FR" FLAG/
 3926 024152      122      103      126  EM115:  .ASCIZ  /RCV ACTIVE NOT CLEAR WITH INIT/
 3927 024211      122      103      126  EM116:  .ASCIZ  /RCV ACTIVE WITHOUT "START" BIT/
 3928 024250      122      104      122  EM117:  .ASCIZ  /RDR ENABLE NOT CLEAR WITH RCV ACTIVE/
 3929
 3930 024315      124      105      123  DH1:    .ASCIZ  /TEST#   ERROR#  TCSR/
 3931 024342      124      105      123  DH2:    .ASCIZ  /TEST#   ERR PC  TBUF/
 3932 024367      124      105      123  DH6:    .ASCIZ  /TEST#   ERR PC  RCSR/
 3933 024414      124      105      123  DH7:    .ASCIZ  /TEST#   ERR PC  RBUF/
 3934 024441      124      105      123  DH10:   .ASCIZ  /TEST#   ERR PC  LKS/
 3935 024465      124      105      123  DH40:   .ASCIZ  /TEST#   ERR PC  GOOD    BAD/
 3936 024521      124      105      123  DH53:   .ASCIZ  /TEST#   ERR PC  LKS     CNT1    CNT2/
 3937 024566      124      105      123  DH103:  .ASCIZ  /TEST#   ERR PC  RCSR    DATA/
 3938 024623      124      105      123  DH105:  .ASCIZ  /TEST#   ERR PC  RCSR    GOOD    BAD/
 3939 024667      124      105      123  DH110:  .ASCIZ  /TEST#   ERR PC  RCSR    TRANS   RCV/
 3940 024733      124      105      123  DH112:  .ASCIZ  /TEST#   ERR PC  RCSR    OLDPC   TRAP ADR/
 3941                                     .EVEN
 3942 025004      015      012      103  M1:     .ASCIZ  <CR><LF>?CZDLDIO DL11-W 11/44 MFM SLU?
 3943                                     .EVEN
 3944 025044      015      012           M2:     .ASCII  <CR><LF>
 3945 025046      040      040      040  M2A:    .ASCIZ  /   DEVICES UNDER TEST  /
 3946
 3947                                     .EVEN
 3948 025076  001072   001016   002640  DT1:    .WORD   $TESTN,$ERRPC,TCSR,0
 3949 025106  001072   001016   002642  DT2:    .WORD   $TESTN,$ERRPC,TBUF,0
 3950 025116  001072   001016   002634  DT6:    .WORD   $TESTN,$ERRPC,RCSR,0
 3951 025126  001072   001016   002636  DT7:    .WORD   $TESTN,$ERRPC,RBUF,0
 3952 025136  001072   001016   002674  DT10:   .WORD   $TESTN,$ERRPC,LKS,0
```

```
 3953 025146  001072  001016  001020  DT40:    .WORD   $TESTN,$ERRPC,$GDADR,$BDADR,0
 3954 025160  001072  001016  002674  DT53:    .WORD   $TESTN,$ERRPC,LKS,FIRST,SECND,0
 3955 025174  001072  001016  002634  DT103:   .WORD   $TESTN,$ERRPC,RCSR,$BDDAT,0
 3956 025206  001072  001016  002634  DT105:   .WORD   $TESTN,$ERRPC,RCSR,$GDDAT,$BDDAT,0
 3957 025222  001072  001016  002634  DT110:   .WORD   $TESTN,$ERRPC,RCSR,XMTCNT,RCVCNT,0
 3958 025236  001072  001016  002634  DT112:   .WORD   $TESTN,$ERRPC,RCSR,OLDPC,BDVECT,0
 3959 025252  000000                  ENDADR:  .WORD   0                ;END ADDRESS FOR CHECKSUM AND SAVE AREA
 3960                                                                   ;OF WRAP AROUND CONSOLE TEST
 3961
 3962          000001                           .END
```

SYMBOL TABLE

```
ABASE = 176500
ACDW1 = 000000
ACDW2 = 000000
ACPUOP= 000000
ADDW0 = 000000
ADDW1 = 000000
ADDW10= 000000
ADDW11= 000000
ADDW12= 000000
ADDW13= 000000
ADDW14= 000000
ADDW15= 000000
ADDW2 = 000000
ADDW3 = 000000
ADDW4 = 000000
ADDW5 = 000000
ADDW6 = 000000
ADDW7 = 000000
ADDW8 = 000000
ADDW9 = 000000
ADEVCT= 000000
ADEVM = 000000
ADR     004132
ADRTBL  002702
AENV  = 000000
AENVM = 000000
AFATAL= 000000
AMADR1= 000000
AMADR2= 000000
AMADR3= 000000
AMADR4= 000000
AMAMS1= 000000
AMAMS2= 000000
AMAMS3= 000000
AMAMS4= 000000
AMSGAD= 000000
AMSGLG= 000000
AMSGTY= 000000
AMTYP1= 000000
AMTYP2= 000000
AMTYP3= 000000
AMTYP4= 000000
APASS = 000000
APRIOR= 000000
APTCSU= 000040
APTENV= 000001
APTSIZ= 000200
APTSPO= 000100
APTSVD  003636
ASWREG= 000000
ATESTN= 000000
AUNIT = 000000
AUSWR = 000400
AVECT1= 000300
AVECT2= 000000
BDVECT  002622
BEGIN   003774
BGNADD= 003014
BIT0  = 000001
BIT00 = 000001
BIT01 = 000002
BIT02 = 000004
BIT03 = 000010
BIT04 = 000020
BIT05 = 000040
BIT06 = 000100
BIT07 = 000200
BIT08 = 000400
BIT09 = 001000
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BPT   = 000003
BPTVEC= 000014
BUF     002454
BUFF    020132
CATCH   014520
CHKSUM  014010
CKSWR = 104406
CLK     013630
CLKCNT  002444
CMPARE  007330
CNTLP   002564
COMP    013432
CONT    006246
CONT41  012354
CPSAVE  015674
CR    = 000015
CRBUF   002656
CRCSR   002654
CRLF  = 000200
CRPSW   002666
CRVECT  002664
CTBUF   002662
CTCSR   002660
CTPSW   002672
CTSTFL  002624
CTVECT  002670
DDISP = 177570
DEVADR  003014
DH1     024315
DH10    024441
DH103   024566
DH105   024623
DH110   024667
DH112   024733
DH2     024342
DH40    024465
DH53    024521
DH6     024367
DH7     024414
DISPLA  001042
DISPRE  000174
DSWR  = 177570
DT1     025076
DT10    025136
DT103   025174
DT105   025206
DT110   025222
DT112   025236
DT2     025106
DT40    025146
DT53    025160
DT6     025116
DT7     025126
DVADT   003664
ECHO    017600
EMTVEC= 000030
EM1     020134
EM10    020400
EM100   023413
EM101   023456
EM102   023504
EM103   023543
EM104   023601
EM105   023636
EM106   023661
EM107   023716
EM11    020423
EM110   023750
EM111   024000
EM112   024040
EM113   024055
EM114   024113
EM115   024152
EM116   024211
EM117   024250
EM12    020466
EM13    020517
EM14    020552
EM15    020613
EM16    020656
EM17    020707
EM2     020160
EM20    020742
EM21    021003
EM22    021046
EM23    021103
EM24    021134
EM25    021167
EM26    021230
EM27    021273
EM3     020204
EM30    021332
EM31    021363
EM32    021416
EM33    021464
EM34    021526
EM35    021564
EM36    021614
EM37    021646
EM4     020251
EM40    021706
EM41    021734
EM42    021774
EM43    022026
EM44    022057
EM45    022113
EM46    022150
EM47    022150
EM5     020273
EM50    022204
EM51    022231
EM52    022261
EM53    022336
EM54    022362
EM55    022520
EM56    022420
EM57    022456
EM6     020330
EM60    022520
EM61    022547
EM62    022573
EM63    022640
EM64    022664
EM65    022710
EM66    022756
EM67    023014
EM7     020354
EM70    023057
EM71    023124
EM72    023273
EM73    023166
EM74    023224
EM75    023273
EM76    023322
EM77    023346
ENDADD= 025252
ENDADR  025252
ENDB7   004676
ENDEV   014272
ENDMG   014416
ENDSTK  002412
ERROR = 104000
ERRVEC= 000004
FIRST   002336
FLAG44  003002
GETB    014126
GETLIN  014062
GOAGIN  014432
GTSWR = 104405
HT    = 000011
ID      004620
INIT    003462
IOTVEC= 000020
JIMSTK  002432
LF    = 000012
LKS     002674
LOC1    002340
LOC2    002342
MANL    003502
MFPT  = 000007
MFR     017763
MOR     017775
MPAR    017752
MSG     017724
MSTOP   020010
M1      025004
M2      025044
M2A     025046
NOEOP   014316
OLDPC   002620
OLDSUM  002436
OUTTST  020020
PFECDF  015206
PFECDH  015146
PFECDT  015176
PFECEM  015106
PFECWS  015076
PIRQ  = 177772
PIRQVE= 000240
PROMPT  002566
PR0   = 000000
PR1   = 000040
PR2   = 000100
PR3   = 000140
PR4   = 000200
PR5   = 000240
PR6   = 000300
PR7   = 000340
PS    = 177776
PSW   = 177776
PUTLIN  014030
PWRVEC= 000024
RBUF    002636
RBUF58= 176502
RCSR    002634
RCSR58= 176500
RCV     013614
RCVCNT  002440
RCVDON  010346
RDCHR = 104407
RDLIN = 104410
```

SYMBOL TABLE

```
RESTTE  014236          TBITVE= 000014          TST46   013636          $ERFLG  001003          $PASS   001074
RESVEC= 000010          TBUF    002642          TST5    004714          $ERMAX  001015          $PASTM  000506
RPSW    002646          TBUF58= 176506          TST6    004760          $ERROR  014544          $POWER  015364
RSTRT   014472          TCLOCK  005536          TST7    005012          $ERRPC  001016          $PWRDN  015212
RTCPSW  002700          TCONS   004020          TURBUF  003006          $ERRTB  001146          $PWRMG  015346
RTCVT   002676          TCSR    002640          TURCSR  003004          $ERRTY  014726          $PWRUP  015264
RVECT   002644          TCSR58= 176504          TUTBUF  003012          $ERTTL  001012          $QUES   001062
R6      =%000006        TIMER   014162          TUTCSR  003010          $ESCAP  001060          $RDCHR  017220
R7      =%000007        TKVEC = 000060          TVECT   002650          $ETABL  001106          $RDLIN  017350
SAVEPS  002434          TMP1    002626          TYPE  = 104401          $ETEND  001146          $RDSZ = 000010
SAVETE  014202          TMP2    002630          TYPOC = 104402          $FATAL  001070          $RTNAD  014410
SAVEO   002344          TMP3    002632          TYPON = 104404          $FFLG   016142          $SAVR6  015362
SAVLOC  002346          TOLER   007342          TYPOS = 104403          $FILLC  001056          $SCOPE  015374
SCOPE = 000004          TPSW    002652          VCTADR  003670          $FILLS  001055          $SETUP= 000137
SCPSW   002452          TPVEC = 000064          VCTTBL  002742          $GDADR  001020          $STUP = 177777
SECND   002616          TRAPVE= 000034          VECT    004152          $GDDAT  001024          $SVLAD  015624
SETADR  004100          TRTVEC= 000014          WACTV   010230          $GET42  014366          $SVPC = 000500
SHIFT   003740          TSTDEV  004050          WAIT    017732          $GTSWR  017006          $SWR  = 161000
SIZE    003512          TSTDVM  003646          WDONE   010362          $HIBTS  000500          $SWREG  001110
SRPSW   002450          TST1    004172          WRAP    013640          $ICNT   001004          $SWRMK= 000000
STACK = 001100          TST10   005044          WRPSW   014502          $ILLUP  015356          $TESTN  001072
START   003046          TST11   005252          WT      010316          $INTAG  001035          $TKB    001046
STKLMT= 177774          TST12   005404          XCONT   013604          $ITEMB  001014          $TKS    001044
STPSW   002446          TST13   005544          XMIT    013542          $LF     001064          $TN   = 000047
SWR     001040          TST14   005614          XMTCNT  002442          $LFLG   016141          $TPB    001052
SWREG   000176          TST15   005764          XRET    013610          $LPADR  001006          $TPFLG  001057
SW0   = 000001          TST16   006154          $APTHD  000500          $LPERR  001010          $TPS    001050
SW00  = 000001          TST17   006262          $ATYC   015722          $MADR1  001120          $TRAP   017522
SW01  = 000002          TST2    004244          $ATY1   015676          $MADR2  001124          $TRAP2  017544
SW02  = 000004          TST20   006522          $ATY3   015704          $MADR3  001130          $TRP  = 000011
SW03  = 000010          TST21   006720          $ATY4   015714          $MADR4  001134          $TRPAD  017556
SW04  = 000020          TST22   007054          $AUTOB  001034          $MAIL   001066          $TSTM   000504
SW05  = 000040          TST23   007222          $BASE   001142          $MAMS1  001116          $TSTNM  001002
SW06  = 000100          TST24   007410          $BDADR  001022          $MAMS2  001122          $TTYIN  017456
SW07  = 000200          TST25   007520          $BDDAT  001026          $MAMS3  001126          $TYPE   016144
SW08  = 000400          TST26   007646          $CHARC  016504          $MAMS4  001132          $TYPEC  016356
SW09  = 001000          TST27   010012          $CKSWR  016736          $MBADR  000502          $TYPEX  016506
SW1   = 000002          TST3    004316          $CMTAG  001000          $MFLG   016140          $TYPOC  016534
SW10  = 002000          TST30   010156          $CM3  = 000000          $MNEW   017511          $TYPON  016550
SW11  = 004000          TST31   010462          $CNTLG  017473          $MSGAD  001102          $TYPOS  016510
SW12  = 010000          TST32   010554          $CNTLU  017466          $MSGLG  001104          $UNIT   001100
SW13  = 020000          TST33   010700          $CPUOP  001114          $MSGTY  001066          $UNITM  000510
SW14  = 040000          TST34   011102          $CRLF   001063          $MSWR   017500          $USWR   001112
SW15  = 100000          TST35   011304          $DEVCT  001076          $MTYP1  001117          $VECT1  001136
SW2   = 000004          TST36   011512          $DEVM   001144          $MTYP2  001123          $VECT2  001140
SW3   = 000010          TST37   011656          $DOAGN  014406          $MTYP3  001127          $XOFF = 000023
SW4   = 000020          TST4    004466          $ENDAD  014376          $MTYP4  001133          $XON  = 000021
SW5   = 000040          TST40   012014          $ENDCT  014356          $NULL   001054          $XTSTR  015406
SW6   = 000100          TST41   012200          $ENULL  014412          $NWTST= 000001          $$GET4= 000000
SW7   = 000200          TST42   012414          $ENV    001106          $OCNT   016732          $OFILL  016733
SW8   = 000400          TST43   012576          $ENVM   001107          $OMODE  016734          .$ERRT  001146
SW9   = 001000          TST44   012744          $EOP    014326          $OVER   015660          .$X   = 000500
TA      002610          TST45   013062          $EOPCT  014350

. ABS.  025254      000
        000000      001
ERRORS DETECTED:  0
```

SYMBOL TABLE

VIRTUAL MEMORY USED: 45360 WORDS (178 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:02:34
CZDLDI.BIN,CZDLDI.SEQ/-SP=CZDLDI.MLB/ML,CZDLDI.P11

.