DUV-11

OFFLINE TRANSMITTER TESTS
MD-11-DZDUT-B

EP-DZDUT-B-DL-B

FICHE 1 OF 1

DEC 1977
digital
MADE IN USA

```
.REM    *
```

```
I D E N T I F I C A T I O N

PRODUCT CODE: MAINDEC-11-DZDUT-B-D

PRODUCT NAME: DUV11 OFFLINE TRANSMITTER TESTS

RELEASE DATE: NOV. 1977

MAINTAINER:  DIAGNOSTICS
```

```
*
.REM    *
```



```
*
```

.REM *

GENERAL DESCRIPTION

THIS DIAGNOSTIC CAN CHAIN 16 DUV11'S. THIS MEANS THAT
16 DEVICES CAN BE SEQUENTIALLY EXERCISED. THE DIAGNOSTIC
MAKES ONE PASS BEFORE PROCEEDING TO THE NEXT DEVICE,
AND CONTINUES EXERCISING ALL DEVICES IN THIS FASHION UNTIL
HALTED.

*
.REM *

1. THE DUV11 OFFLINE TRANSMITTER TESTS VERIFY THAT THE TRANSMITTER
SECTION PROVIDES THE CORRECT ERROR FLAGS, AND THAT IT
TRANSMITS CHARACTERS THRU THE BIT WINDOW AT THE CORRECT
NUMBER OF BITS PER CHARACTER.

*
.REM *

2. REQUIREMENTS

PDP-11/03 COMPUTER (LSI)

DUV11 SYNCRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

2.2 STORAGE
THE PROGRAM LOADS INTO 4K OF MEMORY WITH BOOTSTRAP

3. LOADING PROCEDURE

THE STANDARD PROCEDURE FOR LOADING ABSOLUTE BINARY TAPES
IS TO BE USED.

	STARTING ADDRESS FOR ABSOLUTE LOADER
4K	017500
8K	037500
12K	057500
16K	077500
20K	117500
24K	137500
28K	157500

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

NOTE: ALL SWITCHES PESIDE INTERNAL TO THE CPU AT ADDRESS
176. THESE MAY BE SET VIA THE CONSOLE TTY BY DIRECTLY
MODIFYING LOC. 176.

NOTE: RUNNING UNDER APT-11, THERE IS A USER SWITCH REGISTER
CALLED "$USWR". IN ORDER TO BE FLEXIBLE ON THE AVAILIBILITY OF THE
H315 CONNECTOR, ONE BIT PASSES STATUS TO APT-11.
BIT 0 IN $USWR REFLECTS THIS STATUS, A 0 = CONNECTOR

PRESENT, A 1 = CONNECTOR NOT AVAILIBLE.
THE USER CHANGES THE CONTENTS OF THIS LOCATION
WHEN BUILDING THE E TABLE, BY ANSWERING THE
PROMPT "SWITCH 2".

4.1.1 AFTER PROGRAM LOAD (INITIAL PROGRAM START)
ALL CONSOLE SWITCHES DOWN

4.1.2 TO MODIFY DEVICE VECTOR AND CONTROL REGISTER ADDRESSES
AFTER PROGRAM RESTART OR TO RUN MULTIPLE DEVICES

SW00=1

4.1.3 TO START PROGRAM AT SELECTED TEST AFTER A PROGRAM RESTART
(ONLY IN SINGLE DEVICE TESTS)
SW01=1

4.1.4 TO LOCK ON SELECTED TEST AFTER A PROGRAM RESTART
(ONLY IN SINGLE DEVICE TESTS)

SW14=1
NOTE1: IN GENERAL SW01 WILL BE USED WHEN SW14=1 IS USED
NOTE2: WITHOUT SW01=1 "LOCK ON TEST" WILL DEFAULT TO TEST 1

4.2 STARTING ADDRESS

THE STARTING ADDRESS FOR ALL TESTS IS 000200

THE RETARTING ADDRESS FOR ALL TESTS IS 000200
THE STARTING ADDRESS TO ENTER A SELECTED TEST IS 000200
THE STARTING ADDRESS TO LOCK ON TEST IS 000200

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 INITIAL PROGRAM START

4.3.1.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

4.3.1.2 SET SWITCH REGISTER (LOC. 176) TO ZERO.

4.3.1.3 TYPE 200G.

4.3.1.4 PROGRAM WILL START.

* .REM *

4.3.1.5 THE PROGRAM WILL TYPE "DUV11 DZDUT-B TAPE D" (ONCE ONLY)

* .REM *

4.3.1.6 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT IS ABOUT
TO START TESTING ,AND THEN TESTING WILL BEGIN

4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN

4.3.2.1 THE PROGRAM WILL TYPE "R" AND WILL COMMENCE TESTING

4.3.3 PROGRAM RESTART WITH SW00=1

4.3.3.1 SET SWITCH REGISTER (LOC. 176) TO A 000001.

4.3.3.2 TYPE 200G.

4.3.3.3 PROGRAM WILL START.

4.3.3.4 THE PROGRAM WILL TYPE " 1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.5 TYPE IN THE ADDRESS OF THE FIRST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.4

4.3.3.6 THE PROGRAM WILL TYPE "VECTOR ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.7 TYPE IN THE BASE RECEIVER INTERRUPT VECTOR ADDRESS FOR THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.6

4.3.3.8 THE PROGRAM WILL TYPE "ARE YOU RUNNING MULTIPLE DEVICES ?" (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.9 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS GIVEN, THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.8

IF A "NO" ANSWER IS GIVEN: JUMP TO SECTION 4.3.3.12
IF A "YES" ANSWER IS GIVEN: THE NEXT QUESTION IS ASKED

4.3.3.10 THE PROGRAM WILL TYPE "LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.11 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.10
NOTE: ALL ADDRESSES SHALL BE CONTIGUOUS

4.3.3.11.1 IF AN "OUT OF RANGE" ADDRESS IS TYPED IE. MORE THAN 16 (10) DEVICES AWAY (UPWARDS)..... THE

PROGRAM WILL TYPE "OUT OF RANGE: RETYPE LAST DEVICE RXCSR ADDRESS-"
AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.11.2 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"
AND WILL REPEAT THE MESSAGE OF 4.3.3.11.1

IF A DEVICE ADDRESS LOWER THAN 1ST DEVICE ADDRESS IS TYPED.....
....SCHOOLS OUT.....THERE IS NO PROTECTION FOR THIS.
THE PROGRAM WILL DEFAULT TO TWO DEVICES ACTIVE (UPWARDS FROM 1ST DEVICE ADDRESS). THE SAME APPLIES TO IDENTICAL ADDRESSES TYPED FOR FIRST AND LAST DEVICE.
OBSERVE LOCATION @ ACTREG: SEE SECTION 7.2

4.3.3.12 THE PROGRAM WILL TYPE "# OF SYNC CHARS SELECTED (1 OR 2)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD. REFER TO MANUAL FOR PROPER SWITCH SETTINGS OF SWITCH E55-4.

4.3.3.13 TYPE IN THE APPROPRIATE ANSWER "1" OR "2" FOLLOWED BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"
AND WILL REPEAT THE MESSAGE OF 4.3.3.12

4.3.3.14 THE PROGRAM WILL TYPE " IS SEC XMIT SWITCH E55-2 ON? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.15 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>. (NOTE THAT ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"
AND WILL REPEAT THE MESSAGE OF 4.3.3.14

4.3.3.16 THE PROGRAM WILL TYPE "IS SEC REC SWITCH E55-3 ON? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.17 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"
AND WILL REPEAT THE MESSAGE OF 4.3.3.16

4.3.3.18 THE PROGRAM WILL TYPE "IS OPT CLR ENABLE SWITCH E55-1 ON? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.19 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.18

4.3.3.20 THE PROGRAM WILL TYPE "ARE YOU RUNNING IN MAINT. MODE EXTERNAL ? AND .....DO YOU HAVE THE EXTERNAL MODEM BYPASS JUMPER CONNECTOR ON ? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.21 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE-RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.20

4.3.3.22 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED AND WILL COMMENCE TESTING AT TEST 1

4.3.4 PROGRAM RESTART WITH SW01=1
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED ,,,IT WILL NOT WORK IF MULTIPLE DEVICES ARE SELECTED

IF MULTIPLE DEVICES WERE PREVIOUSLY SELECTED,LOAD 000200, AND SELECT SW00=1 AND ANSWER "NO" TO THE MULTIPLE DEVICE QUESTION
SEE 4.3.3

4.3.4.1 SET SW01=1 IN SWITCH REG (LOC. 176)

4.3.4.2 TYPE 200G.

4.3.4.3 PROGRAM WILL START.

4.3.4.4 THE PROGRAM WILL TYPE "TEST PC-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.4.5 TYPE IN THE ADDRESS OF THE TEST AT WHICH THE PROGRAM IS TO BE STARTED FOLLOWED BY A <CARRIAGE RETURN>

4.3.4.6 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED TESTING AT THE SELECTED TEST

NOTE: CARE MUST BE TAKEN WHEN THIS FEATURE IS USED ,SINCE THERE IS NO PROTECTION AGAINST SELECTING AN ADDRESS THAT IS IN THE MIDDLE OF A TEST

4.3.5 PROGRAM RESTART WITH SW14 =1
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
SEE NOTE IN 4.3.4 FOR MORE DETAILS

4.3.5.1 SET SW14=1 IN SWITCH REG. (LOC. 176)

4.3.5.2 TYPE 200G.

4.3.5.3 PROGRAM WILL START.

4.3.5.4 THE PROGRAM WILL TYPE "LOCK ON SELECTED TEST ? (Y OR N)-"
AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.5.5 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A
<CARRIAGE RETURN>

IF A NO ANSWER IS GIVEN: THIS LOCK ON TEST WILL BE IGNORED
AND THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED
TESTING AT TEST 1

4.3.5.6 IF A YES ANSWER WAS GIVEN: THE PROGRAM WILL ACT AS FOLLOWS...
THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED
TESTING AT TEST 1 AND WILL REMAIN IN TEST 1 UNTIL HALTED
OR IF ANY KEY IS STRUCK ON THE TELETYPE , THE PROGRAM
WILL FREEZE ON THE NEXT TEST UNTIL A KEY IS STRUCK ON
THE TELETYPE AND SO FORTH THRU THE PROGRAM. IF SW01 =1 IT
WILL PERFORM AS IN SECTION 4.3.4 ALLOWING ONE TO FREEZE
ON A SELECTED TEST RATHER THAN DEFAULTING TO TEST 1

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS (INTERNAL TO THE CPU, ACCESSED VIA LOC. 176).

SW15 =1	HALT ON ERROR
SW14 =1	LOOP ON CURRENT TEST
SW13 =1	INHIBIT ERROR TYPEOUT
SW11 =1	INHIBIT ITERATIONS
SW10 =1	ESCAPE TO NEXT TEST ON ERROR
SW09 =1	LOOP ON ERROR
SW01 =1	RESTART PROGRAM AT SELECTED TEST
SW00 =1	RESELECT VECTOR AND CONTROL REGISTER ADRESSES &PARAMETERS AFTER A PROGRAM RESTART

TO INHIBIT "END OF PASS" TYPEOUT - TURN TELETYPE OFF

6. ERRORS

6.1 ERROR HALTS (UNDER LSI ALL HALT ERRORS RETURN CONTROL TO O.D.T.)
THERE ARE FOUR DISTINCT ERROR TYPEOUTS

6.1.1 PC+2 = ERROR PC
WHERE PC +2 IS THE ADDRESS OF THE CALL TO THE ERROR HANDLER +2

REFER TO THE ABOVE "HLT" IN DIAGNOSTIC FOR ERROR DESCRIPTION

CHECK ADDRESS @ RXCSR: TO LOCATE THE DEVICE PRESENTLY UNDER
TEST WHEN RUNNING MULTIPLE DEVICES

6.1.2 PC +2 = REGISTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.3 PC +2 = RECEIVER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING RECEIVER (RXDBUF) REGISTER

WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

6.1.4 PC +2 = TRANSMITTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING TRANSMITTER (TXCSR) REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.5 ERROR DESCRIPTIONS
SEE LISTINGS FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

6.2.1 SW15 =0
IF THE PROGRAM IS RUN WITH SW15 =0 ,NO OPERATOR ACTION IS
REQUIRED TO CONTINUE TESTING

6.2.2 SW15 =1
IF THE PROGRAM IS RUN WITH SW15 =1 ,TO CONTINUE TESTING
AFTER THE PROGRAM HAS HALTED ,PRESS THE PROCESSOR
CONSOLE "CONTINUE SWITCH"

NOTE: THE PC + 2 OF THE "HLT" WILL BE DISPLAYED IN THE DATA LIGHTS

6.2.3 ILLEGAL INTERRUPTS
IF AN INTERRUPT OCCURS TO A VECTOR ADDRESS NOT SELECTED
DURING PROGRAM INITIALIZATION, THE PROGRAM WILL HALT IN
THE TRAPCATCHER. THE ADDRESS AT WHICH THE PROGRAM
HALTS IS 2 GREATER THAN THE ADDRESS TO WHICH THE INTERRUPT
OCCURED. THE PROGRAM MUST BE RESTARTED AT 000200 TO
RECOVER FROM THIS ERROR.

6.2.4 ADDITIONAL TROUBLESHOOTING AIDS ERRCNT: & PASCNT:
CHECK THESE TWO TAG LOCATIONS FOR TOTAL # OF ERRORS AND PASSES RESPECTIVELY.
LOADING 000200 AND RESTARTING WILL CLEAR THESE LOCATIONS.

6.3 END OF PASS ROUTINE
THIS TYPEOUT IS MENTIONED HERE FOR CONVENIENCE
IT IS IN THE FORM:

```
                END OF PASS TAPE Y
                16XXXX = DEVICE

                WHERE Y IS THE TAPE LOADED

                WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

                TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF

        7.      RESTRICTIONS

        7.1     MULTIPLE DEVICES
                UP TO 16(10) DEVICES MAY BE TESTED. HOWEVER, THEY
                MUST HAVE CONTIGUOUS ADDRESSES AND VECTORS

                NOTE:  IF ALL DEVICES UNDER TEST HAVE THE SAME INTERRUPT VECTOR
                YOU CAN CHANGE  "ZERO:     ADD   #10,BASEIV      ;NEXT BLOCK
                (VECTORS)" TO "ZERO:    ADD    #0,BASEIV";
                THEREBY THE VECTOR ADDRESSES WILL NOT BE
                UPDATED AFTER EACH PASS.

        7.2     DISQUALIFYING DEVICES WHEN RUNNING MULTIPLE DEVICES

                WHEN RUNNING MULTIPLE DEVICES    AN ACTIVE BIT IS SET
                FOR EACH DEVICE RUNNING UNDER TEST  IE. BIT 0 FOR
                DEVICE 0   ,BIT 15 FOR DEVICE 15
                TO DISQUALIFY DEVICES:

        7.2.1   IF DEVICE 0 IS TO BE DISQUALIFIED ,SIMPLY RESTART
                PROGRAM WITH SW00 =1 AND OMIT THE FIRST DEVICE.

        7.2.2   IF HOWEVER, DEVICES 1 THRU 15 OR ANY COMBINATION THEREOF
                ARE TO BE DISQUALIFIED....LOAD THE LOCATION OF ACTREG:
                OBSERVE THE ACTIVE BITS (ACTIVE =1, NONACTIVE = 0)
                AND DEPOSIT 0 WHERE THOSE DEVICES ARE TO BE DISQUALIFIED

                7.2.2.1 TO RESTART...TYPE 200G...
                        THE PROGRAM WILL CONTINUE WITH THE DEVICE IT WAS IN BEFORE HALTING.

                7.2.2.2 .......OR .....SET SW00=1 IN SWITCH REG (LOC. 176) AND TYPE 200G....
                        ANSWER THE QUESTION : 1ST DEVICE :  ETC.............
                         ....THE PROGRAM WILL CONTINUE WITH DEVICE 0

                7.2.2.3 IF ALL DEVICES ARE DISQUALIFIED BY MISTAKE   THE PROGRAM
                        WILL TYPEOUT   AN ERROR MESSAGE.......TYPE 200G.

        7.3     CABLE DELAYS
                NOTE:  EXTERNAL LOOP BACK TESTS ONLY (MODEM CABLE WITH H315 CONNECTOR ON)

        7.3.1   TO PROVIDE SUFFICIENT DELAY FOR CLOCK SIGNAL OVER THE CABLE,
                LOCATION "HOLD:" MUST BE MODIFIED TO ACCOMODATE FOR FASTER MACHINES.
                        PRESENTLY "HOLD:" =20  IS SUFFICIENT TIME ON AN 11/03 MACHINE.

                BASICALLY DON'T TRY TO EXCEED 10K TO 12K RATE USING THE EIA DRIVERS
```

```
        7.4     TO USE THE "XOR" TESTER ,THE BRANCH AROUND THE "XOR"
                CODE MUST BE PATCHED TO A "NOP". (SEE LISTINGS FOR DETAILS)


        8.      DEFAULT PARAMETERS:
                1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- RXCSR: 160010

                VECTOR ADDRESS-                                DURIV: 770

                ARE YOU RUNNING MULTIPLE DEVICES ?- NO         MULTD: 0

                LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-  LASTADD: 0

                # OF SYNC CHARS SELECTED -  2          SYNCNO: 377

                IS SEC XMIT SWITCH E55-2 ON?-  YES     SEXMIT: 377

                IS SEC REC SWITCH E55-3 ON?-  YES      SEREC: 377

                IS OPT CLR ENABLE SWITCH E55-1 ON?- YES OPTCLR: 377

                DO YOU HAVE THE EXTERNAL MODEM BYPASS JUMPER
                CONNECTOR ON (H315)-  YES                      JMRBY: 377

        9.      PROGRAM DESCRIPTION

                                                                *
                                                                        .REM    *
        9.1     THIS PROGRAM PERFORMS THE OFFLINE TRANSMITTER SECTION TESTING
                OF THE DEVICE
                SEE LISTING FOR DETAILS
                                                                *
                                                                        .REM    *
        10.     FLOW CHARTS: RECEIVER FLOW, TRANSMITTER FLOW, TRANSMITTER & RECEIVER FLOW

        11.     LISTINGS
                                                                *
```

```
525
526
527
528
529
530
531
532
533
534
535
536
537             000001                  $TN=1
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
```

```
                        .ENABLE ABS

                        ;DUV11 DZDUT-B TAPE D
                        ;COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                ;STARTING PROCEDURE
                                ;TYPE 200G
                                ;PROGRAM WILL TYPE "DUV11 DZDUT-B TAPE D "
                                ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                                ;AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE D"
                                ;AND THEN RESUME TESTING

                        .SBTTL  BASIC DEFINITIONS

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

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

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

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

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

```
                        SW13=   20000
                        SW12=   10000
                        SW11=   4000
                        SW10=   2000
                        SW09=   1000
                        SW08=   400
                        SW07=   200
                        SW06=   100
                        SW05=   40
                        SW04=   20
                        SW03=   10
                        SW02=   4
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        BIT15=  100000
                        BIT14=  40000
                        BIT13=  20000
                        BIT12=  10000
                        BIT11=  4000
                        BIT10=  2000
                        BIT09=  1000
                        BIT08=  400
                        BIT07=  200
                        BIT06=  100
                        BIT05=  40
                        BIT04=  20
                        BIT03=  10
                        BIT02=  4
                        BIT01=  2
                        BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

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

```
   671            000014                  TBITVEC=14              ;;"T" BIT
   672            000014                  TRTVEC= 14              ;;TRACE TRAP
   673            000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
   674            000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
   675            000024                  PWRVEC= 24              ;;POWER FAIL
   676            000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
   677            000034                  TRAPVEC=34              ;;"TRAP" TRAP
   678            000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
   679            000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
   680            000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
```

```
                                        ;STANDARD INTERRUPT VECTORS


                    000174              .=174
000174  000000      DISPREG: 0
000176  000000      SWREG: 0
                    000200              .=200
000200  000167  001746                  JMP     .START          ;GO TO START OF PROGRAM


                    001100              .=1100
001100  000000      .WORD 0
001102  177570      LIGHTS: 177570


                                        ;PROGRAM CONTROL PARAMETERS

001104  000000      RETURN: 0
001106  000000      NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
001110  000000      LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA
001112  000000      PASCNT: 0               ;ADDRESS CONTAINING PASS COUNT
001114  000000      ERRCNT: 0                       ;ERROR COUNT
001116  000000      SAVSP:  0               ;STACK POINTER STORAGE

                                        ;PROGRAM VARIABLES

001120  000020      HOLD:   20              ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
001122  000000      SHIFT:  0               ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
001124  000000      COUNT:  0               ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
001126  000000      SAVPC:  0                       ;PROGRAM COUNTER STORAGE
001130  000000      HLD0:   0
001132  000000      HLD1:   0
001134  000000      HLD2:   0
001136  000000      HLD3:   0
001140  000000      HLD4:   0
001142  000000      HLD5:   0
001144  000000      HLD6:   0

```

```
                                        ;PROGRAM CONVERSATIONAL PARAMETERS
001146  377             SYNCNO: .BYTE   377     ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
001147  377             SEXMIT: .BYTE   377     ;SEC XMIT JUMPER "IN"
001150  377             SEREC:  .BYTE   377     ;SEC REC JUMPER "IN"
001151  377             OPTCLR: .BYTE   377     ;OPTIONAL JUMPER CLR "IN"
001152  000             MULTD:  .BYTE   0       ;NO MULTIPLE DEVICE FLAG
001153  377             JMRBY:  .BYTE   377     ;EXTERNAL MODEM BYPASS JUMPER "IN"
                        .EVEN

                                        ;PROGRAM MULTIPLE DEVICE PARAMETERS
001154  000000          BASEADD:        0       ;PROG CONTROLLED 1ST DEVICE ADDR
001156  000000          KEEPADD:        0       ;SAVED 1ST DEVICE ADDR
001160  000000          LASTADD:        0       ;LAST DEVICE RXCSR ADDR
001162  000000          BASEIV:         0       ;PROG CONTROLLED IV
001164  000000          KEEPIV:         0       ;SAVED INTR VECTOR
001166  000000          ACTREG:         0       ;ACTIVE REGISTER ,,,MODIFY THIS
                                                ;LOCATION TO DISQUALIFY OR QUALIFY
                                                ;DEVICES (1= RUN,,,0= DON'T RUN)
001170  000000          ROTADD:         0       ;ROTATING POINTER FOR ACTREG..POINTS
                                        ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DEVICES

                                        ;PROGRAM CONTROL FLAGS

001172  000            INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
001173  000            STFLG:  .BYTE   0               ;TEST START FLAG
001174  000            LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
        001176          .EVEN
        001400          .=1400
```

```
                        ;INSTRUCTION DEFINITIONS

005746                  PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
005726                  POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
010046                  PUSHR0=10046    ;SAVE R0 ON STACK =MOV R0,-(SP)
012600                  POPR0=12600     ;RESTORE R0 FROM STACK =MOV (SP)+,R0
024646                  PUSH2SP=24646   ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
022626                  POP2SP=22626    ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
                        ;REGISTER DEFINITIONS
                                ;RXCSR BIT DEFINITIONS
100000                  DSC=BIT15               ;DATA SET CHANGE
040000                  RING=BIT14              ;RING
020000                  CTS=BIT13               ;CLR TO SEND
010000                  CARDET=BIT12            ;CARRIER DETECT
004000                  RECACT=BIT11            ;REC ACTIVE
002000                  SRD=BIT10               ;SEC REC DATA
001000                  DSR=BIT9                ;DATA SET RDY
000400                  STPSYN=BIT8             ;STRIP SYNC
000200                  RXDONE=BIT7             ;REC DONE
000100                  RINTEN=BIT6             ;REC INTR ENABLE
000040                  DSINTE=BIT5             ;DSC INTR ENABLE
000020                  SYNSCH=BIT4             ;SYNC SEARCH
000010                  STD=BIT3                ;SEC XMIT DATA
000004                  RTS=BIT2                ;REQ TO SEND
000002                  DTR=BIT1                ;DATA TERM RDY
000001                  VOID=BIT0
                                ;RXDBUF BIT DEFINITIONS
100000                  RXERR=BIT15             ;REC ERROR
040000                  OVRRUN=BIT14            ;OVERRUN
020000                  FRMERR=BIT13            ;FRAME ERROR
010000                  PARER=BIT12             ;PARITY ERROR
                                ;PARCSR BIT DEFINITIONS
001000                  PAREN=BIT9              ;PARITY ENABLE
000400                  EVPAR=BIT8              ;EVEN PARITY SENSE
                                ;PARCSR WRD DEFINITIONS
030000                  SYNINT=30000            ;SYNC EXTERNAL MODE
020000                  SYNEXT=20000            ;SYNC INTERNAL MODE
000000                  ISYMOD=0                ;ISOC MODE
000000                  FIVE=0                  ;WORD LENGTH 5 BITS
002000                  SIX=2000                ;WORD LENGTH 6 BITS
004000                  SEVEN=4000              ;WORD LENGTH 7 BITS
006000                  EIGHT=6000              ;WORD LENGTH 8 BITS
000000                  NOPAR=0                 ;NO PARITY
001000                  ODDPAR=1000             ;ODD PARITY
001400                  EVEPAR=1400             ;EVEN PARITY
                                ;TXCSR BIT DEFINITIONS
100000                  DNA=BIT15               ;DATA NOT AVAILABLE
040000                  MTDATA=BIT14            ;MAINT DATA
020000                  CLK=BIT13               ;CLK
002000                  BITW=BIT10              ;BIT WINDOW
000400                  MRESET=BIT8             ;MASTER RESET
000200                  TXDONE=BIT7             ;XMIT DONE
000100                  TXINTE=BIT6             ;XMIT INTR ENABLE
```

```
807          000040                  DNAINTE=BIT5            ;DNA INTR ENAB
808          000020                  SEND=BIT4               ;SEND
809          000010                  HDXEN=BIT3              ;HDX/FDX
810          000001                  BREAK=BIT0              ;BREAK
811                                          ;TXCSR WRD DEFINITIONS
812          000000                  USER=0                  ;USER MODE
813          004000                  MINT=4000               ;MAINT INT MODE
814          010000                  MEXT=10000              ;MAINT EXT MODE
815          014000                  SYSTST=14000            ;SYSTEM TEST MODE
```

```
                                .SBTTL  COMMON TAGS

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

            001400                      .=.
001400                          $CMTAG:                         ;;START OF COMMON TAGS
001400  000000                          .WORD   0
001402     000                  $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001403     000                  $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001404  000000                  $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001406  000000                  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001410  000000                  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001412  000000                  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001414     000                  $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001415     001                  $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001416  000000                  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001420  000000                  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001422  000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001424  000000                  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001426  000000                  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001430  000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
001432  000000                          .WORD   0
001434     000                  $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001435     000                  $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001436  000000                          .WORD   0
001440  177570                  SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001442  177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001444  177560                  $TKS:   177560                  ;;TTY KBD STATUS
001446  177562                  $TKB:   177562                  ;;TTY KBD BUFFER
001450  177564                  $TPS:   177564                  ;;TTY PRINTER STATUS REG.  ADDRESS
001452  177566                  $TPB:   177566                  ;;TTY PRINTER BUFFER REG.  ADDRESS
001454     000                  $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001455     002                  $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001456     012                  $FILLC: .BYTE   12              ;;INSERT FILL CHARS.  AFTER A "LINE FEED"
001457     000                  $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001460  000000                  $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                                ;;WHICH  ($REGO) WAS OBTAINED
001462  000000                  $REGO:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001464  000000                  $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001466  000000                  $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001470  000000                  $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001472  000000                  $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001474  000000                  $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001476  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
001500  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
001502  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
001504  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
001506  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
001510  000000                  $TMP5:  .WORD   0               ;;USER DEFINED
001512  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001514  000000                  $ESCAPE: 0                      ;;ESCAPE ON ERROR ADDRESS
001516  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001522     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
001523     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
```

```
                                        $LF:    .ASCIZ  <12>            ;;LINE FEED
                                        ;;*****************************************************************
                                        .SBTTL  APT MAILBOX-ETABLE

                                        ;;*****************************************************************
                                        .EVEN
001526                                  $MAIL:                          ;;APT MAILBOX
001526  000000                          $MSGTY: .WORD   AMSGTY   ;;MESSAGE TYPE CODE
001530  000000                          $FATAL: .WORD   AFATAL   ;;FATAL ERROR NUMBER
001532  000000                          $TESTN: .WORD   ATESTN   ;;TEST NUMBER
001534  000000                          $PASS:  .WORD   APASS    ;;PASS COUNT
001536  000000                          $DEVCT: .WORD   ADEVCT   ;;DEVICE COUNT
001540  000000                          $UNIT:  .WORD   AUNIT    ;;I/O UNIT NUMBER
001542  000000                          $MSGAD: .WORD   AMSGAD   ;;MESSAGE ADDRESS
001544  000000                          $MSGLG: .WORD   AMSGLG   ;;MESSAGE LENGTH
001546                                  $ETABLE:                        ;;APT ENVIRONMENT TABLE
001546     000                          $ENV:   .BYTE   AENV     ;;ENVIRONMENT BYTE
001547     000                          $ENVM:  .BYTE   AENVM    ;;ENVIRONMENT MODE BITS
001550  000000                          $SWREG: .WORD   ASWREG   ;;APT SWITCH REGISTER
001552  000000                          $USWR:  .WORD   AUSWR    ;;USER SWITCHES
001554  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
001556     000                          $MAMS1: .BYTE   AMAMS1   ;;HIGH ADDRESS,M.S.  BYTE
001557     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
001560  000000                          $MADR1: .WORD   AMADR1   ;;HIGH ADDRESS,BLK#1
                                        ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001562     000                          $MAMS2: .BYTE   AMAMS2   ;;HIGH ADDRESS,M.S.  BYTE
001563     000                          $MTYP2: .BYTE   AMTYP2   ;;MEM.TYPE,BLK#2
001564  000000                          $MADR2: .WORD   AMADR2   ;;MEM.LAST ADDRESS,BLK#2
001566     000                          $MAMS3: .BYTE   AMAMS3   ;;HIGH ADDRESS,M.S.BYTE
001567     000                          $MTYP3: .BYTE   AMTYP3   ;;MEM.TYPE,BLK#3
001570  000000                          $MADR3: .WORD   AMADR3   ;;MEM.LAST ADDRESS,BLK#3
001572     000                          $MAMS4: .BYTE   AMAMS4   ;;HIGH ADDRESS,M.S.BYTE
001573     000                          $MTYP4: .BYTE   AMTYP4   ;;MEM.TYPE,BLK#4
001574  000000                          $MADR4: .WORD   AMADR4   ;;MEM.LAST ADDRESS,BLK#4
001576  000000                          $VECT1: .WORD   AVECT1   ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001600  000000                          $VECT2: .WORD   AVECT2   ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001602  000000                          $BASE:  .WORD   ABASE    ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001604  000000                          $DEVM:  .WORD   ADEVM    ;;DEVICE MAP
001606  000000                          $CDW1:  .WORD   ACDW1    ;;CONTROLLER DESCRIPTION WORD#1
001610  000000                          $CDW2:  .WORD   ACDW2    ;;CONTROLLER DESCRIPTION WORD#2
001612  000000                          $DDW0:  .WORD   ADDW0    ;;DEVICE DESCRIPTOR WORD#0
001614  000000                          $DDW1:  .WORD   ADDW1    ;;DEVICE DESCRIPTOR WORD#1
001616  000000                          $DDW2:  .WORD   ADDW2    ;;DEVICE DESCRIPTOR WORD#2
001620  000000                          $DDW3:  .WORD   ADDW3    ;;DEVICE DESCRIPTOR WORD#3
001622  000000                          $DDW4:  .WORD   ADDW4    ;;DEVICE DESCRIPTOR WORD#4
001624  000000                          $DDW5:  .WORD   ADDW5    ;;DEVICE DESCRIPTOR WORD#5
```

```
   928  001626  000000                  $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
   929  001630  000000                  $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
   930  001632  000000                  $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
   931  001634  000000                  $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
   932  001636  000000                  $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
   933  001640  000000                  $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
   934  001642  000000                  $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
   935  001644  000000                  $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
   936  001646  000000                  $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
   937  001650  000000                  $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
   938
   939
   940  001652                          $ETEND:
   941
   942
   943
```

```
                        ;INSTRUCTION DEFINITIONS

005746                  PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
005726                  POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
010046                  PUSHR0=10046    ;SAVE R0 ON STACK =MOV R0,-(SP)
012600                  POPR0=12600     ;RESTORE R0 FROM STACK =MOV (SP)+,R0
024646                  PUSH2SP=24646   ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
022626                  POP2SP=22626    ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
                        ;REGISTER DEFINITIONS
                                ;RXCSR BIT DEFINITIONS
100000                  DSC=BIT15               ;DATA SET CHANGE
040000                  RING=BIT14              ;RING
020000                  CTS=BIT13               ;CLR TO SEND
010000                  CARDET=BIT12            ;CARRIER DETECT
004000                  RECACT=BIT11            ;REC ACTIVE
002000                  SRD=BIT10               ;SEC REC DATA
001000                  DSR=BIT9                ;DATA SET RDY
000400                  STPSYN=BIT8             ;STRIP SYNC
000200                  RXDONE=BIT7             ;REC DONE
000100                  RINTEN=BIT6             ;REC INTR ENABLE
000040                  DSINTE=BIT5             ;DSC INTR ENABLE
000020                  SYNSCH=BIT4             ;SYNC SEARCH
000010                  STD=BIT3                ;SEC XMIT DATA
000004                  RTS=BIT2                ;REQ TO SEND
000002                  DTR=BIT1                ;DATA TERM RDY
000001                  VOID=BIT0
                                ;RXDBUF BIT DEFINITIONS
100000                  RXERR=BIT15             ;REC ERROR
040000                  OVRRUN=BIT14            ;OVERRUN
020000                  FRMERR=BIT13            ;FRAME ERROR
010000                  PARER=BIT12             ;PARITY ERROR
                                ;PARCSR BIT DEFINITIONS
001000                  PAREN=BIT9              ;PARITY ENABLE
000400                  EVPAR=BIT8              ;EVEN PARITY SENSE
                                ;PARCSR WRD DEFINITIONS
030000                  SYNINT=30000            ;SYNC EXTERNAL MODE
020000                  SYNEXT=20000            ;SYNC INTERNAL MODE
000000                  ISYMOD=0                ;ISOC MODE
000000                  FIVE=0                  ;WORD LENGTH 5 BITS
002000                  SIX=2000                ;WORD LENGTH 6 BITS
004000                  SEVEN=4000              ;WORD LENGTH 7 BITS
006000                  EIGHT=6000              ;WORD LENGTH 8 BITS
000000                  NOPAR=0                 ;NO PARITY
001000                  ODDPAR=1000             ;ODD PARITY
001400                  EVEPAR=1400             ;EVEN PARITY
                                ;TXCSR BIT DEFINITIONS
100000                  DNA=BIT15               ;DATA NOT AVAILABLE
040000                  MTDATA=BIT14            ;MAINT DATA
020000                  CLK=BIT13               ;CLK
002000                  BITW=BIT10              ;BIT WINDOW
000400                  MRESET=BIT8             ;MASTER RESET
000200                  TXDONE=BIT7             ;XMIT DONE
000100                  TXINTE=BIT6             ;XMIT INTR ENABLE
```

```
1000          000040                  DNAINTE=BIT5            ;DNA INTR ENAB
1001          000020                  SEND=BIT4               ;SEND
1002          000010                  HDXEN=BIT3              ;HDX/FDX
1003          000001                  BREAK=BIT0              ;BREAK
1004                                          ;TXCSR WRD DEFINITIONS
1005          000000                  USER=0                  ;USER MODE
1006          004000                  MINT=4000               ;MAINT INT MODE
1007          010000                  MEXT=10000              ;MAINT EXT MODE
1008          014000                  SYSTST=14000            ;SYSTEM TEST MODE
```

```
1009                                    .SBTTL  ERROR POINTER TABLE
1010
1011                                    ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
1012                                    ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
1013                                    ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
1014                                    ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
1015                                    ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
1016
1017                                    ;*      EM              ;;POINTS TO THE ERROR MESSAGE
1018                                    ;*      DH              ;;POINTS TO THE DATA HEADER
1019                                    ;*      DT              ;;POINTS TO THE DATA
1020                                    ;*      DF              ;;POINTS TO THE DATA FORMAT
1021
1022
1023  001652                            $ERRTB:
1024                                            ;ERROR TABLE
1025  001652  001762                            EM1     ;ERROR 1        REGISTER ERROR
1026  001654  002067                            DH1
1027  001656  002116                            DT1
1028  001660  002132                            DF1
1029  001662  002022                            EM2     ;ERROR 2        RECEIVER ERROR
1030  001664  002067                            DH1
1031  001666  002116                            DT1
1032  001670  002132                            DF1
1033  001672  002043                            EM3     ;ERROR 3        TRANSMITTER ERROR
1034  001674  002067                            DH1
1035  001676  002116                            DT1
1036  001700  002132                            DF1
1037  001702  001746                            EM4     ;ERROR 4        BIT ERROR (GENERAL)
1038  001704  000000                            0
1039  001706  002126                            DT4
1040  001710  002132                            DF1
1041
1042                                            ;DEFAULT DU ADDRESSES
1043  001712  160010                    RXCSR:  160010
1044  001714  160011                    HRXCSR: 160011
1045  001716  160012                    RXDBUF: 160012
1046  001720  160013                    HRXDBUF:        160013
1047  001722  160012                    PARCSR: 160012
1048  001724  160013                    HPARCSR:        160013
1049  001726  160014                    TXCSR:  160014
1050  001730  160015                    HTXCSR: 160015
1051  001732  160016                    TXDBUF: 160016
1052  001734  160017                    HTXDBUF:        160017
1053                                            ;DEFAULT DU VECTORS
1054  001736  000770                    DURIV:  770     ;REC INTR VECTOR
1055  001740  000772                    DURIS:  772     ;REC INTR STATUS
1056  001742  000774                    DUTIV:  774     ;XMIT INTR VECTOR
1057  001744  000776                    DUTIS:  776     ;XMIT INTR STATUS
1058                                            ;ERROR MESSAGES
1059  001746  020040  051105  047522    EM4:    .ASCIZ  /  ERROR PC /
1060  001754  020122  041520  000040
1061  001762  020040  047503  050115    EM1:    .ASCIZ /  COMPARISON ERROR ON REGISTERS/
1062  001770  051101  051511  047117
1063  001776  042440  051122  051117
1064  002004  047440  020116  042522
```

```
1065  002012  044507  052123  051105
1066  002020  000123
1067  002022  020040  042522  042503  EM2:    .ASCIZ  /  RECEIVER ERROR/
1068  002030  053111  051105  042440
1069  002036  051122  051117     000
1070  002043     040  052040  040522  EM3:    .ASCIZ  /  TRANSMITTER ERROR/
1071  002050  051516  044515  052124
1072  002056  051105  042440  051122
1073  002064  051117     000
1074                                          ;DATA HEADERS FOR ERROR MESSAGES
1075  002067     105  051122  041520  DH1:    .ASCIZ  /ERRPC  WANTED  ACTUAL/
1076  002074  020040  040527  052116
1077  002102  042105  020040  041501
1078  002110  052524  046101     000
1079          002116                  .EVEN
1080                                          ;DATA TABLES FOR ERROR MESSAGES
1081  002116  001416  001130  001132  DT1:    .WORD   $ERRPC,HLD0,HLD1,0
1082  002124  000000
1083
1084  002126  001416  000000          DT4:    .WORD   $ERRPC,0
1085
1086  002132     000     000     000  DF1:    .BYTE   0,0,0,0
1087  002135     000
1088                                  .EVEN
1089                                  .SBTTL  ACT11 HOOKS
1090
1091                                  ;;**************************************************************************
1092                                  ;HOOKS REQUIRED BY ACT11
1093          002136                          $SVPC=.                 ;SAVE PC
1094          000046                          .=46
1095  000046  012670                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1096          000052                          .=52
1097  000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
1098          002136                          .=$SVPC                 ;; RESTORE PC
1099                                  .SBTTL  APT PARAMETER BLOCK
1100
1101                                  ;;**************************************************************************
1102                                  ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1103                                  ;;**************************************************************************
1104          002136                          .$X=.   ;;SAVE CURRENT LOCATION
1105          000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1106  000024  000200                          200     ;;FOR APT START UP
1107          000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1108  000044  002136                          $APTHDR ;;POINT TO APT HEADER BLOCK
1109          002136                          .=.$X   ;;RESET LOCATION COUNTER
1110                                  ;;**************************************************************************
1111                                  ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1112                                  ;INTERFACE SPEC.
1113
1114  002136                          $APTHD:
1115  002136  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1116  002140  001526                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1117  002142  000010                  $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
1118  002144  000010                  $PASTM: .WORD   10      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1119  002146  000000                  $UNITM: .WORD           ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1120  002150  000052                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
1121
1122
1123                                    ;PROGRAM INITIALIZATION
1124                                    ;LOCK OUT INTERRUPTS
1125                                    ;SET UP PROCESSOR STACK
1126                                    ;SET UP POWER FAIL VECTOR
1127                                    ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1128                                    ;TYPE TITLE MESSAGE
1129
1130  002152                        .START:
1131                                .SBTTL  INITIALIZE THE COMMON TAGS
1132                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1133  002152  012706  001400               MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1134  002156  005026                       CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1135  002160  022706  001440               CMP     #SWR,R6 ;;DONE?
1136  002164  001374                       BNE     .-6             ;;LOOP BACK IF NO
1137  002166  012706  001100               MOV     ##STACK,SP              ;;SETUP THE STACK POINTER
1138                                ;;INITIALIZE A FEW VECTORS
1139  002172  012737  016314  000020       MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1140  002200  012737  000340  000022       MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1141  002206  012737  014204  000030       MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1142  002214  012737  000340  000032       MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1143  002222  012737  016650  000034       MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1144  002230  012737  000340  000036       MOV     #340,@#TRAPVEC+2;LEVEL 7
1145  002236  012737  015006  000024       MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1146  002244  012737  000340  000026       MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1147  002252  005067  177234               CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1148  002256  005067  177232               CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1149  002262  112767  000001  177125       MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1150  002270  012767  002270  177110       MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1151  002276  012767  002276  177104       MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1152                                ;;SIZE FOR A HARDWARE SWITCH REGISTER.  IF NOT FOUND OR IT IS
1153                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1154  002304  013746  000004               MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1155  002310  012737  002344  000004       MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1156  002316  012767  177570  177114       MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1157  002324  012767  177570  177110       MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1158  002332  022777  177777  177100       CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1159  002340  001012                       BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1160                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
1161  002342  000403                       BR      65$             ;;BRANCH IF NO TIMEOUT
1162  002344  012716  002352       64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1163  002350  000002                       RTI
1164  002352  012767  000176  177060 65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1165  002360  012767  000174  177054       MOV     #DISPREG,DISPLAY
1166  002366  012637  000004       66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1167
1168  002372  005067  177136               CLR     $PASS           ;;CLEAR PASS COUNT
1169  002376  132767  000200  177143       BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1170  002404  001403                       BEQ     67$             ;;YES,USE NON-APT SWITCH
1171  002406  012767  001550  177024       MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1172  002414                        67$:
1173  002414  012706  001100               MOV     #STACK,SP               ;SET STACK
1174  002420  106427  000340               MTPS    #340                    ;LOCK INTERRUPTS
1175  002424  012737  015006  000024       MOV     #.PFAIL,@#24            ;SET UP POWER FAIL VECTOR
1176  002432  105067  176535               CLRB    STFLG                   ;CLEAR START FLAG
```

```
1177  002436  005067  176450                CLR     PASCNT                  ;CLEAR PASS COUNT
1178  002442  105067  176735                CLRB    $ERFLG                  ;CLEAR ERROR FLAG
1179  002446  005067  176740                CLR     $ERTTL                  ;CLEAR ERROR COUNT
1180  002452  005067  176740                CLR     $ERRPC                  ;CLEAR LAST EPROR POINTER
1181  002456  012767  000001  176716        MOV     #1,$TSTNM               ;SET UP FOR TEST 1
1182  002464  012767  002152  176412        MOV     #.START,RETURN          ;SET UP FOR POWER FAIL BEFORE
1183                                                                        ;TESTING STARTS
1184  002472  013746  000006                MOV     @#6,-(SP)
1185  002476  013746  000004                MOV     @#4,-(SP)
1186  002502  012737  002516  000004        MOV     #1$,@#4
1187  002510  005777  176724                TST     @SWR
1188  002514  000407                        BR      2$
1189  002516  012767  000176  176714  1$:   MOV     #SWREG,SWR
1190  002524  012767  000174  176710        MOV     #DISPREG,DISPLAY
1191  002532  022626                        CMP     (SP)+,(SP)+
1192  002534  012637  000004          2$:   MOV     (SP)+,@#4
1193  002540  012637  000006                MOV     (SP)+,@#6
1194  002544  022767  000176  176666        CMP     #SWREG,SWR
1195  002552  001007                        BNE     3$
1196  002554  005737  000042                TST     @#42                    ;CHECK FOR CHAIN
1197  002560  001402                        BEQ     33$
1198  002562  000167  000522                JMP     .BEGIN
1199  002566  004767  010200          33$:  JSR     PC,CNTLU
1200  002572  105767  176374          3$:   TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
1201  002576  001004                        BNE     ONCE
1202  002600  104401  015146                TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
1203  002604  105167  176362                COMB    INIFLG                  ;IF NOT SET FLAG AND DO
1204  002610  105767  176732          ONCE: TSTB    $ENV            ;APT CONTROL?
1205  002614  001410                        BEQ     11$             ;BR IF NO
1206  002616  032767  000001  176726        BIT     #1,$USWR        ;EXTENAL JUMPER ON?
1207  002624  001002                        BNE     12$             ;NO
1208  002626  105067  176321                CLRB    JMRBY           ;CLEAR FLAG
1209  002632  000167  000452          12$:  JMP     .BEGIN          ;GO DO IT
1210  002636  032777  000001  176574  11$:  BIT     #SW00,@SWR      ;RESELECT VECTOR & CONTROL REG?
1211  002644  001002                        BNE     1$
1212  002646  000167  000436                JMP     .BEGIN
1213  002652  012700  000300          1$:   MOV     #300,R0 ;RESTORE VECTOR AREA TO TRAPCATCHER
1214  002656  012701  000302                MOV     #302,R1 ;START AT LOCATION 300
1215  002662  012702  000004                MOV     #4,R2
1216  002666  010110          2$:   MOV     R1,(R0)
1217  002670  005011                        CLR     (R1)
1218  002672  060200                        ADD     R2,R0
1219  002674  060201                        ADD     R2,R1
1220  002676  022701  001000                CMP     #1000,R1        ;END AT LOCATION 776
1221  002702  002771                        BLT     2$
1222  002704  104406                        INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1223  002706  015214                        MREGAD  ;MESSAGE
1224  002710  104410                        PARAM   ;CONVERT STRING
1225  002712  160000                        160000  ;LOW LIMIT
1226  002714  167776                        167776  ;HIGH LIMIT
1227  002716  017144                        DUBASE  ;STORE AT THIS LOCATION
1228  002720     001          .BYTE  1        ;MASK
1229  002721     001          .BYTE  1        ;HOW MANY TIMES + 2
1230  002722  016767  014216  176226        MOV     DUBASE,KEEPADD  ;SAVE
1231  002730  004767  014056                JSR     PC,DUADDR
1232  002734  016767  176216  176212        MOV     KEEPADD,BASEADD ;RESTORE FOR ROTATION
```

```
1233  002742  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1234  002744  015201                          MVECTO  ;MESSAGE
1235  002746  104410                          PARAM   ;CONVERT STRING
1236  002750  000300                          300     ;LOW LIMIT
1237  002752  000776                          776     ;HIGH LIMIT
1238  002754  001736                          DURIV   ;STORE AT THIS LOCATION
1239  002756     001                  .BYTE   1       ;MASK
1240  002757     004                  .BYTE   4       ;HOW MANY TIMES + 2
1241  002760  016767  176752  176176          MOV     DURIV,KEEPIV    ;SAVE
1242  002766  016767  176744  176166          MOV     DURIV,BASEIV    ;SET UP FOR ROTATION
1243  002774  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1244  002776  015244                          MMULT   ;MESSAGE
1245  003000  104414                          SETFLG  ;SET FLAG BASED UPON INPUT STRING
1246  003002  001152                          MULTD   ;THIS FLAG
1247  003004  105767  176142                  TSTB    MULTD   ;ARE THERE MULTIPLE DEVICES
1248                                                          ;ON THE SYSTEM ?
1249  003010  100406                          BMI     BBB     ;YES, ASK NEXT QUESTION
1250  003012  005067  176150                  CLR     ACTREG
1251  003016  005067  176146                  CLR     ROTADD
1252  003022  000167  000140                  JMP     OUTMUL  ;JUMP AROUND NEXT QUESTION
1253  003026                          BBB:
1254  003026  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1255  003030  015273                          MLASTD  ;MESSAGE
1256  003032  104410                          PARAM   ;CONVERT STRING
1257  003034  160000                          160000  ;LOW LIMIT
1258  003036  167776                          167776  ;HIGH LIMIT
1259  003040  001160                          LASTADD ;STORE AT THIS LOCATION
1260  003042     001                  .BYTE   1       ;MASK
1261  003043     001                  .BYTE   1       ;HOW MANY TIMES + 2
1262                                  ;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1263  003044  012767  000001  176116  1$:     MOV     #1,ROTADD       ;SET UP POINTER
1264  003052  005067  176110                  CLR     ACTREG  ;CLR ACTIVE REGISTER
1265  003056  056767  176106  176102  2$:     BIS     ROTADD,ACTREG   ;MAKE THIS DEVICE ACTIVE
1266  003064  000241                          CLC
1267  003066  006167  176076                  ROL     ROTADD  ;SET UP POINTER
1268  003072  103421                          BCS     3$      ;ARE YOU OUT OF RANGE ?
1269  003074  062767  000010  176052          ADD     #10,BASEADD     ;SET UP BASE ADDRESS
1270  003102  026767  176052  176044          CMP     LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
1271  003110  101362                          BHI     2$      ;NO DO IT AGAIN
1272  003112  056767  176052  176046          BIS     ROTADD,ACTREG   ;THIS ASSUMES THAT THERE ARE AT
1273                                                          ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1274                                                          ;MULTIPLE DEVICE QUESTION
1275  003120  012767  000001  176042  4$:     MOV     #1,ROTADD       ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1276  003126  016767  176024  176020          MOV     KEEPADD,BASEADD ;DITTO
1277  003134  000414                          BR      OUTMUL  ;CONTINUE QUESTIONS
1278  003136  016767  176014  176010  3$:     MOV     KEEPADD,BASEADD ;RESTORE
1279  003144  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1280  003146  015367                          MRANGE  ;MESSAGE
1281  003150  104410                          PARAM   ;CONVERT STRING
1282  003152  160000                          160000  ;LOW LIMIT
1283  003154  167776                          167776  ;HIGH LIMIT
1284  003156  001160                          LASTADD ;STORE AT THIS LOCATION
1285  003160     001                  .BYTE   1       ;MASK
1286  003161     001                  .BYTE   1       ;HOW MANY TIMES + 2
1287  003162  000167  177656                  JMP     1$      ;DO IT AGAIN
1288  003166  012767  000340  013612  OUTMUL: MOV     #340,DUPRT
```

```
1289  003174  004767  013536                    JSR     PC,DULEV
1290                                            ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1291                                            ;BUFFER TO THE CHARACTERS "1" AND "2".
1292                                            ;IF THE CHARACTER IS "1" CLEAR THE FLAG
1293                                            ;IF THE CHARACTER IS "2" SET THE FLAG
1294  003200                            AAA:
1295  003200  104406                           INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1296  003202  015605                           MSYNC   ;MESSAGE
1297  003204  122767  000061  012734  3$:      CMPB    #'1,INBUF       ;IS IT "1" ?
1298  003212  001003                           BNE     1$
1299  003214  105067  175726                  CLRB    SYNCNO  ;000
1300  003220  000412                           BR      4$
1301  003222  122767  000062  012716  1$:      CMPB    #'2,INBUF       ;IS IT "2" ?
1302  003230  001004                           BNE     2$
1303  003232  112767  177777  175706          MOVB    #-1,SYNCNO      ;377
1304  003240  000402                           BR      4$
1305  003242  104407                    2$:    INSTER  ;RETRY
1306  003244  000757                           BR      3$
1307  003246  000240                    4$:    NOP
1308  003250  104406                           INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1309  003252  015653                           MWIRE6  ;MESSAGE
1310  003254  104414                           SETFLG  ;SET FLAG BASED UPON INPUT STRING
1311  003256  001147                           SEXMIT  ;THIS FLAG
1312  003260  104406                           INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1313  003262  015724                           MWIRE5  ;MESSAGE
1314  003264  104414                           SETFLG  ;SET FLAG BASED UPON INPUT STRING
1315  003266  001150                           SEREC   ;THIS FLAG
1316  003270  104406                           INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1317  003272  015774                           MWIRE4  ;MESSAGE
1318  003274  104414                           SETFLG  ;SET FLAG BASED UPON INPUT STRING
1319  003276  001151                           OPTCLR  ;THIS FLAG
1320  003300  104406                           INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1321  003302  016053                           MEXTJ   ;MESSAGE
1322  003304  104414                           SETFLG  ;SET FLAG BASED UPON INPUT STRING
1323  003306  001153                           JMRBY   ;THIS FLAG
1324
1325                                            ;TEST START AND RESTART
1326
1327  003310  012706  001100          .BEGIN:  MOV     #STACK,SP               ;SET UP STACK
1328  003314  106427  000340                   MTPS    #340                    ;LOCK OUT INTERRUPTS
1329  003320  032777  000002  176112           BIT     #SW01,@SWR              ;IF SW01=1, GET STARTING PC
1330  003326  001413                           BEQ     3$
1331  003330  104406                           INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1332  003332  015537                           MTSTPC  ;MESSAGE
1333  003334  104410                           PARAM   ;CONVERT STRING
1334  003336  003374                           TST1    ;LOW LIMIT
1335  003340  017500                           17500   ;HIGH LIMIT
1336  003342  001402                           $TSTNM  ;STORE AT THIS LOCATION
1337  003344     001                  .BYTE    1       ;MASK
1338  003345     001                  .BYTE    1       ;HOW MANY TIMES + 2
1339  003346  016767  176030  175530           MOV     $TSTNM,RETURN
1340  003354  000403                           BR      4$
1341  003356  012767  003374  175520  3$:      MOV     #TST1,RETURN            ;START AT TEST 1
1342  003364  104401  015533          4$:     TYPE    ,MR                     ;TYPE R
1343  003370  000177  175510                   JMP     @RETURN                 ;START TESTING
1344
```

```
                                        ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
                                        ;;OF THE RECEIVER LOGIC
                                        ;;MODE: SYNINT
                                        ;;LENGTH: EIGHT
                                        ;;NOTE:  RXDONE SHOULD NEVER ASSERT
                                        ;;CHAR:  26 (SYNC)
                                        ;;
                                    ;;**********************************************************************
003374  000004                    TST1:   SCOPE
003376  052777  000400  176322            BIS     #MRESET,@TXCSR  ;MASTER RESET
003404  012777  030000  176310            MOV     #SYNINT,@PARCSR ;SET THE MODE
003412  052777  000400  176306            BIS     #MRESET,@TXCSR  ;MASTER RESET

                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
003420  012777  064001  176300            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
003426  012777  036026  176266            MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
003434  052777  000020  176250            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                        ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
003442  042777  020000  176256            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003450  052777  020000  176250            BIS     #CLK,@TXCSR     ;POKE CLK UP
                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
003456  042777  020000  176242            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003464  052777  020000  176234            BIS     #CLK,@TXCSR     ;POKE CLK UP
003472  052777  000400  176212            BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
003500  012767  000003  175416            MOV     #3,COUNT        ;# OF SYNC CHARS
003506  012767  000026  175764  1$:       MOV     #26,$TMP1       ;CHAR TO BE SHIFTED
003514  012767  000010  175400            MOV     #8.,SHIFT       ;# OF SHIFTS
003522  004767  013420                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
003526  105777  176160                    TSTB    @RXCSR  ;RXDONE ?
003532  100001                            BPL     .+4
003534  104004                            ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
003536  005367  175362                    DEC     COUNT   ;# OF SYNC CHARS
003542  001361                            BNE     1$

                                    ;;THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER RESET"
                                    ;;WHILE IN STRIP SYNC MODE
                                    ;;THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
                                    ;;STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
                                    ;;.....BUT IF AN ERROR SHOULD OCCUR.... THIS AUTOMATIC RESET
                                    ;;IS DISCOMBOBULATED
                                    ;;IE.  FORCE PARITY ERROR WHILE STRIP SYNC IS SET
                                    ;;NOTE:  NORMALLY THE LOGIC RESETS THE RXDONE &ERROR FLAGS
                                    ;;PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT....
                                    ;;BUT, IF AN RXERR OCCURS   RXDONE PLUS RXERR ARE ASSERTED
                                    ;;MODE:  ISOC (ISYMOD)
                                    ;;LENGTH:  EIGHT
                                    ;;PARITY:  EVEPAR
                                    ;;CHARACTER EXPECTED: 26
                                    ;;CHARACTER SENT:  SYNC CHARACTER
                                    ;;NOTE:  THIS TEST USES ONLY THE RECEIVER LOGIC
                                    ;;
                                    ;;**********************************************************************
003544  000004                    TST2:   SCOPE
003546  052777  000400  176152            BIS     #MRESET,@TXCSR  ;MASTER RESET
```

```
1401  003554  012777  000000  176140          MOV     #ISYMOD,@PARCSR ;SET THE MODE
1402  003562  052777  000400  176136          BIS     #MRESET,@TXCSR  ;MASTER RESET
1403
1404                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1405  003570  012777  064001  176130          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1406
1407                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1408  003576  012777  007426  176116          MOV     #ISYMOD!EIGHT!EVEPAR!26,@PARCSR
1409  003604  016703  176106                 MOV     RXDBUF,R3       ;SET UP FOR ERROR MSG
1410  003610  012767  000003  175306          MOV     #3,COUNT        ;# OF TIMES SYNC CHAR WILL BE SENT
1411  003616  052777  000020  176066          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1412                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1413  003624  042777  020000  176074          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1414  003632  052777  020000  176066          BIS     #CLK,@TXCSR     ;POKE CLK UP
1415                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1416  003640  042777  020000  176060          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1417  003646  052777  020000  176052          BIS     #CLK,@TXCSR     ;POKE CLK UP
1418  003654  052777  000400  176030          BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
1419  003662  012767  000013  175232  2$:     MOV     #11.,SHIFT      ;# OF SHIFTS
1420  003670  012767  003054  175602          MOV     #3054,$TMP1     ;SYNC CHAR + START&STOP+ PARITY
1421  003676  004767  013244          1$:     JSR     PC,RPOKE        ;SHIFT IN THIS CHARACTER
1422  003702  105777  176004                  TSTB    @RXCSR  ;RXDONE = 0 ?
1423  003706  100001                          BPL     .+4
1424  003710  104004                          ERROR   4       ;RXDONE SHOULD NOT BE SET
1425  003712  005367  175206                  DEC     COUNT   ;# OF  SYNC CHARS
1426  003716  001361                          BNE     2$      ;GO AGAIN ?
1427  003720  012700  000026                  MOV     #26,R0  ;EXPECTED
1428  003724  017701  175766                  MOV     @RXDBUF,R1      ;ACTUAL
1429                                          ;NOTE THAT THIS IS THE FIRST TIME
1430                                          ;RXDBUF IS READ.....THERE SHOULD BE
1431                                          ;NO OVER RUN ERROR      4S
1432  003730  020001                  CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
1433  003732  001401                  BEQ     .+4
1434  003734  104002                  ERROR   2       ;DATA CHARS SHOULD COMPARE
1435                                          ;THERE SHOULD BE NO RXERR'S
1436  003736  012767  000004  175160          MOV     #4,COUNT        ;# OF TIMES
1437  003744  012700  110026                  MOV     #RXERR!PARER!26,R0      ;EXPECTED
1438  003750  012767  002054  175522          MOV     #2054,$TMP1     ;BAD SYNC CHAR (WRONG PARITY)
1439  003756  012767  000013  175136  3$:     MOV     #11.,SHIFT      ;# OF SHIFTS
1440  003764  004767  013156                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1441  003770  105777  175716                  TSTB    @RXCSR  ;RXDONE = 1?
1442  003774  100401                          BMI     .+4
1443  003776  104004                          ERROR   4       ;RXDONE SHOULD BE SET
1444  004000  017701  175712                  MOV     @RXDBUF,R1      ;ACTUAL DATA
1445  004004  020001                  CMP     R0,R1   ;COMPARE EXP VS ACT
1446  004006  001401                  BEQ     .+4
1447  004010  104000                  ERROR   ;DID THE RESPECTIVE ERROR       4 STOP THE
1448                                          ;AUTOMATIC RESSETING OF RXDONE & ERROR FLAGS
1449                                          ;.....CHECK THIS.....
1450  004012  005367  175106                  DEC     COUNT   ;# OF SYNC CHARS
1451  004016  001445                          BEQ     5$      ;FINISHED ? GET OUT OF TEST
1452  004020  022767  000003  175076          CMP     #3,COUNT        ;# OF SYNC CHARS
1453  004026  001423                          BEQ     6$      ;CHECK FRAME ERROR ?
1454  004030  022767  000002  175066          CMP     #2,COUNT        ;# OF SYNC CHARS
1455  004036  001426                          BEQ     7$      ;CHECK FRAME ERROR & BAD PARITY ?
1456                                  ;NOPE THEN IT (COUNT) MUST BE = 1  THEREFORE...
```

```
1457  004040  012767  000013  175054          MOV     #11.,SHIFT      ;# OF SHIFTS
1458  004046  012767  000054  175424          MOV     #54,$TMP1       ;FRAME & PARITY ERROR
1459  004054  004767  013066                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1460                                  ;NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
1461  004060  012767  000054  175412          MOV     #54,$TMP1       ;FRAME & PARITY ERROR
1462  004066  012700  170026                  MOV     #RXERR!OVRRUN!FRMERR!PARER!26,R0        ;EXPECTED
1463  004072  000167  177660                  JMP     3$      ;DO IT AGAIN
1464  004076  012767  001054  175374  6$:     MOV     #1054,$TMP1     ;BAD STOP BIT  FOR FRAME ERROR
1465  004104  012700  120026                  MOV     #RXERR!FRMERR!26,R0     ;EXPECTED
1466  004110  000167  177642                  JMP     3$      ;DO IT AGAIN
1467  004114  012767  000054  175356  7$:     MOV     #54,$TMP1       ;BAD STOP BIT & PARITY
1468  004122  012700  130026                  MOV     #RXERR!FRMERR!PARER!26,R0       ;EXPECTED
1469  004126  000167  177624                  JMP     3$      ;DO IT AGAIN
1470  004132                          5$:
1471                                  ;;THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER RESET"
1472                                  ;;WHILE IN STRIP SYNC MODE
1473                                  ;;THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
1474                                  ;;STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
1475                                  ;;.....BUT IF AN ERROR SHOULD OCCUR....THIS AUTOMATIC RESET
1476                                  ;;IS DISCOMBOBULATED
1477                                  ;;IE. FORCE PARITY ERROR WHILE STRIP SYNC IS SET
1478                                  ;;NOTE: NORMALLY THE LOGIC RESETS THE RXDONE &ERROR FLAGS
1479                                  ;;PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT....
1480                                  ;;BUT, IF AN RXERR OCCURS   RXDONE PLUS RXERR ARE ASSERTED
1481                                  ;;MODE: ISOC (ISYMOD)
1482                                  ;;LENGTH: SEVEN
1483                                  ;;PARITY: EVEPAR
1484                                  ;;CHARACTER EXPECTED: 226
1485                                                  ;NOTE THAT THE PARITY BIT SHOULD SHOW
1486                                                  ;UP IN THE DATA     IE. BIT SEVEN FOR
1487                                                  ;SEVEN LEVEL CODE
1488                                  ;;CHARACTER SENT: SYNC CHARACTER
1489                                  ;;NOTE: THIS TEST USES ONLY THE RECEIVER LOGIC
1490                                  ;;
1491                                  ;;*****************************************************************
1492  004132  000004                  TST3:   SCOPE
1493  004134  052777  000400  175564          BIS     #MRESET,@TXCSR  ;MASTER RESET
1494  004142  012777  000000  175552          MOV     #ISYMOD,@PARCSR ;SET THE MODE
1495  004150  052777  000400  175550          BIS     #MRESET,@TXCSR  ;MASTER RESET
1496
1497                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1498  004156  012777  064001  175542          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1499
1500                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1501  004164  012777  005626  175530          MOV     #ISYMOD!SEVEN!EVEPAR!226,@PARCSR
1502  004172  016703  175520                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MSG
1503  004176  012767  000003  174720          MOV     #3,COUNT        ;# OF TIMES SYNC CHAR WILL BE SENT
1504  004204  052777  000020  175500          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1505                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1506  004212  042777  020000  175506          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1507  004220  052777  020000  175500          BIS     #CLK,@TXCSR     ;POKE CLK UP
1508                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1509  004226  042777  020000  175472          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1510  004234  052777  020000  175464          BIS     #CLK,@TXCSR     ;POKE CLK UP
1511  004242  052777  000400  175442          BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
1512  004250  012767  000012  174644  2$:     MOV     #10.,SHIFT      ;# OF SHIFTS
```

```
1513  004256  012767  001454  175214          MOV     #1454,$TMP1     ;SYNC CHAR + START&STOP+ PARITY
1514  004264  004767  012656          1$:     JSR     PC,RPOKE        ;SHIFT IN THIS CHARACTER
1515  004270  105777  175416                  TSTB    @RXCSR  ;RXDONE = 0 ?
1516  004274  100001                          BPL     .+4
1517  004276  104004                          ERROR   4       ;RXDONE SHOULD NOT BE SET
1518  004300  005367  174620                  DEC     COUNT   ;# OF  SYNC CHARS
1519  004304  001361                          BNE     2$      ;GO AGAIN ?
1520  004306  012700  000226                  MOV     #226,R0 ;EXPECTED
1521  004312  017701  175400                  MOV     @RXDBUF,R1      ;ACTUAL
1522                                                          ;NOTE THAT THIS IS THE FIRST TIME
1523                                                          ;RXDBUF IS READ.....THERE SHOULD BE
1524                                                          ;NO OVER RUN ERROR      4S
1525  004316  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
1526  004320  001401                          BEQ     .+4
1527  004322  104002                          ERROR   2       ;DATA CHARS SHOULD COMPARE
1528                                                          ;THERE SHOULD BE NO RXERR'S
1529  004324  012767  000004  174572          MOV     #4,COUNT        ;# OF TIMES
1530  004332  012700  110026                  MOV     #RXERR!PARER!26,R0      ;EXPECTED
1531  004336  012767  001054  175134          MOV     #1054,$TMP1     ;BAD SYNC CHAR (WRONG PARITY)
1532  004344  012767  000012  174550  3$:     MOV     #10.,SHIFT      ;# OF SHIFTS
1533  004352  004767  012570                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1534  004356  105777  175330                  TSTB    @RXCSR  ;RXDONE = 1?
1535  004362  100401                          BMI     .+4
1536  004364  104004                          ERROR   4       ;RXDONE SHOULD BE SET
1537  004366  017701  175324                  MOV     @RXDBUF,R1      ;ACTUAL DATA
1538  004372  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
1539  004374  001401                          BEQ     .+4
1540  004376  104000                          EPROR   ;DID THE RESPECTIVE ERROR       4 STOP THE
1541                                                          ;AUTOMATIC RESSETING OF RXDONE & ERROR FLAGS
1542                                                          ;.....CHECK THIS.....
1543                                                          ;NOTE THAT THE PARITY BIT SHOULD
1544                                                          ;SHOW UP IN THE DATA
1545                                                          ;IE.  BIT SEVEN FOR SEVEN LEVEL CODE
1546  004400  005367  174520                  DEC     COUNT   ;# OF SYNC CHARS
1547  004404  001445                          BEQ     5$      ;FINISHED ? GET OUT OF TEST
1548  004406  022767  000003  174510          CMP     #3,COUNT        ;# OF SYNC CHARS
1549  004414  001423                          BEQ     6$      ;CHECK FRAME ERROR ?
1550  004416  022767  000002  174500          CMP     #2,COUNT        ;# OF SYNC CHARS
1551  004424  001426                          BEQ     7$      ;CHECK FRAME ERROR & BAD PARITY ?
1552                                          ;NOPE THEN IT (COUNT) MUST BE = 1  THEREFORE....
1553  004426  012767  000012  174466          MOV     #10.,SHIFT      ;# OF SHIFTS
1554  004434  012767  000054  175036          MOV     #54,$TMP1       ;FRAME & PARITY ERROR
1555  004442  004767  012500                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1556                                  ;NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
1557  004446  012767  000054  175024          MOV     #54,$TMP1       ;FRAME & PARITY ERROR
1558  004454  012700  170026                  MOV     #RXERR!OVRRUN!FRMERR!PARER!26,R0        ;EXPECTED
1559  004460  000167  177660                  JMP     3$      ;DO IT AGAIN
1560  004464  012767  000454  175006  6$:     MOV     #454,$TMP1      ;BAD STOP BIT  FOR FRAME ERROR
1561  004472  012700  120226                  MOV     #RXERR!FRMERR!226,R0    ;EXPECTED
1562  004476  000167  177642                  JMP     3$      ;DO IT AGAIN
1563  004502  012767  000054  174770  7$.     MOV     #54,$TMP1       ;BAD STOP BIT & PARITY
1564  004510  012700  130026                  MOV     #RXERR!FRMERR!PARER!26,R0       ;EXPECTED
1565  004514  000167  177624                  JMP     3$      ;DO IT AGAIN
1566  004520                          5$:
1567                                  ;;THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER RESET"
1568                                  ;;WHILE IN STRIP SYNC MODE
```

```
                                        ;;THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
                                        ;;STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
                                        ;;.....BUT IF AN ERROR SHOULD OCCUR.... THIS AUTOMATIC RESET
                                        ;;IS DISCOMBOBULATED
                                        ;;IE. FORCE PARITY ERROR WHILE STRIP SYNC IS SET
                                        ;;NOTE: NORMALLY THE LOGIC RESETS THE RXDONE &ERROR FLAGS
                                        ;;PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT....
                                        ;;BUT, IF AN RXERR OCCURS   RXDONE PLUS RXERR ARE ASSERTED
                                        ;;MODE: ISOC (ISYMOD)
                                        ;;LENGTH: SIX
                                        ;;PARITY: EVEPAR
                                        ;;CHARACTER EXPECTED: 126
                                                        ;NOTE THAT THE PARITY BIT SHOULD SHOW
                                                        ;UP IN THE DATA     IE. BIT SIX FOR
                                                        ;SIX LEVEL CODE
                                        ;;CHARACTER SENT: SYNC CHARACTER
                                        ;;NOTE: THIS TEST USES ONLY THE RECEIVER LOGIC
                                        ;;
                                        ;;*****************************************************************
004520  000004                        TST4:   SCOPE
004522  052777  000400  175176                BIS     #MRESET,@TXCSR  ;MASTER RESET
004530  012777  000000  175164                MOV     #ISYMOD,@PARCSR ;SET THE MODE
004536  052777  000400  175162                BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
004544  012777  064001  175154                MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
004552  012777  003526  175142                MOV     #ISYMOD!SIX!EVEPAR!126,@PARCSR
004560  016703  175132                       MOV     RXDBUF,R3       ;SET UP FOR ERROR MSG
004564  012767  000003  174332                MOV     #3,COUNT        ;# OF TIMES SYNC CHAR WILL BE SENT
004572  052777  000020  175112                BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                               ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
004600  042777  020000  175120                BIC     #CLK,@TXCSR     ;POKE CLK DOWN
004606  052777  020000  175112                BIS     #CLK,@TXCSR     ;POKE CLK UP
                                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
004614  042777  020000  175104                BIC     #CLK,@TXCSR     ;POKE CLK DOWN
004622  052777  020000  175076                BIS     #CLK,@TXCSR     ;POKE CLK UP
004630  052777  000400  175054                BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
004636  012767  000011  174256  2$:           MOV     #9.,SHIFT       ;# OF SHIFTS
004644  012767  000654  174626                MOV     #654,$TMP1      ;SYNC CHAR + START&STOP+ PARITY
004652  004767  012270            1$:         JSR     PC,RPOKE        ;SHIFT IN THIS CHARACTER
004656  105777  175030                       TSTB    @RXCSR  ;RXDONE = 0 ?
004662  100001                              BPL     .+4
004664  104004                              ERROR   4       ;RXDONE SHOULD NOT BE SET
004666  005367  174232                       DEC     COUNT   ;# OF  SYNC CHARS
004672  001361                              BNE     2$      ;GO AGAIN ?
004674  012700  000126                       MOV     #126,R0 ;EXPECTED
004700  017701  175012                       MOV     @RXDBUF,R1      ;ACTUAL
                                                        ;NOTE THAT THIS IS THE FIRST TIME
                                                        ;RXDBUF IS READ..... THERE SHOULD BE
                                                        ;NO OVER RUN ERROR      4S
004704  020001                              CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
004706  001401                              BEQ     .+4
004710  104002                              ERROR   2       ;DATA CHARS SHOULD COMPARE
                                                        ;THERE SHOULD BE NO RXERR'S
```

```
1625  004712  012767  000004  174204          MOV     #4,COUNT        ;# OF TIMES
1626  004720  012700  110026                  MOV     #RXERR!PARER!26,R0      ;EXPECTED
1627  004724  012767  000454  174546          MOV     #454,$TMP1      ;BAD SYNC CHAR (WRONG PARITY)
1628  004732  012767  000011  174162  3$:     MOV     #9.,SHIFT       ;# OF SHIFTS
1629  004740  004767  012202                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1630  004744  105777  174742                  TSTB    @RXCSR  ;RXDONE = 1?
1631  004750  100401                          BMI     .+4
1632  004752  104004                          ERROR   4       ;RXDONE SHOULD BE SET
1633  004754  017701  174736                  MOV     @RXDBUF,R1      ;ACTUAL DATA
1634  004760  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
1635  004762  001401                          BEQ     .+4
1636  004764  104000                          ERROR   ;DID THE RESPECTIVE ERROR       4 STOP THE
1637                                          ;AUTOMATIC RESSETING OF RXDONE & ERROR FLAGS
1638                                          ;.....CHECK THIS.....
1639                                          ;NOTE THAT THE PARITY BIT SHOULD
1640                                          ;SHOW UP IN THE DATA
1641                                          ;IE.  BIT SIX FOR SIX LEVEL CODE
1642  004766  005367  174132                  DEC     COUNT   ;# OF SYNC CHARS
1643  004772  001445                          BEQ     5$      ;FINISHED ? GET OUT OF TEST
1644  004774  022767  000003  174122          CMP     #3,COUNT        ;# OF SYNC CHARS
1645  005002  001423                          BEQ     6$      ;CHECK FRAME ERROR ?
1646  005004  022767  000002  174112          CMP     #2,COUNT        ;# OF SYNC CHARS
1647  005012  001426                          BEQ     7$      ;CHECK FRAME ERROR & BAD PARITY ?
1648                                  ;NOPE THEN IT (COUNT) MUST BE = 1  THEREFORE....
1649  005014  012767  000011  174100          MOV     #9.,SHIFT       ;# OF SHIFTS
1650  005022  012767  000054  174450          MOV     #54,$TMP1       ;FRAME & PARITY ERROR
1651  005030  004767  012112                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1652                                  ;NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
1653  005034  012767  000054  174436          MOV     #54,$TMP1       ;FRAME & PARITY ERROR
1654  005042  012700  170026                  MOV     #RXERR!OVRRUN!FRMERR!PARER!26,R0        ;EXPECTED
1655  005046  000167  177660                  JMP     3$      ;DO IT AGAIN
1656  005052  012767  000254  174420  6$:     MOV     #254,$TMP1      ;BAD STOP BIT  FOR FRAME ERROR
1657  005060  012700  120126                  MOV     #RXERR!FRMERR!126,R0    ;EXPECTED
1658  005064  000167  177642                  JMP     3$      ;DO IT AGAIN
1659  005070  012767  000054  174402  7$:     MOV     #54,$TMP1       ;BAD STOP BIT & PARITY
1660  005076  012700  130026                  MOV     #RXERR!FRMERR!PARER!26,R0       ;EXPECTED
1661  005102  000167  177624                  JMP     3$      ;DO IT AGAIN
1662  005106                          5$:
1663                                  ;;THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER RESET"
1664                                  ;;WHILE IN STRIP SYNC MODE
1665                                  ;;THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
1666                                  ;;STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
1667                                  ;;.....BUT IF AN ERROR SHOULD OCCUR.... THIS AUTOMATIC RESET
1668                                  ;;IS DISCOMBOBULATED
1669                                  ;;IE. FORCE PARITY ERROR WHILE STRIP SYNC IS SET
1670                                  ;;NOTE: NORMALLY THE LOGIC RESETS THE RXDONE &ERROR FLAGS
1671                                  ;;PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT....
1672                                  ;;BUT, IF AN RXERR OCCURS   RXDONE PLUS RXERR ARE ASSERTED
1673                                  ;;MODE:  ISOC (ISYMOD)
1674                                  ;;LENGTH:  FIVE
1675                                  ;;PARITY:  EVEPAR
1676                                  ;;CHARACTER EXPECTED: 66
1677                                                  ;NOTE THAT THE PARITY BIT SHOULD SHOW
1678                                                  ;UP IN THE DATA     IE. BIT FIVE FOR
1679                                                  ;FIVE LEVEL CODE
1680                                  ;;CHARACTER SENT:  SYNC CHARACTER
```

INITIALIZE THE COMMON TAGS SEQ 0036

```
1681                                    ;;NOTE: THIS TEST USES ONLY THE RECEIVER LOGIC
1682                                    ;;
1683                                    ;;*****************************************************************
1684  005106  000004                    TST5:   SCOPE
1685  005110  052777  000400  174610            BIS     #MRESET,@TXCSR  ;MASTER RESET
1686  005116  012777  000000  174576            MOV     #ISYMOD,@PARCSR ;SET THE MODE
1687  005124  052777  000400  174574            BIS     #MRESET,@TXCSR  ;MASTER RESET
1688
1689                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1690  005132  012777  064001  174566            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1691
1692                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1693  005140  012777  001466  174554            MOV     #ISYMOD!FIVE!EVEPAR!66,@PARCSR
1694  005146  016703  174544                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MSG
1695  005152  012767  000003  173744            MOV     #3,COUNT        ;# OF TIMES SYNC CHAR WILL BE SENT
1696  005160  052777  000020  174524            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1697                                            ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1698  005166  042777  020000  174532            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1699  005174  052777  020000  174524            BIS     #CLK,@TXCSR     ;POKE CLK UP
1700                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1701  005202  042777  020000  174516            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1702  005210  052777  020000  174510            BIS     #CLK,@TXCSR     ;POKE CLK UP
1703  005216  052777  000400  174466            BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
1704  005224  012767  000010  173670  2$:       MOV     #8.,SHIFT       ;# OF SHIFTS
1705  005232  012767  000354  174240            MOV     #354,$TMP1      ;SYNC CHAR + START&STOP+ PARITY
1706  005240  004767  011702          1$:       JSR     PC,RPOKE        ;SHIFT IN THIS CHARACTER
1707  005244  105777  174442                    TSTB    @RXCSR  ;RXDONE = 0 ?
1708  005250  100001                            BPL     .+4
1709  005252  104004                            ERROR   4       ;RXDONE SHOULD NOT BE SET
1710  005254  005367  173644                    DEC     COUNT   ;# OF  SYNC CHARS
1711  005260  001361                            BNE     2$      ;GO AGAIN ?
1712  005262  012700  000066                    MOV     #66,R0  ;EXPECTED
1713  005266  017701  174424                    MOV     @RXDBUF,R1      ;ACTUAL
1714                                            ;NOTE THAT THIS IS THE FIRST TIME
1715                                            ;RXDBUF IS READ.....THERE SHOULD BE
1716                                            ;NO OVER RUN ERROR      4S
1717  005272  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
1718  005274  001401                            BEQ     .+4
1719  005276  104002                            ERROR   2       ;DATA CHARS SHOULD COMPARE
1720                                            ;THERE SHOULD BE NO RXERR'S
1721  005300  012767  000004  173616            MOV     #4,COUNT        ;# OF TIMES
1722  005306  012700  110026                    MOV     #RXERR!PARER!26,R0      ;EXPECTED
1723  005312  012767  000254  174160            MOV     #254,$TMP1      ;BAD SYNC CHAR (WRONG PARITY)
1724  005320  012767  000010  173574  3$:       MOV     #8.,SHIFT       ;# OF SHIFTS
1725  005326  004767  011614                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1726  005332  105777  174354                    TSTB    @RXCSR  ;RXDONE = 1?
1727  005336  100401                            BMI     .+4
1728  005340  104004                            ERROR   4       ;RXDONE SHOULD BE SET
1729  005342  017701  174350                    MOV     @RXDBUF,R1      ;ACTUAL DATA
1730  005346  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
1731  005350  001401                            BEQ     .+4
1732  005352  104000                            ERROR   ;DID THE RESPECTIVE ERROR       4 STOP THE
1733                                            ;AUTOMATIC RESSETING OF RXDONE & ERROR FLAGS
1734                                            ;.....CHECK THIS.....
1735                                            ;NOTE THAT THE PARITY BIT SHOULD
1736                                            ;SHOW UP IN THE DATA
```

```
1737                                                    ;IE.  BIT FIVE FOR FIVE LEVEL CODE
1738  005354  005367  173544                  DEC     COUNT   ;# OF SYNC CHARS
1739  005360  001445                          BEQ     5$      ;FINISHED ? GET OUT OF TEST
1740  005362  022767  000003  173534          CMP     #3,COUNT        ;# OF SYNC CHARS
1741  005370  001423                          BEQ     6$      ;CHECK FRAME ERROR ?
1742  005372  022767  000002  173524          CMP     #2,COUNT        ;# OF SYNC CHARS
1743  005400  001426                          BEQ     7$      ;CHECK FRAME ERROR & BAD PARITY ?
1744                                          ;NOPE THEN IT (COUNT) MUST BE = 1  THEREFORE....
1745  005402  012767  000010  173512          MOV     #8.,SHIFT       ;# OF SHIFTS
1746  005410  012767  000054  174062          MOV     #54,$TMP1       ;FRAME & PARITY ERROR
1747  005416  004767  011524                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1748                                  ;NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
1749  005422  012767  000054  174050          MOV     #54,$TMP1       ;FRAME & PARITY ERROR
1750  005430  012700  170026                  MOV     #RXERR!OVRRUN!FRMERR!PARER!26,R0        ;EXPECTED
1751  005434  000167  177660                  JMP     3$      ;DO IT AGAIN
1752  005440  012767  000154  174032  6$:     MOV     #154,$TMP1      ;BAD STOP BIT  FOR FRAME ERROR
1753  005446  012700  120066                  MOV     #RXERR!FRMERR!66,R0     ;EXPECTED
1754  005452  000167  177642                  JMP     3$      ;DO IT AGAIN
1755  005456  012767  000054  174014  7$:     MOV     #54,$TMP1       ;BAD STOP BIT & PARITY
1756  005464  012700  130026                  MOV     #RXERR!FRMERR!PARER!26,R0       ;EXPECTED
1757  005470  000167  177624                  JMP     3$      ;DO IT AGAIN
1758  005474                          5$:
1759                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF
1760                                          ;;THE TRANSMITTER SECTION, IT USES THE DNA FLAG
1761                                          ;;AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1762                                          ;;CORRECTLY
1763                                          ;;NOTE:  DNA COMES UP ON THE FIRST RISING BIT
1764                                          ;;EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1765                                          ;;LOADED INTO TXDBUF
1766                                          ;;MODE: SYNINT
1767                                          ;;PARITY: NO PARITY
1768                                          ;;LENGTH: FIVE
1769                                          ;;
1770                                  ;;*****************************************************************
1771  005474  000004                  TST6:   SCOPE
1772  005476  052777  000400  174222          BIS     #MRESET,@TXCSR  ;MASTER RESET
1773  005504  012777  030000  174210          MOV     #SYNINT,@PARCSR ;SET THE MODE
1774  005512  052777  000400  174206          BIS     #MRESET,@TXCSR  ;MASTER RESET
1775
1776                                  ;SET MAINTENANCE MODE & SEND
1777                                          ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1778  005520  012777  004020  174200          MOV     #MINT!SEND,@TXCSR
1779
1780                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1781  005526  012777  030026  174166          MOV     #SYNINT!FIVE!NOPAR!26,@PARCSR
1782  005534  016703  174166                  MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
1783  005540  112777  000021  174164          MOVB    #21,@TXDBUF     ;LOAD CHAR
1784  005546  012767  000021  173724          MOV     #21,$TMP1       ;SHIFTED CHAR
1785  005554  012767  000005  173340          MOV     #5,SHIFT        ;# OF SHIFTS
1786                                  ;POKE CLK TO GET INTO SYNCRONIZATION
1787  005562  052777  020000  174136          BIS     #CLK,@TXCSR     ;POKE CLK UP
1788  005570  042777  020000  174130          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1789  005576  005000                  1$:     CLR     R0
1790  005600  006067  173674                  ROR     $TMP1   ;FORCE CARRY
1791  005604  103002                          BCC     2$
1792  005606  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BIT WINDOW
```

```
1793  005612                                  2$:
1794  005612  052777  020000  174106          BIS     #CLK,@TXCSR     ;POKE CLK UP
1795  005620  042777  020000  174100          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1796  005626  017701  174074                  MOV     @TXCSR,R1       ;ACTUAL
1797  005632  042701  075777                  BIC     #075777,R1      ;SAVE BITW & DNA
1798  005636  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
1799  005640  001401                          BEQ     .+4
1800  005642  104003                          ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1801                                                          ;BIT,....ALSO CHECK DNA
1802  005644  005367  173252                  DEC     SHIFT   ;# OF SHIFTS
1803  005650  001352                          BNE     1$      ;DO IT AGAIN ?
1804                                  ;NOW POKE CLK TO SEE DNA
1805  005652  052777  020000  174046          BIS     #CLK,@TXCSR     ;POKE CLK
1806  005660  012700  100000                  MOV     #100000,R0      ;EXPECTED
1807  005664  017701  174036                  MOV     @TXCSR,R1       ;ACTUAL
1808  005670  042701  077777                  BIC     #77777,R1       ;SAVE DNA ONLY
1809  005674  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
1810  005676  001401                          BEQ     .+4
1811  005700  104003                          ERROR   3       ;DNA SHOULD BE SET
1812                                                          ;IF DNA DID NOT SET ,CHECK WORD LENGTH
1813                                                          ;SELECT LOGIC OF THE TRANSMITTER
1814  005702  005777  174020                  TST     @TXCSR  ;DNA ?
1815  005706  100001                          BPL     .+4
1816  005710  104004                          ERROR   4               ;DNA SHOULD NOT BE SET
1817                                                          ;IT SHOULD HAVE BEEN CLEARED FROM
1818                                                          ;PREVIOUS READ
1819
1820                                                  ;;THIS TEST VERIFYS WORD LENGTH SELECT OF
1821                                                  ;;THE TRANSMITTER SECTION,IT USES THE DNA FLAG
1822                                                  ;;AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1823                                                  ;;CORRECTLY
1824                                                  ;;NOTE: DNA COMES UP ON THE FIRST RISING BIT
1825                                                  ;;EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1826                                                  ;;LOADED INTO TXDBUF
1827                                                  ;;MODE: SYNINT
1828                                                  ;;PARITY: NO PARITY
1829                                                  ;;LENGTH: SIX
1830                                                  ;;
1831                                  ;;**************************************************************************
1832  005712  000004                  TST7:   SCOPE
1833  005714  052777  000400  174004          BIS     #MRESET,@TXCSR  ;MASTER RESET
1834  005722  012777  030000  173772          MOV     #SYNINT,@PARCSR ;SET THE MODE
1835  005730  052777  000400  173770          BIS     #MRESET,@TXCSR  ;MASTER RESET
1836
1837                                  ;SET MAINTENANCE MODE & SEND
1838                                          ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1839  005736  012777  004020  173762          MOV     #MINT!SEND,@TXCSR
1840
1841                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1842  005744  012777  032026  173750          MOV     #SYNINT!SIX!NOPAR!26,@PARCSR
1843  005752  016703  173750                  MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
1844  005756  112777  000021  173746          MOVB    #21,@TXDBUF     ;LOAD CHAR
1845  005764  012767  000021  173506          MOV     #21,$TMP1       ;SHIFTED CHAR
1846  005772  012767  000006  173122          MOV     #6,SHIFT        ;# OF SHIFTS
1847                                  ;POKE CLK TO GET INTO SYNCRONIZATION
1848  006000  052777  020000  173720          BIS     #CLK,@TXCSR     ;POKE CLK UP
```

```
1849  006006  042777  020000  173712          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1850  006014  005000                  1$:     CLR     R0
1851  006016  006067  173456                  ROR     $TMP1   ;FORCE CARRY
1852  006022  103002                          BCC     2$
1853  006024  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BIT WINDOW
1854  006030                          2$:
1855  006030  052777  020000  173670          BIS     #CLK,@TXCSR     ;POKE CLK UP
1856  006036  042777  020000  173662          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1857  006044  017701  173656                  MOV     @TXCSR,R1       ;ACTUAL
1858  006050  042701  075777                  BIC     #075777,R1      ;SAVE BITW & DNA
1859  006054  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
1860  006056  001401                          BEQ     .+4
1861  006060  104003                          ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1862                                                          ;BIT,....ALSO CHECK DNA
1863  006062  005367  173034                  DEC     SHIFT   ;# OF SHIFTS
1864  006066  001352                          BNE     1$      ;DO IT AGAIN ?
1865                                  ;NOW POKE CLK TO SEE DNA
1866  006070  052777  020000  173630          BIS     #CLK,@TXCSR     ;POKE CLK
1867  006076  012700  100000                  MOV     #100000,R0      ;EXPECTED
1868  006102  017701  173620                  MOV     @TXCSR,R1       ;ACTUAL
1869  006106  042701  077777                  BIC     #77777,R1       ;SAVE DNA ONLY
1870  006112  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
1871  006114  001401                          BEQ     .+4
1872  006116  104003                          ERROR   3       ;DNA SHOULD BE SET
1873                                                          ;IF DNA DID NOT SET ,CHECK WORD LENGTH
1874                                                          ;SELECT LOGIC OF THE TRANSMITTER
1875  006120  005777  173602                  TST     @TXCSR  ;DNA ?
1876  006124  100001                          BPL     .+4
1877  006126  104004                          ERROR   4               ;DNA SHOULD NOT BE SET
1878                                                          ;IT SHOULD HAVE BEEN CLEARED FROM
1879                                                          ;PREVIOUS READ
1880
1881                                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF
1882                                                          ;;THE TRANSMITTER SECTION,IT USES THE DNA FLAG
1883                                                          ;;AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1884                                                          ;;CORRECTLY
1885                                                          ;;NOTE: DNA COMES UP ON THE FIRST RISING BIT
1886                                                          ;;EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1887                                                          ;;LOADED INTO TXDBUF
1888                                                          ;;MODE: SYNINT
1889                                                          ;;PARITY: NO PARITY
1890                                                          ;;LENGTH: SEVEN
1891                                                          ;;
1892                                  ;;**************************************************************************
1893  006130  000004                  TST10:  SCOPE
1894  006132  052777  000400  173566          BIS     #MRESET,@TXCSR  ;MASTER RESET
1895  006140  012777  030000  173554          MOV     #SYNINT,@PARCSR ;SET THE MODE
1896  006146  052777  000400  173552          BIS     #MRESET,@TXCSR  ;MASTER RESET
1897
1898                                  ;SET MAINTENANCE MODE & SEND
1899                                          ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1900  006154  012777  004020  173544          MOV     #MINT!SEND,@TXCSR
1901
1902                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1903  006162  012777  034026  173532          MOV     #SYNINT!SEVEN!NOPAR!26,@PARCSR
1904  006170  016703  173532                  MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
```

```
1905  006174  112777  000021  173530          MOVB    #21,@TXDBUF     ;LOAD CHAR
1906  006202  012767  000021  173270          MOV     #21,$TMP1       ;SHIFTED CHAR
1907  006210  012767  000007  172704          MOV     #7,SHIFT        ;# OF SHIFTS
1908                                  ;POKE CLK TO GET INTO SYNCRONIZATION
1909  006216  052777  020000  173502          BIS     #CLK,@TXCSR     ;POKE CLK UP
1910  006224  042777  020000  173474          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1911  006232  005000                  1$:     CLR     R0
1912  006234  006067  173240                  ROR     $TMP1   ;FORCE CARRY
1913  006240  103002                          BCC     2$
1914  006242  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BIT WINDOW
1915  006246                          2$:
1916  006246  052777  020000  173452          BIS     #CLK,@TXCSR     ;POKE CLK UP
1917  006254  042777  020000  173444          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1918  006262  017701  173440                  MOV     @TXCSR,R1       ;ACTUAL
1919  006266  042701  075777                  BIC     #075777,R1      ;SAVE BITW & DNA
1920  006272  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
1921  006274  001401                          BEQ     .+4
1922  006276  104003                          ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1923                                                          ;BIT,....ALSO CHECK DNA
1924  006300  005367  172616                  DEC     SHIFT   ;# OF SHIFTS
1925  006304  001352                          BNE     1$      ;DO IT AGAIN ?
1926                                  ;NOW POKE CLK TO SEE DNA
1927  006306  052777  020000  173412          BIS     #CLK,@TXCSR     ;POKE CLK
1928  006314  012700  100000                  MOV     #100000,R0      ;EXPECTED
1929  006320  017701  173402                  MOV     @TXCSR,R1       ;ACTUAL
1930  006324  042701  077777                  BIC     #77777,R1       ;SAVE DNA ONLY
1931  006330  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
1932  006332  001401                          BEQ     .+4
1933  006334  104003                          ERROR   3       ;DNA SHOULD BE SET
1934                                                          ;IF DNA DID NOT SET ,CHECK WORD LENGTH
1935                                                          ;SELECT LOGIC OF THE TRANSMITTER
1936  006336  005777  173364                  TST     @TXCSR  ;DNA ?
1937  006342  100001                          BPL     .+4
1938  006344  104004                          ERROR   4               ;DNA SHOULD NOT BE SET
1939                                                          ;IT SHOULD HAVE BEEN CLEARED FROM
1940                                                          ;PREVIOUS READ
1941
1942                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF
1943                                          ;;THE TRANSMITTER SECTION, IT USES THE DNA FLAG
1944                                          ;;AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1945                                          ;;CORRECTLY
1946                                          ;;NOTE:  DNA COMES UP ON THE FIRST RISING BIT
1947                                          ;;EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1948                                          ;;LOADED INTO TXDBUF
1949                                          ;;MODE: SYNINT
1950                                          ;;PARITY: NO PARITY
1951                                          ;;LENGTH: EIGHT
1952                                          ;;
1953                                  ;;*****************************************************************
1954  006346  000004                  TST11:  SCOPE
1955  006350  052777  000400  173350          BIS     #MRESET,@TXCSR  ;MASTER RESET
1956  006356  012777  030000  173336          MOV     #SYNINT,@PARCSR ;SET THE MODE
1957  006364  052777  000400  173334          BIS     #MRESET,@TXCSR  ;MASTER RESET
1958
1959                                  ;SET MAINTENANCE MODE & SEND
1960                                          ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
```

```
                                        MOV     #MINT!SEND,@TXCSR

                                ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
                                        MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
                                        MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
                                        MOVB    #21,@TXDBUF     ;LOAD CHAR
                                        MOV     #21,$TMP1       ;SHIFTED CHAR
                                        MOV     #8.,SHIFT       ;# OF SHIFTS
                                ;POKE CLK TO GET INTO SYNCRONIZATION
                                        BIS     #CLK,@TXCSR     ;POKE CLK UP
                                        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
                                1$:     CLR     R0
                                        ROR     $TMP1   ;FORCE CARRY
                                        BCC     2$
                                        BIS     #BITW,R0        ;EQUIV OF BIT WINDOW
                                2$:
                                        BIS     #CLK,@TXCSR     ;POKE CLK UP
                                        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
                                        MOV     @TXCSR,R1       ;ACTUAL
                                        BIC     #075777,R1      ;SAVE BITW & DNA
                                        CMP     R0,R1   ;COMPARE EXP VS ACT
                                        BEQ     .+4
                                        ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
                                                        ;BIT,....ALSO CHECK DNA
                                        DEC     SHIFT   ;# OF SHIFTS
                                        BNE     1$      ;DO IT AGAIN ?
                                ;NOW POKE CLK TO SEE DNA
                                        BIS     #CLK,@TXCSR     ;POKE CLK
                                        MOV     #100000,R0      ;EXPECTED
                                        MOV     @TXCSR,R1       ;ACTUAL
                                        BIC     #77777,R1       ;SAVE DNA ONLY
                                        CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
                                        BEQ     .+4
                                        ERROR   3       ;DNA SHOULD BE SET
                                                        ;IF DNA DID NOT SET ,CHECK WORD LENGTH
                                                        ;SELECT LOGIC OF THE TRANSMITTER
                                        TST     @TXCSR  ;DNA ?
                                        BPL     .+4
                                        ERROR   4               ;DNA SHOULD NOT BE SET
                                                        ;IT SHOULD HAVE BEEN CLEARED FROM
                                                        ;PREVIOUS READ

                                        ;;THIS TEST VERIFYS WORD LENGTH SELECT OF
                                        ;;THE TRANSMITTER SECTION,IT USES THE DNA FLAG
                                        ;;AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
                                        ;;CORRECTLY
                                        ;;NOTE: DNA COMES UP ON THE FIRST RISING BIT
                                        ;;EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
                                        ;;LOADED INTO TXDBUF
                                        ;;MODE: SYNEXT
                                        ;;PARITY: NO PARITY
                                        ;;LENGTH: FIVE
                                        ;;
                                ;;**************************************************************
                                TST12:  SCOPE
                                        BIS     #MRESET,@TXCSR  ;MASTER RESET
```

```
2017  006574  012777  020000  173120          MOV     #SYNEXT,@PARCSR ;SET THE MODE
2018  006602  052777  000400  173116          BIS     #MRESET,@TXCSR  ;MASTER RESET
2019
2020                                  ;SET MAINTENANCE MODE & SEND
2021                                          ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2022  006610  012777  004020  173110          MOV     #MINT!SEND,@TXCSR
2023
2024                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2025  006616  012777  020026  173076          MOV     #SYNEXT!FIVE!NOPAR!26,@PARCSR
2026  006624  016703  173076                  MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
2027  006630  112777  000021  173074          MOVB    #21,@TXDBUF     ;LOAD CHAR
2028  006636  012767  000021  172634          MOV     #21,$TMP1       ;SHIFTED CHAR
2029  006644  012767  000005  172250          MOV     #5,SHIFT        ;# OF SHIFTS
2030                                  ;POKE CLK TO GET INTO SYNCRONIZATION
2031  006652  052777  020000  173046          BIS     #CLK,@TXCSR     ;POKE CLK UP
2032  006660  042777  020000  173040          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2033  006666  005000                  1$:     CLR     R0
2034  006670  006067  172604                  ROR     $TMP1   ;FORCE CARRY
2035  006674  103002                          BCC     2$
2036  006676  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BIT WINDOW
2037  006702                          2$:
2038  006702  052777  020000  173016          BIS     #CLK,@TXCSR     ;POKE CLK UP
2039  006710  042777  020000  173010          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2040  006716  017701  173004                  MOV     @TXCSR,R1       ;ACTUAL
2041  006722  042701  075777                  BIC     #075777,R1      ;SAVE BITW & DNA
2042  006726  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
2043  006730  001401                          BEQ     .+4
2044  006732  104003                          ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2045                                                          ;BIT.....ALSO CHECK DNA
2046  006734  005367  172162                  DEC     SHIFT   ;# OF SHIFTS
2047  006740  001352                          BNE     1$      ;DO IT AGAIN ?
2048                                  ;NOW POKE CLK TO SEE DNA
2049  006742  052777  020000  172756          BIS     #CLK,@TXCSR     ;POKE CLK
2050  006750  012700  100000                  MOV     #100000,R0      ;EXPECTED
2051  006754  017701  172746                  MOV     @TXCSR,R1       ;ACTUAL
2052  006760  042701  077777                  BIC     #77777,R1       ;SAVE DNA ONLY
2053  006764  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
2054  006766  001401                          BEQ     .+4
2055  006770  104003                          ERROR   3       ;DNA SHOULD BE SET
2056                                                          ;IF DNA DID NOT SET ,CHECK WORD LENGTH
2057                                                          ;SELECT LOGIC OF THE TRANSMITTER
2058  006772  005777  172730                  TST     @TXCSR  ;DNA ?
2059  006776  100001                          BPL     .+4
2060  007000  104004                          ERROR   4               ;DNA SHOULD NOT BE SET
2061                                                          ;IT SHOULD HAVE BEEN CLEARED FROM
2062                                                          ;PREVIOUS READ
2063
2064                                  ;;THIS TEST VERIFYS WORD LENGTH SELECT OF
2065                                  ;;THE TRANSMITTER SECTION,IT USES THE DNA FLAG
2066                                  ;;AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
2067                                  ;;CORRECTLY
2068                                  ;;NOTE:  DNA COMES UP ON THE FIRST RISING BIT
2069                                  ;;EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
2070                                  ;;LOADED INTO TXDBUF
2071                                  ;;MODE: SYNEXT
2072                                  ;;PARITY: NO PARITY
```

```
                                        ;;LENGTH: SIX
                                        ;;
                                ;;**************************************************************
007002  000004                TST13:  SCOPE
007004  052777  000400  172714        BIS     #MRESET,@TXCSR  ;MASTER RESET
007012  012777  020000  172702        MOV     #SYNEXT,@PARCSR ;SET THE MODE
007020  052777  000400  172700        BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINTENANCE MODE & SEND
                                        ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
007026  012777  004020  172672        MOV     #MINT!SEND,@TXCSR

                                ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
007034  012777  022026  172660        MOV     #SYNEXT!SIX!NOPAR!26,@PARCSR
007042  016703  172660                MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
007046  112777  000021  172656        MOVB    #21,@TXDBUF     ;LOAD CHAR
007054  012767  000021  172416        MOV     #21,$TMP1       ;SHIFTED CHAR
007062  012767  000006  172032        MOV     #6,SHIFT        ;# OF SHIFTS

                                ;POKE CLK TO GET INTO SYNCRONIZATION
007070  052777  020000  172630        BIS     #CLK,@TXCSR     ;POKE CLK UP
007076  042777  020000  172622        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
007104  005000                1$:     CLR     R0
007106  006067  172366                ROR     $TMP1   ;FORCE CARRY
007112  103002                        BCC     2$
007114  052700  002000                BIS     #BITW,R0        ;EQUIV OF BIT WINDOW
007120                          2$:
007120  052777  020000  172600        BIS     #CLK,@TXCSR     ;POKE CLK UP
007126  042777  020000  172572        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
007134  017701  172566                MOV     @TXCSR,R1       ;ACTUAL
007140  042701  075777                BIC     #075777,R1      ;SAVE BITW & DNA
007144  020001                        CMP     R0,R1   ;COMPARE EXP VS ACT
007146  001401                        BEQ     .+4
007150  104003                        ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
                                                        ;BIT,....ALSO CHECK DNA
007152  005367  171744                DEC     SHIFT   ;# OF SHIFTS
007156  001352                        BNE     1$      ;DO IT AGAIN ?
                                ;NOW POKE CLK TO SEE DNA
007160  052777  020000  172540        BIS     #CLK,@TXCSR     ;POKE CLK
007166  012700  100000                MOV     #100000,R0      ;EXPECTED
007172  017701  172530                MOV     @TXCSR,R1       ;ACTUAL
007176  042701  077777                BIC     #77777,R1       ;SAVE DNA ONLY
007202  020001                        CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
007204  001401                        BEQ     .+4
007206  104003                        ERROR   3       ;DNA SHOULD BE SET
                                                        ;IF DNA DID NOT SET ,CHECK WORD LENGTH
                                                        ;SELECT LOGIC OF THE TRANSMITTER
007210  005777  172512                TST     @TXCSR  ;DNA ?
007214  100001                        BPL     .+4
007216  104004                        ERROR   4               ;DNA SHOULD NOT BE SET
                                                        ;IT SHOULD HAVE BEEN CLEARED FROM
                                                        ;PREVIOUS READ

                                        ;;THIS TEST VERIFYS WORD LENGTH SELECT OF
                                        ;;THE TRANSMITTER SECTION,IT USES THE DNA FLAG
                                        ;;AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
                                        ;;CORRECTLY
```

```
                                        ;;NOTE:  DNA COMES UP ON THE FIRST RISING BIT
                                        ;;EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
                                        ;;LOADED INTO TXDBUF
                                        ;;MODE: SYNEXT
                                        ;;PARITY: NO PARITY
                                        ;;LENGTH: SEVEN
                                        ;;
                                    ;;**************************************************************************
007220  000004                    TST14:  SCOPE
007222  052777  000400  172476            BIS     #MRESET,@TXCSR  ;MASTER RESET
007230  012777  020000  172464            MOV     #SYNEXT,@PARCSR ;SET THE MODE
007236  052777  000400  172462            BIS     #MRESET,@TXCSR  ;MASTER RESET

                                    ;SET MAINTENANCE MODE & SEND
                                        ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
007244  012777  004020  172454            MOV     #MINT!SEND,@TXCSR

                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
007252  012777  024026  172442            MOV     #SYNEXT!SEVEN!NOPAR!26,@PARCSR
007260  016703  172442                    MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
007264  112777  000021  172440            MOVB    #21,@TXDBUF     ;LOAD CHAR
007272  012767  000021  172200            MOV     #21,$TMP1       ;SHIFTED CHAR
007300  012767  000007  171614            MOV     #7,SHIFT        ;# OF SHIFTS
                                    ;POKE CLK TO GET INTO SYNCRONIZATION
007306  052777  020000  172412            BIS     #CLK,@TXCSR     ;POKE CLK UP
007314  042777  020000  172404            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
007322  005000                    1$:     CLR     R0
007324  006067  172150                    ROR     $TMP1   ;FORCE CARRY
007330  103002                    BCC     2$
007332  052700  002000                    BIS     #BITW,R0        ;EQUIV OF BIT WINDOW
007336                          2$:
007336  052777  020000  172362            BIS     #CLK,@TXCSR     ;POKE CLK UP
007344  042777  020000  172354            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
007352  017701  172350                    MOV     @TXCSR,R1       ;ACTUAL
007356  042701  075777                    BIC     #075777,R1      ;SAVE BITW & DNA
007362  020001                    CMP     R0,R1   ;COMPARE EXP VS ACT
007364  001401                    BEQ     .+4
007366  104003                    ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
                                                    ;BIT.... ALSO CHECK DNA
007370  005367  171526                    DEC     SHIFT   ;# OF SHIFTS
007374  001352                    BNE     1$      ;DO IT AGAIN ?
                                    ;NOW POKE CLK TO SEE DNA
007376  052777  020000  172322            BIS     #CLK,@TXCSR     ;POKE CLK
007404  012700  100000                    MOV     #100000,R0      ;EXPECTED
007410  017701  172312                    MOV     @TXCSR,R1       ;ACTUAL
007414  042701  077777                    BIC     #77777,R1       ;SAVE DNA ONLY
007420  020001                    CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
007422  001401                    BEQ     .+4
007424  104003                    ERROR   3       ;DNA SHOULD BE SET
                                                    ;IF DNA DID NOT SET ,CHECK WORD LENGTH
                                                    ;SELECT LOGIC OF THE TRANSMITTER
007426  005777  172274                    TST     @TXCSR  ;DNA ?
007432  100001                    BPL     .+4
007434  104004                    ERROR   4               ;DNA SHOULD NOT BE SET
                                                    ;IT SHOULD HAVE BEEN CLEARED FROM
                                                    ;PREVIOUS READ
```

```
                                        ;;THIS TEST VERIFYS WORD LENGTH SELECT OF
                                        ;;THE TRANSMITTER SECTION,IT USES THE DNA FLAG
                                        ;;AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
                                        ;;CORRECTLY
                                        ;;NOTE: DNA COMES UP ON THE FIRST RISING BIT
                                        ;;EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
                                        ;;LOADED INTO TXDBUF
                                        ;;MODE: SYNEXT
                                        ;;PARITY: NO PARITY
                                        ;;LENGTH: EIGHT
                                        ;;
                                    ;;*****************************************************************
007436  000004                    TST15:  SCOPE
007440  052777  000400  172260            BIS     #MRESET,@TXCSR  ;MASTER RESET
007446  012777  020000  172246            MOV     #SYNEXT,@PARCSR ;SET THE MODE
007454  052777  000400  172244            BIS     #MRESET,@TXCSR  ;MASTER RESET

                                    ;SET MAINTENANCE MODE & SEND
                                            ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
007462  012777  004020  172236            MOV     #MINT!SEND,@TXCSR

                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
007470  012777  026026  172224            MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
007476  016703  172224                    MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
007502  112777  000021  172222            MOVB    #21,@TXDBUF     ;LOAD CHAR
007510  012767  000021  171762            MOV     #21,$TMP1       ;SHIFTED CHAR
007516  012767  000010  171376            MOV     #8.,SHIFT       ;# OF SHIFTS

                                    ;POKE CLK TO GET INTO SYNCRONIZATION
007524  052777  020000  172174            BIS     #CLK,@TXCSR     ;POKE CLK UP
007532  042777  020000  172166            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
007540  005000                    1$:     CLR     R0
007542  006067  171732                    ROR     $TMP1   ;FORCE CARRY
007546  103002                            BCC     2$
007550  052700  002000                    BIS     #BITW,R0        ;EQUIV OF BIT WINDOW
007554                              2$:
007554  052777  020000  172144            BIS     #CLK,@TXCSR     ;POKE CLK UP
007562  042777  020000  172136            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
007570  017701  172132                    MOV     @TXCSR,R1       ;ACTUAL
007574  042701  075777                    BIC     #075777,R1      ;SAVE BITW & DNA
007600  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
007602  001401                            BEQ     .+4
007604  104003                            ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
                                                        ;BIT.....ALSO CHECK DNA
007606  005367  171310                    DEC     SHIFT   ;# OF SHIFTS
007612  001352                            BNE     1$      ;DO IT AGAIN ?

                                    ;NOW POKE CLK TO SEE DNA
007614  052777  020000  172104            BIS     #CLK,@TXCSR     ;POKE CLK
007622  012700  100000                    MOV     #100000,R0      ;EXPECTED
007626  017701  172074                    MOV     @TXCSR,R1       ;ACTUAL
007632  042701  077777                    BIC     #77777,R1       ;SAVE DNA ONLY
007636  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
007640  001401                            BEQ     .+4
007642  104003                            ERROR   3       ;DNA SHOULD BE SET
                                                        ;IF DNA DID NOT SET ,CHECK WORD LENGTH
                                                        ;SELECT LOGIC OF THE TRANSMITTER
```

```
2241  007644  005777  172056          TST     @TXCSR  ;DNA ?
2242  007650  100001                  BPL     .+4
2243  007652  104004                  ERROR   4               ;DNA SHOULD NOT BE SET
2244                                                  ;IT SHOULD HAVE BEEN CLEARED FROM
2245                                                  ;PREVIOUS READ
2246
2247                                  ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2248                                  ;;OF THE TRANSMITTER SECTION.
2249                                  ;;IT ALSO CHECKS DNA TIMING
2250                                  ;;MODE: SYNINT
2251                                  ;;LENGTH: FIVE PLUS PARITY
2252                                  ;;PARITY: EVEPAR
2253                                  ;;CHARACTER: 25
2254                                  ;;
2255                          ;;**************************************************************************
2256  007654  000004          TST16:  SCOPE
2257  007656  052777  000400  172042  BIS     #MRESET,@TXCSR  ;MASTER RESET
2258  007664  012777  030000  172030  MOV     #SYNINT,@PARCSR ;SET THE MODE
2259  007672  052777  000400  172026  BIS     #MRESET,@TXCSR  ;MASTER RESET
2260
2261                          ;SET MAINTENANCE MODE & SEND
2262                                  ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2263  007700  012777  004020  172020  MOV     #MINT!SEND,@TXCSR
2264
2265                          ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2266  007706  012777  031426  172006  MOV     #SYNINT!FIVE!EVEPAR!26,@PARCSR
2267  007714  016703  172006          MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
2268  007720  112777  000025  172004  MOVB    #25,@TXDBUF     ;LOAD DATA CHAR
2269  007726  012767  000065  171544  MOV     #65,$TMP1       ;TO BE SHIFTED CHAR
2270  007734  012767  000006  171160  MOV     #6,SHIFT        ;# OF SHIFTS
2271                          ;POKE CLK TO GET INTO SYNCRONIZATION
2272  007742  052777  020000  171756  BIS     #CLK,@TXCSR     ;POKE CLK UP
2273  007750  042777  020000  171750  BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2274  007756  005000          1$:     CLR     R0
2275  007760  006067  171514          ROR     $TMP1   ;FORCE CARRY
2276  007764  103002                  BCC     2$      ;BR IF CARRY CLR
2277  007766  052700  002000          BIS     #BITW,R0        ;EQUIV OF BITW
2278  007772                  2$:
2279  007772  052777  020000  171726  BIS     #CLK,@TXCSR     ;POKE CLK UP
2280  010000  042777  020000  171720  BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2281  010006  017701  171714          MOV     @TXCSR,R1       ;ACTUAL
2282  010012  042701  075777          BIC     #075777,R1      ;SAVE BITW & DNA
2283  010016  020001                  CMP     R0,R1   ;COMPARE EXP VS ACT
2284  010020  001401                  BEQ     .+4
2285  010022  104003                  ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2286                                                  ;BIT....ALSO CHECK DNA
2287  010024  005367  171072          DEC     SHIFT   ;# OF SHIFTS
2288  010030  001352                  BNE     1$      ;DO IT AGAIN ?
2289                          ;NOW POKE CLK TO SEE DNA
2290  010032  052777  020000  171666  BIS     #CLK,@TXCSR     ;POKE CLK
2291  010040  012700  100000          MOV     #100000,R0      ;EXPECTED
2292  010044  017701  171656          MOV     @TXCSR,R1       ;ACTUAL
2293  010050  042701  077777          BIC     #77777,R1       ;SAVE DNA ONLY
2294  010054  020001                  CMP     R0,R1           ;COMPARE EXP VS ACT
2295  010056  001401                  BEQ     .+4
2296  010060  104003                  ERROR   3       ;DNA SHOULD BE SET
```

```
                                                        ;IF DNA DID NOT SET
                                                        ;CHECK WORD LENGTH SELECT LOGIC

                                                ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
                                                ;;OF THE TRANSMITTER SECTION.
                                                ;;IT ALSO CHECKS DNA TIMING
                                                ;;MODE: SYNINT
                                                ;;LENGTH: FIVE PLUS PARITY
                                                ;;PARITY: ODDPAR
                                                ;;CHARACTER: 25
                                                ;;
                                        ;;**************************************************************************
010062  000004                          TST17:  SCOPE
010064  052777  000400  171634                  BIS     #MRESET,@TXCSR  ;MASTER RESET
010072  012777  030000  171622                  MOV     #SYNINT,@PARCSR ;SET THE MODE
010100  052777  000400  171620                  BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINTENANCE MODE & SEND
                                                ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
010106  012777  004020  171612                  MOV     #MINT!SEND,@TXCSR

                                        ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
010114  012777  031026  171600                  MOV     #SYNINT!FIVE!ODDPAR!26,@PARCSR
010122  016703  171600                          MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
010126  112777  000025  171576                  MOVB    #25,@TXDBUF     ;LOAD DATA CHAR
010134  012767  000025  171336                  MOV     #25,$TMP1       ;TO BE SHIFTED CHAR
010142  012767  000006  170752                  MOV     #6,SHIFT        ;# OF SHIFTS

                                        ;POKE CLK TO GET INTO SYNCRONIZATION
010150  052777  020000  171550          1$:     BIS     #CLK,@TXCSR     ;POKE CLK UP
010156  042777  020000  171542                  BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010164  005000                                  CLR     R0
010166  006067  171306                          ROR     $TMP1   ;FORCE CARRY
010172  103002                                  BCC     2$      ;BR IF CARRY CLR
010174  052700  002000                          BIS     #BITW,R0        ;EQUIV OF BITW
010200                                  2$:
010200  052777  020000  171520                  BIS     #CLK,@TXCSR     ;POKE CLK UP
010206  042777  020000  171512                  BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010214  017701  171506                          MOV     @TXCSR,R1       ;ACTUAL
010220  042701  075777                          BIC     #075777,R1      ;SAVE BITW & DNA
010224  020001                                  CMP     R0,R1   ;COMPARE EXP VS ACT
010226  001401                                  BEQ     .+4
010230  104003                                  ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
                                                                ;BIT,...ALSO CHECK DNA
010232  005367  170664                          DEC     SHIFT   ;# OF SHIFTS
010236  001352                                  BNE     1$      ;DO IT AGAIN ?

                                        ;NOW POKE CLK TO SEE DNA
010240  052777  020000  171460                  BIS     #CLK,@TXCSR     ;POKE CLK
010246  012700  100000                          MOV     #100000,R0      ;EXPECTED
010252  017701  171450                          MOV     @TXCSR,R1       ;ACTUAL
010256  042701  077777                          BIC     #77777,R1       ;SAVE DNA ONLY
010262  020001                                  CMP     R0,R1           ;COMPARE EXP VS ACT
010264  001401                                  BEQ     .+4
010266  104003                                  ERROR   3       ;DNA SHOULD BE SET
                                                                ;IF DNA DID NOT SET
                                                                ;CHECK WORD LENGTH SELECT LOGIC
```

```
                                        ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
                                        ;;OF THE TRANSMITTER SECTION.
                                        ;;IT ALSO CHECKS DNA TIMING
                                        ;;MODE: ISYMOD
                                        ;;LENGTH: FIVE PLUS PARITY
                                        ;;PARITY: EVEPAR
                                        ;;CHARACTER: 25
                                        ;;
                                    ;;**********************************************************************
010270  000004                    TST20:  SCOPE
010272  052777  000400  171426            BIS     #MRESET,@TXCSR  ;MASTER RESET
010300  012777  000000  171414            MOV     #ISYMOD,@PARCSR ;SET THE MODE
010306  052777  000400  171412            BIS     #MRESET,@TXCSR  ;MASTER RESET

                                    ;SET MAINTENANCE MODE & SEND
                                            ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
010314  012777  004020  171404            MOV     #MINT!SEND,@TXCSR

                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
010322  012777  001426  171372            MOV     #ISYMOD!FIVE!EVEPAR!26,@PARCSR
010330  016703  171372                    MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
010334  112777  000025  171370            MOVB    #25,@TXDBUF     ;LOAD DATA CHAR
010342  012767  000352  171130            MOV     #352,$TMP1      ;TO BE SHIFTED CHAR.
010350  012767  000010  170544            MOV     #8.,SHIFT       ;# OF SHIFTS

                                    ;POKE CLK TO GET INTO SYNCRONIZATION
010356  052777  020000  171342            BIS     #CLK,@TXCSR     ;POKE CLK UP
010364  042777  020000  171334            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010372  005000                    1$:     CLR     R0
010374  006067  171100                    ROR     $TMP1   ;FORCE CARRY
010400  103002                            BCC     2$      ;BR IF CARRY CLR
010402  052700  002000                    BIS     #BITW,R0        ;EQUIV OF BITW
010406                            2$:
010406  052777  020000  171312            BIS     #CLK,@TXCSR     ;POKE CLK UP
010414  042777  020000  171304            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010422  017701  171300                    MOV     @TXCSR,R1       ;ACTUAL
010426  042701  075777                    BIC     #075777,R1      ;SAVE BITW & DNA
010432  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
010434  001401                            BEQ     .+4
010436  104003                            ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
                                                        ;BIT,...ALSO CHECK DNA
010440  005367  170456                    DEC     SHIFT   ;# OF SHIFTS
010444  001352                            BNE     1$      ;DO IT AGAIN ?
                                    ;NOW POKE CLK TO SEE DNA
010446  052777  020000  171252            BIS     #CLK,@TXCSR     ;POKE CLK
010454  012700  000000                    MOV     #0,R0   ;EXPECTED
010460  017701  171242                    MOV     @TXCSR,R1       ;ACTUAL
010464  042701  077777                    BIC     #77777,R1       ;SAVE DNA ONLY
010470  020001                            CMP     R0,R1           ;COMPARE EXP VS ACT
010472  001401                            BEQ     .+4
010474  104003                            ERROR   3       ;DNA SHOULD BE SET
                                                        ;IF DNA DID NOT SET
                                                        ;CHECK WORD LENGTH SELECT LOGIC

                                        ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
                                        ;;OF THE TRANSMITTER SECTION.
                                        ;;IT ALSO CHECKS DNA TIMING
```

```
                                        ;;MODE: ISYMOD
                                        ;;LENGTH: FIVE PLUS PARITY
                                        ;;PARITY: ODDPAR
                                        ;;CHARACTER: 25
                                        ;;
                                ;;*****************************************************************
010476  000004                TST21:  SCOPE
010500  052777  000400  171220        BIS     #MRESET,@TXCSR  ;MASTER RESET
010506  012777  000000  171206        MOV     #ISYMOD,@PARCSR ;SET THE MODE
010514  052777  000400  171204        BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINTENANCE MODE & SEND
                                        ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
010522  012777  004020  171176        MOV     #MINT!SEND,@TXCSR

                                ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
010530  012777  001026  171164        MOV     #ISYMOD!FIVE!ODDPAR!26,@PARCSR
010536  016703  171164                MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
010542  112777  000025  171162        MOVB    #25,@TXDBUF     ;LOAD DATA CHAR
010550  012767  000252  170722        MOV     #252,$TMP1      ;TO BE SHIFTED CHAR
010556  012767  000010  170336        MOV     #8.,SHIFT       ;# OF SHIFTS

                                ;POKE CLK TO GET INTO SYNCRONIZATION
010564  052777  020000  171134        BIS     #CLK,@TXCSR     ;POKE CLK UP
010572  042777  020000  171126        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010600  005000                1$:     CLR     R0
010602  006067  170672                ROR     $TMP1   ;FORCE CARRY
010606  103002                        BCC     2$      ;BR IF CARRY CLR
010610  052700  002000                BIS     #BITW,R0        ;EQUIV OF BITW
010614                          2$:
010614  052777  020000  171104        BIS     #CLK,@TXCSR     ;POKE CLK UP
010622  042777  020000  171076        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010630  017701  171072                MOV     @TXCSR,R1       ;ACTUAL
010634  042701  075777                BIC     #075777,R1      ;SAVE BITW & DNA
010640  020001                        CMP     R0,R1   ;COMPARE EXP VS ACT
010642  001401                        BEQ     .+4
010644  104003                        ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
                                                        ;BIT,...ALSO CHECK DNA
010646  005367  170250                DEC     SHIFT   ;# OF SHIFTS
010652  001352                        BNE     1$      ;DO IT AGAIN ?
                                ;NOW POKE CLK TO SEE DNA
010654  052777  020000  171044        BIS     #CLK,@TXCSR     ;POKE CLK
010662  012700  000000                MOV     #0,R0   ;EXPECTED
010666  017701  171034                MOV     @TXCSR,R1       ;ACTUAL
010672  042701  077777                BIC     #77777,R1       ;SAVE DNA ONLY
010676  020001                        CMP     R0,R1           ;COMPARE EXP VS ACT
010700  001401                        BEQ     .+4
010702  104003                        ERROR   3       ;DNA SHOULD BE SET
                                                        ;IF DNA DID NOT SET
                                                        ;CHECK WORD LENGTH SELECT LOGIC


                                        ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
                                        ;;OF THE TRANSMITTER SECTION.
                                        ;;IT ALSO CHECKS DNA TIMING
                                        ;;MODE: SYNINT
                                        ;;LENGTH: SIX PLUS PARITY
```

```
                                        ;;PARITY:EVEPAR
                                        ;;CHARACTER:25
                                        ;;
                                    ;;**********************************************************************
2469  010704  000004                TST22:  SCOPE
2470  010706  052777  000400  171012        BIS     #MRESET,@TXCSR  ;MASTER RESET
2471  010714  012777  030000  171000        MOV     #SYNINT,@PARCSR ;SET THE MODE
2472  010722  052777  000400  170776        BIS     #MRESET,@TXCSR  ;MASTER RESET
2473
2474                                    ;SET MAINTENANCE MODE & SEND
2475                                        ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2476  010730  012777  004020  170770        MOV     #MINT!SEND,@TXCSR
2477
2478                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2479  010736  012777  033426  170756        MOV     #SYNINT!SIX!EVEPAR!26,@PARCSR
2480  010744  016703  170756              MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
2481  010750  112777  000025  170754        MOVB    #25,@TXDBUF     ;LOAD DATA CHAR
2482  010756  012767  000125  170514        MOV     #125,$TMP1      ;TO BE SHIFTED CHAR
2483  010764  012767  000007  170130        MOV     #7,SHIFT        ;# OF SHIFTS
2484                                    ;POKE CLK TO GET INTO SYNCRONIZATION
2485  010772  052777  020000  170726        BIS     #CLK,@TXCSR     ;POKE CLK UP
2486  011000  042777  020000  170720        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2487  011006  005000                1$:     CLR     R0
2488  011010  006067  170464              ROR     $TMP1   ;FORCE CARRY
2489  011014  103002                       BCC     2$      ;BR IF CARRY CLR
2490  011016  052700  002000              BIS     #BITW,R0        ;EQUIV OF BITW
2491  011022                        2$:
2492  011022  052777  020000  170676        BIS     #CLK,@TXCSR     ;POKE CLK UP
2493  011030  042777  020000  170670        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2494  011036  017701  170664              MOV     @TXCSR,R1       ;ACTUAL
2495  011042  042701  075777              BIC     #075777,R1      ;SAVE BITW & DNA
2496  011046  020001                       CMP     R0,R1   ;COMPARE EXP VS ACT
2497  011050  001401                       BEQ     .+4
2498  011052  104003                       ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2499                                                        ;BIT,...ALSO CHECK DNA
2500  011054  005367  170042              DEC     SHIFT   ;# OF SHIFTS
2501  011060  001352                       BNE     1$      ;DO IT AGAIN ?
2502                                    ;NOW POKE CLK TO SEE DNA
2503  011062  052777  020000  170636        BIS     #CLK,@TXCSR     ;POKE CLK
2504  011070  012700  100000              MOV     #100000,R0      ;EXPECTED
2505  011074  017701  170626              MOV     @TXCSR,R1       ;ACTUAL
2506  011100  042701  077777              BIC     #77777,R1       ;SAVE DNA ONLY
2507  011104  020001                       CMP     R0,R1           ;COMPARE EXP VS ACT
2508  011106  001401                       BEQ     .+4
2509  011110  104003                       ERROR   3       ;DNA SHOULD BE SET
                                                        ;IF DNA DID NOT SET
                                                        ;CHECK WORD LENGTH SELECT LOGIC
2510
2511
2512
2513                                    ;;THIS TEST VERIFYS CHARACTER PLL'S PARITY GENERATION
2514                                    ;;OF THE TRANSMITTER SECTION.
2515                                    ;;IT ALSO CHECKS DNA TIMING
2516                                    ;;MODE:SYNINT
2517                                    ;;LENGTH:SIX PLUS PARITY
2518                                    ;;PARITY:ODDPAR
2519                                    ;;CHARACTER:25
2520                                    ;;
```

```
2521                                    ;;**************************************************************
2522  011112  000004                    TST23:  SCOPE
2523  011114  052777  000400  170604            BIS     #MRESET,@TXCSR  ;MASTER RESET
2524  011122  012777  030000  170572            MOV     #SYNINT,@PARCSR ;SET THE MODE
2525  011130  052777  000400  170570            BIS     #MRESET,@TXCSR  ;MASTER RESET
2526
2527                                    ;SET MAINTENANCE MODE & SEND
2528                                            ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2529  011136  012777  004020  170562            MOV     #MINT!SEND,@TXCSR
2530
2531                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2532  011144  012777  033026  170550            MOV     #SYNINT!SIX!ODDPAR!26,@PARCSR
2533  011152  016703  170550                    MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
2534  011156  112777  000025  170546            MOVB    #25,@TXDBUF     ;LOAD DATA CHAR
2535  011164  012767  000025  170306            MOV     #25,$TMP1       ;TO BE SHIFTED CHAR
2536  011172  012767  000007  167722            MOV     #7,SHIFT        ;# OF SHIFTS
2537                                    ;POKE CLK TO GET INTO SYNCRONIZATION
2538  011200  052777  020000  170520            BIS     #CLK,@TXCSR     ;POKE CLK UP
2539  011206  042777  020000  170512            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2540  011214  005000                    1$:     CLR     R0
2541  011216  006067  170256                    ROR     $TMP1   ;FORCE CARRY
2542  011222  103002                            BCC     2$      ;BR IF CARRY CLR
2543  011224  052700  002000                    BIS     #BITW,R0        ;EQUIV OF BITW
2544  011230                            2$:
2545  011230  052777  020000  170470            BIS     #CLK,@TXCSR     ;POKE CLK UP
2546  011236  042777  020000  170462            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2547  011244  017701  170456                    MOV     @TXCSR,R1       ;ACTUAL
2548  011250  042701  075777                    BIC     #075777,R1      ;SAVE BITW & DNA
2549  011254  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
2550  011256  001401                            BEQ     .+4
2551  011260  104003                            ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2552                                                            ;BIT,...ALSO CHECK DNA
2553  011262  005367  167634                    DEC     SHIFT   ;# OF SHIFTS
2554  011266  001352                            BNE     1$      ;DO IT AGAIN ?
2555                                    ;NOW POKE CLK TO SEE DNA
2556  011270  052777  020000  170430            BIS     #CLK,@TXCSR     ;POKE CLK
2557  011276  012700  100000                    MOV     #100000,R0      ;EXPECTED
2558  011302  017701  170420                    MOV     @TXCSR,R1       ;ACTUAL
2559  011306  042701  077777                    BIC     #77777,R1       ;SAVE DNA ONLY
2560  011312  020001                            CMP     R0,R1           ;COMPARE EXP VS ACT
2561  011314  001401                            BEQ     .+4
2562  011316  104003                            ERROR   3       ;DNA SHOULD BE SET
2563                                                            ;IF DNA DID NOT SET
2564                                                            ;CHECK WORD LENGTH SELECT LOGIC
2565
2566                                            ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2567                                            ;;OF THE TRANSMITTER SECTION.
2568                                            ;;IT ALSO CHECKS DNA TIMING
2569                                            ;;MODE: ISYMOD
2570                                            ;;LENGTH: SIX PLUS PARITY
2571                                            ;;PARITY: EVEPAR
2572                                            ;;CHARACTER: 25
2573                                            ;;
2574                                    ;;**************************************************************
2575  011320  000004                    TST24:  SCOPE
2576  011322  052777  000400  170376            BIS     #MRESET,@TXCSR  ;MASTER RESET
```

```
2577  011330  012777  000000  170364          MOV     #ISYMOD,@PARCSR ;SET THE MODE
2578  011336  052777  000400  170362          BIS     #MRESET,@TXCSR  ;MASTER RESET
2579
2580                                  ;SET MAINTENANCE MODE & SEND
2581                                          ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2582  011344  012777  004020  170354          MOV     #MINT!SEND,@TXCSR
2583
2584                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2585  011352  012777  003426  170342          MOV     #ISYMOD!SIX!EVEPAR!26,@PARCSR
2586  011360  016703  170342                  MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
2587  011364  112777  000025  170340          MOVB    #25,@TXDBUF     ;LOAD DATA CHAR
2588  011372  012767  000652  170100          MOV     #652,$TMP1      ;TO BE SHIFTED CHAR
2589  011400  012767  000011  167514          MOV     #9.,SHIFT       ;# OF SHIFTS
2590                                  ;POKE CLK TO GET INTO SYNCRONIZATION
2591  011406  052777  020000  170312          BIS     #CLK,@TXCSR     ;POKE CLK UP
2592  011414  042777  020000  170304          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2593  011422  005000                  1$:     CLR     R0
2594  011424  006067  170050                  ROR     $TMP1   ;FORCE CARRY
2595  011430  103002                          BCC     2$      ;BR IF CARRY CLR
2596  011432  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BITW
2597  011436                          2$:
2598  011436  052777  020000  170262          BIS     #CLK,@TXCSR     ;POKE CLK UP
2599  011444  042777  020000  170254          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2600  011452  017701  170250                  MOV     @TXCSR,R1       ;ACTUAL
2601  011456  042701  075777                  BIC     #075777,R1      ;SAVE BITW & DNA
2602  011462  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
2603  011464  001401                          BEQ     .+4
2604  011466  104003                          ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2605                                                          ;BIT,...ALSO CHECK DNA
2606  011470  005367  167426                  DEC     SHIFT   ;# OF SHIFTS
2607  011474  001352                          BNE     1$      ;DO IT AGAIN ?
2608                                  ;NOW POKE CLK TO SEE DNA
2609  011476  052777  020000  170222          BIS     #CLK,@TXCSR     ;POKE CLK
2610  011504  012700  000000                  MOV     #0,R0   ;EXPECTED
2611  011510  017701  170212                  MOV     @TXCSR,R1       ;ACTUAL
2612  011514  042701  077777                  BIC     #77777,R1       ;SAVE DNA ONLY
2613  011520  020001                          CMP     R0,R1           ;COMPARE EXP VS ACT
2614  011522  001401                          BEQ     .+4
2615  011524  104003                          ERROR   3       ;DNA SHOULD BE SET
2616                                                          ;IF DNA DID NOT SET
2617                                                          ;CHECK WORD LENGTH SELECT LOGIC
2618
2619                                          ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2620                                          ;;OF THE TRANSMITTER SECTION.
2621                                          ;;IT ALSO CHECKS DNA TIMING
2622                                          ;;MODE: ISYMOD
2623                                          ;;LENGTH: SIX PLUS PARITY
2624                                          ;;PARITY: ODDPAR
2625                                          ;;CHARACTER: 25
2626                                          ;;
2627                                  ;;**************************************************************************
2628  011526  000004                  TST25:  SCOPE
2629  011530  052777  000400  170170          BIS     #MRESET,@TXCSR  ;MASTER RESET
2630  011536  012777  000000  170156          MOV     #ISYMOD,@PARCSR ;SET THE MODE
2631  011544  052777  000400  170154          BIS     #MRESET,@TXCSR  ;MASTER RESET
2632
```

```
2633                                    ;SET MAINTENANCE MODE & SEND
2634                                            ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2635  011552  012777  004020  170146            MOV     #MINT!SEND,@TXCSR
2636
2637                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2638  011560  012777  003026  170134            MOV     #ISYMOD!SIX!ODDPAR!26,@PARCSR
2639  011566  016703  170134                    MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
2640  011572  112777  000025  170132            MOVB    #25,@TXDBUF     ;LOAD DATA CHAR
2641  011600  012767  000452  167672            MOV     #452,$TMP1      ;TO BE SHIFTED CHAR
2642  011606  012767  000011  167306            MOV     #9.,SHIFT       ;# OF SHIFTS
2643                                    ;POKE CLK TO GET INTO SYNCRONIZATION
2644  011614  052777  020000  170104            BIS     #CLK,@TXCSR     ;POKE CLK UP
2645  011622  042777  020000  170076            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2646  011630  005000                    1$:     CLR     R0
2647  011632  006067  167642                    ROR     $TMP1   ;FORCE CARRY
2648  011636  103002                            BCC     2$      ;BR IF CARRY CLR
2649  011640  052700  002000                    BIS     #BITW,R0        ;EQUIV OF BITW
2650  011644                            2$:
2651  011644  052777  020000  170054            BIS     #CLK,@TXCSR     ;POKE CLK UP
2652  011652  042777  020000  170046            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2653  011660  017701  170042                    MOV     @TXCSR,R1       ;ACTUAL
2654  011664  042701  075777                    BIC     #075777,R1      ;SAVE BITW & DNA
2655  011670  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
2656  011672  001401                            BEQ      +4
2657  011674  104003                            ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2658                                                            ;BIT,...ALSO CHECK DNA
2659  011676  005367  167220                    DEC     SHIFT   ;# OF SHIFTS
2660  011702  001352                            BNE     1$      ;DO IT AGAIN ?
2661                                    ;NOW POKE CLK TO SEE DNA
2662  011704  052777  020000  170014            BIS     #CLK,@TXCSR     ;POKE CLK
2663  011712  012700  000000                    MOV     #0,R0   ;EXPECTED
2664  011716  017701  170004                    MOV     @TXCSR,R1       ;ACTUAL
2665  011722  042701  077777                    BIC     #77777,R1       ;SAVE DNA ONLY
2666  011726  020001                            CMP     R0,R1           ;COMPARE EXP VS ACT
2667  011730  001401                            BEQ     .+4
2668  011732  104003                            ERROR   3       ;DNA SHOULD BE SET
2669                                                            ;IF DNA DID NOT SET
2670                                                            ;CHECK WORD LENGTH SELECT LOGIC
2671
2672
2673
2674                                            ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2675                                            ;;OF THE TRANSMITTER SECTION.
2676                                            ;;IT ALSO CHECKS DNA TIMING
2677                                            ;;MODE: SYNINT
2678                                            ;;LENGTH: SEVEN PLUS PARITY
2679                                            ;;PARITY: EVEPAR
2680                                            ;;CHARACTER: 125
2681                                            ;;
2682                                    ;;**********************************************************************
2683  011734  000004                    TST26:  SCOPE
2684  011736  052777  000400  167762            BIS     #MRESET,@TXCSR  ;MASTER RESET
2685  011744  012777  030000  167750            MOV     #SYNINT,@PARCSR ;SET THE MODE
2686  011752  052777  000400  167746            BIS     #MRESET,@TXCSR  ;MASTER RESET
2687
2688                                    ;SET MAINTENANCE MODE & SEND
```

```
2689                                            ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2690  011760  012777  004020  167740            MOV     #MINT!SEND,@TXCSR
2691
2692                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2693  011766  012777  035426  167726            MOV     #SYNINT!SEVEN!EVEPAR!26,@PARCSR
2694  011774  016703  167726                    MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
2695  012000  112777  000125  167724            MOVB    #125,@TXDBUF    ;LOAD DATA CHAR
2696  012006  012767  000125  167464            MOV     #125,$TMP1      ;TO BE SHIFTED CHAR
2697  012014  012767  000010  167100            MOV     #8.,SHIFT       ;# OF SHIFTS
2698                                    ;POKE CLK TO GET INTO SYNCRONIZATION
2699  012022  052777  020000  167676            BIS     #CLK,@TXCSR     ;POKE CLK UP
2700  012030  042777  020000  167670            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2701  012036  005000                    1$:     CLR     R0
2702  012040  006067  167434                    ROR     $TMP1   ;FORCE CARRY
2703  012044  103002                            BCC     2$      ;BR IF CARRY CLR
2704  012046  052700  002000                    BIS     #BITW,R0        ;EQUIV OF BITW
2705  012052                            2$:
2706  012052  052777  020000  167646            BIS     #CLK,@TXCSR     ;POKE CLK UP
2707  012060  042777  020000  167640            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2708  012066  017701  167634                    MOV     @TXCSR,R1       ;ACTUAL
2709  012072  042701  075777                    BIC     #075777,R1      ;SAVE BITW & DNA
2710  012076  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
2711  012100  001401                            BEQ     .+4
2712  012102  104003                            ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2713                                                            ;BIT,...ALSO CHECK DNA
2714  012104  005367  167012                    DEC     SHIFT   ;# OF SHIFTS
2715  012110  001352                            BNE     1$      ;DO IT AGAIN ?
2716                                    ;NOW POKE CLK TO SEE DNA
2717  012112  052777  020000  167606            BIS     #CLK,@TXCSR     ;POKE CLK
2718  012120  012700  100000                    MOV     #100000,R0      ;EXPECTED
2719  012124  017701  167576                    MOV     @TXCSR,R1       ;ACTUAL
2720  012130  042701  077777                    BIC     #77777,R1       ;SAVE DNA ONLY
2721  012134  020001                            CMP     R0,R1           ;COMPARE EXP VS ACT
2722  012136  001401                            BEQ     .+4
2723  012140  104003                            ERROR   3       ;DNA SHOULD BE SET
2724                                                            ;IF DNA DID NOT SET
2725                                                            ;CHECK WORD LENGTH SELECT LOGIC
2726
2727                                            ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2728                                            ;;OF THE TRANSMITTER SECTION.
2729                                            ;;IT ALSO CHECKS DNA TIMING
2730                                            ;;MODE: SYNINT
2731                                            ;;LENGTH: SEVEN PLUS PARITY
2732                                            ;;PARITY: ODDPAR
2733                                            ;;CHARACTER: 125
2734                                            ;;
2735                                    ;;**************************************************************
2736  012142  000004                    TST27:  SCOPE
2737  012144  052777  000400  167554            BIS     #MRESET,@TXCSR  ;MASTER RESET
2738  012152  012777  030000  167542            MOV     #SYNINT,@PARCSR ;SET THE MODE
2739  012160  052777  000400  167540            BIS     #MRESET,@TXCSR  ;MASTER RESET
2740
2741                                    ;SET MAINTENANCE MODE & SEND
2742                                            ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2743  012166  012777  004020  167532            MOV     #MINT!SEND,@TXCSR
2744
```

```
                                ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
012174  012777  035026  167520          MOV     #SYNINT!SEVEN!ODDPAR!26,@PARCSR
012202  016703  167520                  MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
012206  112777  000125  167516          MOVB    #125,@TXDBUF    ;LOAD DATA CHAR
012214  012767  000325  167256          MOV     #325,$TMP1      ;TO BE SHIFTED CHAR
012222  012767  000010  166672          MOV     #8.,SHIFT       ;# OF SHIFTS
                                ;POKE CLK TO GET INTO SYNCRONIZATION
012230  052777  020000  167470          BIS     #CLK,@TXCSR     ;POKE CLK UP
012236  042777  020000  167462          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
012244  005000                  1$:     CLR     R0
012246  006067  167226                  ROR     $TMP1   ;FORCE CARRY
012252  103002                          BCC     2$      ;BR IF CARRY CLR
012254  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BITW
012260                          2$:
012260  052777  020000  167440          BIS     #CLK,@TXCSR     ;POKE CLK UP
012266  042777  020000  167432          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
012274  017701  167426                  MOV     @TXCSR,R1       ;ACTUAL
012300  042701  075777                  BIC     #075777,R1      ;SAVE BITW & DNA
012304  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
012306  001401                          BEQ     .+4
012310  104003                          ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
                                                        ;BIT,...ALSO CHECK DNA
012312  005367  166604                  DEC     SHIFT   ;# OF SHIFTS
012316  001352                          BNE     1$      ;DO IT AGAIN ?
                                ;NOW POKE CLK TO SEE DNA
012320  052777  020000  167400          BIS     #CLK,@TXCSR     ;POKE CLK
012326  012700  100000                  MOV     #100000,R0      ;EXPECTED
012332  017701  167370                  MOV     @TXCSR,R1       ;ACTUAL
012336  042701  077777                  BIC     #77777,R1       ;SAVE DNA ONLY
012342  020001                          CMP     R0,R1           ;COMPARE EXP VS ACT
012344  001401                          BEQ     .+4
012346  104003                          ERROR   3       ;DNA SHOULD BE SET
                                                        ;IF DNA DID NOT SET
                                                        ;CHECK WORD LENGTH SELECT LOGIC
```

```
                                ;END OF PASS
                                ;TYPE NAME OF TEST
                                ;UPDATE PASS COUNT
                                ;CHECK FOR EXIT TO ACT-11
                                ;RESTART TEST

012350  000004          .EOP:   SCOPE
012352  004767  000340          JSR     PC,CKSWR
012356  104401                  TYPE                            ;TYPE NAME OF TEST
012360  015506                  MEPASS
012362  104413  012614          CONVRT  ,OUTCRY
012366  104401  015325          TYPE    ,DEVICE
012372  105767  166554          TSTB    MULTD   ;ARE YOU RUNNING MULTIPLE DEVICES ?
012376  001511                  BEQ     CCC     ;NO, JUMP AROUND
012400  005767  166562          TST     ACTREG  ;ARE ANY DEVICES ACTIVE ?
012404  001007                  BNE     RUNIT   ;YES
012406  104401  015337          TYPE    ,MCOW   ;NO
012412  016700  166550          MOV     ACTREG,R0       ;DISPLAY ACTREG
012416  000000                  HALT            ;SELECT SOMETHING TO RUN @ ACTREG.
```

```
2801                                                ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2802  012420  000167  167526                        JMP     .START  ;START OVER AGAIN.....YOU DESELECTED EVERYTHING
2803  012424  062767  000010  166522  RUNIT:  ADD     #10,BASEADD     ;NEXT BLOCK (ADDRESSES)
2804  012432  062767  000010  166522  ZERO:   ADD     #10,BASEIV      ;NEXT BLOCK (VECTORS)
2805  012440  000241                          CLC
2806  012442  006167  166522                  ROL     ROTADD  ;UP DATE ROTATING POINTER
2807  012446  103410                          BCS     2$      ;IS IT THE LAST DEVICE
2808                                                        ;TO BE TESTED IN THIS PASS ?
2809  012450  036767  166514  166510          BIT     ROTADD,ACTREG   ;TEST THIS DEVICE FOR ACTIVE STATUS
2810  012456  001762                          BEQ     RUNIT   ;IF NOT ACTIVE, TRY NEXT ADDRESS
2811  012460  004767  000034                  JSR     PC,REPLAY       ;CALCULATE NEW PARAMETERS
2812  012464  000167  000210                  JMP     RESTRT  ;YES IT WAS ACTIVE,TEST THIS DEVICE
2813  012470  012767  000001  166472  2$:     MOV     #1,ROTADD       ;OK!,NOW SET UP  ROTATING
2814                                                        ;POINTER FOR NEXT MULTIPLE PASS
2815  012476  016767  166454  166450          MOV     KEEPADD,BASEADD ;RESTORE BASE ADDRESS
2816  012504  016767  166454  166450          MOV     KEEPIV,BASEIV   ;RESTORE BASE INTERRUPT VECTORS
2817  012512  004767  000002                  JSR     PC,REPLAY       ;CALC NEW PARAMETERS
2818  012516  000441                          BR      CCC     ;JUMP AROUND REPLAY
2819  012520  016767  166430  004416  REPLAY: MOV     BASEADD,DUBASE  ;SET UP FOR NEW ADDRESSES
2820  012526  004767  004260                  JSR     PC,DUADDR       ;CREATE NEW ADDRESSES
2821  012532  016767  166424  167176          MOV     BASEIV,DURIV    ;CREATE DURIV
2822  012540  062767  000002  166414          ADD     #2,BASEIV
2823  012546  016767  166410  167164          MOV     BASEIV,DURIS    ;CREATE DURIS
2824  012554  062767  000002  166400          ADD     #2,BASEIV
2825  012562  016767  166374  167152          MOV     BASEIV,DUTIV    ;CREATE DUTIV
2826  012570  062767  000002  166364          ADD     #2,BASEIV
2827  012576  016767  166360  167140          MOV     BASEIV,DUTIS    ;CREATE DUTIS
2828  012604  016767  167126  166350          MOV     DURIV,BASEIV    ;RESTORE
2829  012612  000207                          RTS     PC
2830
2831  012614  000001                  OUTCRY: 1
2832  012616     006     002                  .BYTE   6,2
2833  012620  001712                          RXCSR
2834
2835  012622                          CCC:
2836  012622  005067  166554                  CLR     $TSTNM                  ;CLEAR TEST NUMBER
2837  012626  005067  166564                  CLR     $ERRPC                  ;CLEAR LAST ERROR PC
2838  012632  005067  166545                  CLR     $ERFLG                  ;CLEAR ERROR FLAG
2839  012636  005267  166250                  INC     PASCNT                  ;UPDATE PASS COUNT
2840  012642  016767  166244  166232          MOV     PASCNT,LIGHTS           ;DISPLAY PASS COUNT
2841  012650  016767  166236  166656          MOV     PASCNT,$PASS            ;PASS COUNT TO APT
2842  012656  013701  000042                  MOV     @#42,R1                 ;CHECK FOR ACT-11 OR DDP
2843  012662  001406                          BEQ     RESTRT          ;IF NO CONTINUE TESTING
2844  012664  000005                          RESET
2845  012666  000005                          RESET
2846  012670  004711                  $ENDAD: JSR     PC,(R1)
2847  012672  000240                          NOP
2848  012674  000240                          NOP
2849  012676  000240                          NOP
2850  012700                          RESTRT:
2851  012700  012767  003376  166500          MOV     #TST1+2,$LPADR  ;LOAD LAST ADDR
2852  012706  004767  000004                  JSR     PC,CKSWR
2853  012712  000167  170372                  JMP     .BEGIN
2854
2855                                          ;CHECK SWITCH REGISTER ROUTINE.
2856                                          ;CHECKS TO ALLOW FOR < G> TO ALLOW
```

```
                                                ;THE CHANGING OF LOCATION 176

013716  005737  000042            CKSWR:  TST     @#42
013722  001040                            BNE     OUT
013724  022767  000176  166506            CMP     #SWREG,SWR      ;SOFTWARE SWR PRESENT?
013732  001034                            BNE     OUT             ;NO--LEAVE
013734  105777  166504                    TSTB    @$TKS           ;CHECK TTY READY
013740  100031                            BPL     OUT             ;NO--LEAVE
013742  017767  166500  000422            MOV     @$TKB,.MSG      ;GET CHARACTER
013750  042767  177600  000414            BIC     #177600,.MSG    ;STRIP JUNK
013756  122767  000007  000406            CMPB    #7,.MSG         ;IS IT < G> ?
013764  001017                            BNE     OUT             ;NO
013766  104401  016113                    TYPE    ,MCNTG
013772  005137  013032            CNTLU:  COM     @#RDSW
013776  104401  016123                    TYPE    ,MMSWR
014002  104413                            CONVRT
014004  013034                            SWREGL
014006  104406  016134                    INSTR,MMNEW
014012  104410                            PARAM
014014  000000                            0
014016  177777                            177777
014020  000176                            SWREG
014022     000     001            .BYTE   0,1
014024  005037  013032            OUT:    CLR     @#RDSW
014030  000207                            RTS     PC
014032  000000                    RDSW:   .WORD   0
014034  000001                    SWREGL: 1
014036     006     002                    .BYTE   6,2
014040  000176                            SWREG

014042  000005                            5

                                                ;CHECK FOR FREEZE ON CURRENT DATA

014044  004767  177646            .SCOP1: JSR     PC,CKSWR
014050  032777  001000  166362            BIT     #SW09,@SWR
014056  001402                            BEQ     1$
014060  016716  166024                    MOV     LOCK,(SP)
014064  000002                    1$:     RTI
                                .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
                                ;*
```

```
2913  013066  105767  166365          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
2914  013072  100002                          BPL     1$              ;;BR IF YES
2915  013074  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
2916  013076  000430                          BR      3$              ;;LEAVE
2917  013100  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
2918  013102  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
2919  013106  122767  000001  166432          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
2920  013114  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
2921  013116  132767  000100  166423          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
2922  013124  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
2923  013126  010067  000004                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
2924  013132  004767  000006'                 JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
2925  013136  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
2926  013140  132767  000040  166401  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
2927  013146  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
2928  013150  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
2929  013152  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
2930  013154  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
2931  013156  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
2932  013160  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
2933  013164  000002                          RTI                     ;;RETURN
2934  013166  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
2935  013172  001430                          BEQ     8$
2936  013174  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
2937  013200  001006                          BNE     5$
2938  013202  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
2939  013204  104401                          TYPE                    ;;TYPE A CR AND LF
2940  013206  001523                          $CRLF
2941  013210  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
2942  013214  000755                          BR      2$              ;;GET NEXT CHARACTER
2943  013216  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
2944  013222  126726  166230          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
2945  013226  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
2946  013230  016746  166220                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
2947                                                                  ;;AND THE NULL CHAR.
2948  013234  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
2949  013240  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
2950  013242  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
2951  013246  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
2952  013252  000770                          BR      7$              ;;LOOP
2953
2954                                  ;HORIZONTAL TAB PROCESSOR
2955
2956  013254  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
2957  013260  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
2958  013264  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
2959  013272  001372                          BNE     9$              ;;TAB STOP
2960  013274  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
2961  013276  000724                          BR      2$              ;;GET NEXT CHARACTER
2962  013300  105777  166144          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
2963  013304  100375                          BPL     $TYPEC
2964  013306  116677  000002  166136          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
2965  013314  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
2966  013322  001003                          BNE     1$              ;;BRANCH IF NO
2967  013324  105067  000014                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
2968  013330  000406                          BR      $TYPEX          ;;EXIT
```

```
2969  013332  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
2970  013340  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
2971  013342  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
2972  013344  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STOPAGE
2973  013346  000207                  $TYPEX: RTS     PC
2974
2975
2976                                          ;ASCII STRING INPUT ROUTINE
2977
2978  013350  017667  000000  000014  .INSTR: MOV     @(SP)+,.MSG     ;PICK UP MESSAGE
2979  013356  062716  000002                  ADD     #2,(SP)         ;JUMP AROUND MESSAGE FOR RTI
2980  013362  105767  166160                  TSTB    $ENV            ;APT CONTROL
2981  013366  001036                          BNE     INSTR2          ;YES NO TYPE
2982  013370  104401                  .INST1: TYPE
2983  013372  000000                  .MSG:   0
2984  013374  012704  016146                  MOV     #INBUF,R4       ;GET STARTING LOC OF INBUF
2985  013400  012703  000007                  MOV     #7,R3           ;MAX # OF CHARS
2986  013404  105777  166034          1$:     TSTB    @$TKS   ;TTY FLAG
2987  013410  100375                          BPL     1$
2988  013412  117714  166030                  MOVB    @$TKB,(R4)      ;TAKE CHAR
2989  013416  142714  000200                  BICB    #200,(R4)       ;STRIP
2990  013422  121427  000025                  CMPB    (R4),#25        ;IS IT ^G>
2991  013426  001760                          BEQ     .INST1
2992  013430  122427  000015                  CMPB    (R4)+,#15       ;CHECK FOR CR
2993  013434  001413                          BEQ     INSTR2
2994  013436  105777  166006          2$:     TSTB    @$TPS   ;TEST FLAG
2995  013442  100375                          BPL     2$
2996  013444  117777  165776  166000          MOVB    @$TKB,@$TPB     ;ECHO CHARACTER
2997  013452  005303                          DEC     R3      ;DID YOU TYPE TOO MANY CHARS ?
2998  013454  001353                          BNE     1$
2999  013456  104401                  .INSTE: TYPE
3000  013460  015433                          MQM     ;?
3001  013462  000742                          BR      .INST1  ;RETRY
3002  013464  000002                  INSTR2: RTI
3003
3004                                          ;CONVERT ASCII STRING TO OCTAL
3005
3006  013466  011605                  .PARAM: MOV     (SP),R5 ;PUT CONTENTS OF SP INTO R5
3007  013470  012567  000162                  MOV     (R5)+,LOLIM     ;PUT LOW LIMIT INTO LOLIM
3008  013474  012567  000160                  MOV     (R5)+,HILIM     ;PUT HIGH LIMIT INTO HILIM
3009  013500  012567  000156                  MOV     (R5)+,DEVADR    ;PUT STORE LOC INTO DEVADR
3010  013504  112567  000154                  MOVB    (R5)+,LOBITS    ;PUT MASK INTO LOBITS
3011  013510  112567  000151                  MOVB    (R5)+,ADRCNT    ;PUT COUNT INTO ADRCNT
3012  013514  010516                          MOV     R5,(SP) ;RESTORE RETURN ADDR ON STACK FOR RTI
3013  013516  005005                  PARAM1: CLR     R5
3014  013520  012704  016146                  MOV     #INBUF,R4
3015  013524  122714  000015                  CMPB    #15,(R4)        ;CR ?
3016  013530  001420                          BEQ     PARERR  ;YOU TYPED CR TOO SOON !
3017  013532  121427  000060          1$:     CMPB    (R4),#60        ;LOW LIMIT ASCII 0
3018  013536  002415                          BLT     PARERR
3019  013540  121427  000067                  CMPB    (R4),#67        ;HIGH LIMIT ASCII 7
3020  013544  003012                          BGT     PARERR
3021  013546  142714  000060                  BICB    #60,(R4)        ;CONVERT TO OCTAL
3022  013552  152405                          BISB    (R4)+,R5        ;STORE AWAY ITS AN OK CHAR
3023  013554  122714  000015                  CMPB    #15,(R4)        ;CR ?
3024  013560  001414                          BEQ     LIMITS  ;NOW CHECK FOR HIGH &LOW LIMIT CONDS
```

```
3025  013562  006305                        ASL     R5      ;ALLOCATE ROOM FOR NEXT CHAR
3026  013564  006305                        ASL     R5
3027  013566  006305                        ASL     R5
3028  013570  000760                        BR      1$
3029  013572  122714  000015        PARERR: CMPB    #15,(R4)        ;CR?
3030  013576  001003                        BNE     120$
3031  013600  005737  013032                TST     @#RDSW          ;CK SWR USED
3032  013604  001023                        BNE     PARTI
3033  013606  104407                120$:   INSTER  ;RETRY
3034  013610  000742                        BR      PARAM1
3035
3036                                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
3037
3038  013612  020567  000042        LIMITS: CMP     R5,HILIM
3039  013616  101365                        BHI     PARERR  ;THE # IS TOO HIGH
3040  013620  020567  000032                CMP     R5,LOLIM
3041  013624  103762                        BLO     PARERR  ;THE # IS TOO LOW
3042  013626  136705  000032                BITB    LOBITS,R5       ;TEST BY MASKINGTHE #
3043  013632  001357                        BNE     PARERR
3044
3045                                        ;STORE NUMBER AT SPECIFIED ADDRESS
3046
3047  013634  016704  000022                MOV     DEVADR,R4       ;GET STARTING ADDR OF
3048  013640  010524                1$:     MOV     R5,(R4)+        ;STORE AT THIS ADDR
3049  013642  062705  000002                ADD     #2,R5
3050  013646  105367  000013                DECB    ADRCNT  ;HOW MANY TIMES + 2 ?
3051  013652  001372                        BNE     1$
3052  013654  000002                PARTI:  RTI
3053  013656  000000                LOLIM:  0
3054  013660  000000                HILIM:  0
3055  013662  000000                DEVADR: 0
3056  013664  00000C                LOBITS: 0
3057          013665                ADRCNT=LOBITS+1
3058
3059                                        ;SAVE PC OF TEST THAT FAILED AND R0-R5
3060
3061  013666  016667  000004  165232 .SAV05: MOV     4(SP),SAVPC
3062
3063                                        ;SAVE R0-R5
3064
3065  013674  010567  165574        SV05:   MOV     R5,$REG5
3066  013700  010467  165566                MOV     R4,$REG4
3067  013704  010367  165560                MOV     R3,$REG3
3068  013710  010267  165552                MOV     R2,$REG2
3069  013714  010167  165544                MOV     R1,$REG1
3070  013720  010067  165536                MOV     R0,$REG0
3071  013724  000002                        RTI
3072
3073                                        ;RESTORE R0-R5
3074
3075  013726  016700  165530        .RES05: MOV     $REG0,R0
3076  013732  016701  165526                MOV     $REG1,R1
3077  013736  016702  165524                MOV     $REG2,R2
3078  013742  016703  165522                MOV     $REG3,R3
3079  013746  016704  165520                MOV     $REG4,R4
3080  013752  016705  165516                MOV     $REG5,R5
```

```
3081  013756  000002                          RTI
3082
3083                                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3084
3085  013760  104401                  .CONVR: TYPE
3086  013762  015437                          MCRLF   ;CR  LF
3087  013764  017601  000000                  MOV     @(SP),R1        ;PICK UP DATA POINTER
3088  013770  062716  000002                  ADD     #2,(SP) ;SET UP SP FOR RTI
3089  013774  012167  000130                  MOV     (R1)+,WRDCNT    ;PICK UP # OF WORDS FROM TABLE
3090  014000  112167  000126          1$:     MOVB    (R1)+,CHRCNT    ;PICK UP # OF CHARS FROM TABLE
3091  014004  112167  000123                  MOVB    (R1)+,SPACNT    ;PICK UP # OF SPACES FROM TABLE
3092  014010  013167  000120                  MOV     @(R1)+,BINWRD   ;PICK UP ADDRESS OF  MSG
3093                                                                  ;FROM TABLE
3094  014014  016704  000114          2$:     MOV     BINWRD,R4       ;SAVE
3095  014020  116705  000106                  MOVB    CHRCNT,R5       ;SAVE
3096  014024  012700  016210                  MOV     #TEMP,R0        ;STARTING ADDRESS OF TEMP BLOCK
3097  014030  010403                  3$:     MOV     R4,R3           ;SAVE
3098  014032  042703  177770                  BIC     #177770,R3      ;CLR OUT UPPER BITS .. SAVE CHAR
3099  014036  062703  000260                  ADD     #260,R3 ;CONVERT TO ASCII
3100  014042  110320                          MOVB    R3,(R0)+        ;STORE AWAY
3101  014044  006204                          ASR     R4      ;SHIFT FOR NEXT #
3102  014046  006204                          ASR     R4      ;DITTO
3103  014050  006204                          ASR     R4      ;DITTO
3104  014052  005305                          DEC     R5      ;DEC CHAR COUNT
3105  014054  001365                          BNE     3$      ;DO IT AGAIN ?
3106  014056  012703  016252                  MOV     #MDATA,R3       ;STARTING ADDRESS OF MDATA BLOCK
3107  014062  114023                  4$:     MOVB    -(R0),(R3)+     ;REVERSE THE ORDER OF NUMBERS
3108  014064  105367  000042                  DECB    CHRCNT  ;DEC CHAR COUNT
3109  014070  001374                          BNE     4$      ;DO IT AGAIN ?
3110  014072  105767  000035                  TSTB    SPACNT  ;HOW MANY SPACES ?
3111  014076  001405                          BEQ     6$      ;TYPE # IF BR =0
3112  014100  112723  000240          5$:     MOVB    #240,(R3)+      ;"SPACE" IN ASCII
3113  014104  105367  000023                  DECB    SPACNT  ;DEC # OF SPACE COUNT
3114  014110  001373                          BNE     5$      ;DO IT AGAIN ?
3115  014112  105013                  6$:     CLRB    (R3)    ;INSERT "0" FOR TTY OUTPUT ROUTINE
3116  014114  104401                          TYPE
3117  014116  016252                          MDATA   ;THIS MESSAGE
3118  014120  005367  000004                  DEC     WRDCNT  ;HOW MANY #'S ?
3119  014124  001325                          BNE     1$      ;DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
3120  014126  000002                          RTI     ;RETURN TO PROGRAM
3121  014130  000000                  WRDCNT: 0
3122  014132  000000                  CHRCNT: 0
3123          014133                  SPACNT=CHRCNT+1
3124  014134  000000                  BINWRD: 0
3125
3126                                          ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
3127                                          ;BUFFER TO THE CHARACTERS "N" AND "Y".
3128                                          ;IF THE CHARACTER IS "N" CLEAR THE FLAG
3129                                          ;IF THE CHARACTER IS "Y" SET THE FLAG
3130
3131  014136  017605  000000          .SETFLG: MOV    @(SP),R5
3132  014142  122767  000116  001776          CMPB    #'N,INBUF       ;IS IT "N" ?
3133  014150  001002                          BNE     1$
3134  014152  105015                          CLRB    (R5)    ;000
3135  014154  000406                          BR      2$
3136  014156  122767  000131  001762  1$:     CMPB    #'Y,INBUF       ;IS IT "Y" ?
```

```
3137  014164  001005                          BNE     3$
3138  014166  112715  177777                  MOVB    #-1,(R5)        ;377
3139  014172  062716  000002          2$:     ADD     #2,(SP)
3140  014176  000002                          RTI
3141  014200  104407                3$:       INSTER  ;RETRY
3142  014202  000755                          BR      .SETFLG
3143                                 .SBTTL  ERROR HANDLER ROUTINE
3144
3145                                 ;;**********************************************************************
3146                                 ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3147                                 ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3148                                 ;*AND GO TO SAVIT ON ERROR
3149                                 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3150                                 ;*SW15=1        HALT ON ERROR
3151                                 ;*SW13=1        INHIBIT ERROR TYPEOUTS
3152                                 ;*SW10=1        BELL ON ERROR
3153                                 ;*SW09=1        LOOP ON ERROR
3154                                 ;*CALL
3155                                 ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
3156
3157  014204                         $ERROR:
3158  014204  105267  165173        7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
3159  014210  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
3160  014212  016777  165164  165222          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
3161  014220  032777  002000  165212          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
3162  014226  001402                          BEQ     1$              ;;NO - SKIP
3163  014230  104401  001516                  TYPE    ,$BELL          ;;RING BELL
3164  014234  005267  165152        1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
3165  014240  011667  165152                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
3166  014244  162767  000002  165144          SUB     #2,$ERRPC
3167  014252  117767  165140  165134          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
3168  014260  032777  020000  165152          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
3169  014266  001004                          BNE     20$             ;;SKIP TYPEOUTS
3170  014270  004767  000072                  JSR     PC,SAVIT        ;;GO TO USER ERROR ROUTINE
3171  014274  104401  001523                  TYPE    ,$CRLF
3172  014300                         20$:
3173  014300  122767  000001  165240          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
3174  014306  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
3175  014310  116767  165100  000004          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
3176  014316  004767  000016'                 JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
3177  014322     000                 21$:    .BYTE   0
3178  014323     000                         .BYTE   0
3179  014324  000777                 22$:    BR      22$             ;;APT ERROR LOOP
3180  014326  005777  165106        2$:     TST     @SWR            ;;HALT ON ERROR
3181  014332  100001                          BPL     3$              ;;SKIP IF CONTINUE
3182  014334  000000                          HALT                    ;;HALT ON ERROR!
3183  014336  032777  001000  165074  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
3184  014344  001402                          BEQ     4$              ;;BR IF NO
3185  014346  016716  165036                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
3186  014352  005767  165136        4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
3187  014356  001402                          BEQ     5$              ;;BR IF NONE
3188  014360  016716  165130                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
3189  014364                         5$:
3190  014364  000002                          RTI                     ;;RETURN
3191  014366  010067  164536        SAVIT:  MOV     R0,HLD0
3192  014372  010167  164534                  MOV     R1,HLD1
```

```
3193  014376  010267  164532                  MOV     R2,HLD2
3194  014402  010367  164530                  MOV     R3,HLD3
3195  014406  010467  164526                  MOV     R4,HLD4
3196  014412  010567  164524                  MOV     R5,HLD5
3197  014416  016767  164760  164520          MOV     $TSTNM,HLD6
3198
3199                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
3200
3201                                  ;;**********************************************************************
3202                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
3203                                  ;*ERROR IS TO BE REPORTED.  IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
3204                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
3205
3206  014424                          $ERRTYP:
3207  014424  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3208  014430  010046                          MOV     R0,-(SP)        ;;SAVE R0
3209  014432  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
3210  014434  153700  001414                  BISB    @#$ITEMB,R0
3211  014440  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
3212                                                                  ;;TYPE THE PC OF THE ERROR
3213  014442  016746  164750                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
3214                                                                  ;;ERROR ADDRESS
3215  014446  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3216  014450  000426                          BR      6$              ;;GET OUT
3217  014452  005300                  1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
3218  014454  006300                          ASL     R0              ;;      WORK FOR THE ERROR TABLE
3219  014456  006300                          ASL     R0
3220  014460  006300                          ASL     R0
3221  014462  062700  001652                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
3222  014466  012067  000004                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
3223  014472  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
3224  014474  104401                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
3225  014476  000000                  2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
3226  014500  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3227  014504  012067  000004          3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
3228  014510  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
3229  014512  104401                          TYPE                    ;;TYPE THE "DATA HEADER"
3230  014514  000000                  4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
3231  014516  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3232  014522  011000                  5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
3233  014524  001004                          BNE     7$              ;;GO TYPE THE DATA
3234  014526  012600                  6$:     MOV     (SP)+,R0        ;;RESTORE R0
3235  014530  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3236  014534  000207                          RTS     PC              ;;RETURN
3237  014536                          7$:
3238  014536  013046                          MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
3239  014540  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3240  014542  005710                          TST     (R0)            ;;IS THERE ANOTHER NUMBER?
3241  014544  001770                          BEQ     6$              ;;BR IF NO
3242  014546  104401  014554                  TYPE    ,8$             ;;TYPE TWO(2) SPACES
3243  014552  000771                          BR      7$              ;;LOOP
3244  014554  020040      000         8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
3245          014560                          .EVEN
3246                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3247
3248                                  ;;**********************************************************************
```

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

014560  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
014564  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
014572  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
014576  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
014602  000406                          BR      $TYPON
014604  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
014612  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
014620  112767  000005  000154  $TYPON. MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
014626  010346                          MOV     R3,-(SP)        ;;SAVE R3
014630  010446                          MOV     R4,-(SP)        ;;SAVE R4
014632  010546                          MOV     R5,-(SP)        ;;SAVE R5
014634  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
014640  005404                          NEG     R4
014642  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
014646  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
014652  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
014656  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
014662  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
014664  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
014666  000404                          BR      3$              ;;GO DO MSB
014670  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
014672  006105                          ROL     R5
014674  006105                          ROL     R5
014676  010503                          MOV     R5,R3
014700  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
014702  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
014706  100016                          BPL     7$              ;;BR IF NO
014710  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
014714  001002                          BNE     4$              ;;TEST FOR 0
014716  005704                          TST     R4              ;;SUPPRESS THIS 0?
014720  001403                          BEQ     5$              ;;BR IF YES
014722  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
014724  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
014730  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
```

```
3305  014734  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
3306  014740  104401  015000                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
3307  014744  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
3308  014750  003347                          BGT     2$              ;;BR IF MORE TO DO
3309  014752  002402                          BLT     6$              ;;BR IF DONE
3310  014754  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
3311  014756  000744                          BR      2$              ;;GO DO THE LAST DIGIT
3312  014760  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
3313  014762  012604                          MOV     (SP)+,R4        ;;RESTORE R4
3314  014764  012603                          MOV     (SP)+,R3        ;;RESTORE R3
3315  014766  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
3316  014774  012616                          MOV     (SP)+,(SP)
3317  014776  000002                          RTI                     ;;RETURN
3318  015000     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
3319  015001     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
3320  015002     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
3321  015003     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
3322  015004  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
3323                                          ;ENTER HERE ON POWER FAILURE
3324
3325
3326  015006                          $PWRDN:
3327  015006  010046                  .PFAIL: MOV     R0,-(SP)                ;SAVE R0-R5 ON PROCESSOR STACK
3328  015010  010146                          MOV     R1,-(SP)
3329  015012  010246                          MOV     R2,-(SP)
3330  015014  010346                          MOV     R3,-(SP)
3331  015016  010446                          MOV     R4,-(SP)
3332  015020  010546                          MOV     R5,-(SP)
3333  015022  016746  162776                  MOV     24,-(SP)
3334  015026  010667  164064                  MOV     SP,SAVSP                ;SAVE STACK POINTER
3335  015032  012767  015044  162764          MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
3336  015040  000000                          HALT                            ;HALT ON POWER DOWN NORMAL
3337  015042  000777                          BR      .
3338
3339                                          ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3340
3341  015044  016706  164046          RESTAR: MOV     SAVSP,SP                ;RESTORE STACK POINTER
3342  015050  012605                          MOV     (SP)+,R5                ;RESTORE R0-R5
3343  015052  012604                          MOV     (SP)+,R4
3344  015054  012603                          MOV     (SP)+,R3
3345  015056  012602                          MOV     (SP)+,R2
3346  015060  012601                          MOV     (SP)+,R1
3347  015062  012600                          MOV     (SP)+,R0
3348  015064  012767  015006  162732          MOV     #.PFAIL,24              ;SET UP FOR POWER FAILURE
3349  015072  106427  000340                  MTPS    #340
3350  015076  012706  001100                  MOV     #STACK,SP
3351  015102  005067  001102                  CLR     TEMP
3352  015106  005267  001076                  INC     TEMP
3353  015112  001375                          BNE     .-4
3354  015114  104413                          CONVRT
3355  015116  015140                          PFTAB
3356  015120  104401                          TYPE
3357  015122  015442                          MPFAIL
3358  015124  005067  164253                  CLR     $ERFLG
3359  015130  005067  164262                  CLR     $ERRPC
3360  015134  000177  163744                  JMP     @RETURN
```

```
3361  015140  000001                  PFTAB:  1
3362  015142     006     002                  .BYTE   6,2
3363  015144  000207                          RETURN
3364  015146  005015  042012  053125  MTITLE: .ASCIZ  <15><12><12>/DUV11 DZDUT-B TAPE D /<15><12>
3365  015154  030461  042040  042132
3366  015162  052125  041055  052040
3367  015170  050101  020105  020104
3368  015176  005015     000
3369  015201     015  053012  041505  MVECTO: .ASCIZ  <15><12>/VEC ADD-/
3370  015206  040440  042104  000055
3371  015214  005015  051461  020124  MREGAD: .ASCIZ  <15><12>/1ST DEV: REC CSR ADD-/
3372  015222  042504  035126  051040
3373  015230  041505  041440  051123
3374  015236  040440  042104  000055
3375  015244  005015  052515  052114  MMULT:  .ASCIZ  <15><12>/MULT DEV ? (Y OR N)-/
3376  015252  042040  053105  037440
3377  015260  024040  020131  051117
3378  015266  047040  026451     000
3379  015273     015  046012  051501  MLASTD: .ASCIZ  <15><12>/LAST DEV: REC CSR ADDR-/
3380  015300  020124  042504  035126
3381  015306  051040  041505  041440
3382  015314  051123  040440  042104
3383  015322  026522     000
3384  015325     075  042504  044526  DEVICE: .ASCIZ  /=DEVICE  /
3385  015332  042503  020040     000
3386  015337     015  051412  046105  MCOW:   .ASCIZ  <15><12>/SELECT TO RUN @ACTREG/
3387  015344  041505  020124  047524
3388  015352  051040  047125  040040
3389  015360  041501  051124  043505
3390  015366     000
3391  015367     015  047412  043126  MRANGE: .ASCIZ  <15><12>/OVFLO: RETYPE LAST DEV RXCSR ADDS-/
3392  015374  047514  051072  052105
3393  015402  050131  020105  040514
3394  015410  052123  042040  053105
3395  015416  051040  041530  051123
3396  015424  040440  042104  026523
3397  015432     000
3398  015433     040  037440     000  MQM:    .ASCIZ  /  ?/
3399  015437     015  000012          MCRLF:  .ASCIZ   <15><12>
3400  015442  043120  044501  026114  MPFAIL: .ASCIZ  /PFAIL,  RESTART AT TEST IN PROGRESS/
3401  015450  020040  042522  052123
3402  015456  051101  020124  052101
3403  015464  052040  051505  020124
3404  015472  047111  050040  047522
3405  015500  051107  051505  000123
3406  015506  005015  047105  020104  MEPASS: .ASCIZ  <15><12>/END OF PASS TAPE D/
3407  015514  043117  050040  051501
3408  015522  020123  040524  042520
3409  015530  042040     000
3410  015533     015  051012     000  MR:     .ASCIZ  <15><12>/R/
3411  015537     015  052012  051505  MTSTPC: .ASCIZ  <15><12>/TEST PC-/
3412  015544  020124  041520  000055
3413  015552  005015  047514  045503  MLOCK:  .ASCIZ  <15><12>/LOCK ON  TEST? (Y OR N)-/
3414  015560  047440  020116  052040
3415  015566  051505  037524  024040
3416  015574  020131  051117  047040
```

```
3417  015602  026451     000
3418  015605     015  021412  047440  MSYNC:  .ASCIZ  <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/
3419  015612  020106  054523  041516
3420  015620  041440  040510  051522
3421  015626  051440  046105  041505
3422  015634  042524  020104  020050
3423  015642  020061  051117  031040
3424  015650  026451     000
3425  015653     015  044412  020123  MWIRE6: .ASCIZ  <15><12>/IS SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
3426  015660  042523  020103  046530
3427  015666  052111  051440  044527
3428  015674  041524  020110  032505
3429  015702  026465  020062  047111
3430  015710  020077  054450  047440
3431  015716  020122  024516  000055
3432  015724  005015  051511  051440  MWIRE5: .ASCIZ  <15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
3433  015732  041505  051040  041505
3434  015740  051440  044527  041524
3435  015746  020110  032505  026465
3436  015754  020063  047111  020077
3437  015762  054450  047440  020122
3438  015770  024516  000055
3439  015774  005015  051511  047440  MWIRE4: .ASCIZ  <15><12>/IS OPT CLR ENABLE SWITCH E55-1 IN? (Y OR N)-/
3440  016002  052120  041440  051114
3441  016010  042440  040516  046102
3442  016016  020105  053523  052111
3443  016024  044103  042440  032465
3444  016032  030455  044440  037516
3445  016040  024040  020131  051117
3446  016046  047040  026451     000
3447  016053     015  005012  031510  MEXTJ:  .ASCIZ  <15><12><12>/H315 CONNECTOR ON ?(Y OR N)-/
3448  016060  032461  041440  047117
3449  016066  042516  052103  051117
3450  016074  047440  020116  024077
3451  016102  020131  051117  047040
3452  016110  026451     000
3453  016113     015  020012  043536  MCNTG:  .ASCIZ  <15><12>/  G  /
3454  016120  020040     000
3455  016123     040  053523  036522  MMSWR:  .ASCIZ  / SWR=   /
3456  016130  020040  000040
3457  016134  020040  047040  053505  MMNEW:  .ASCIZ  /   NEW=  /
3458  016142  020075  000040
3459                                  .EVEN
3460
3461                                  ;BUFFERS FOR INPUT-OUTPUT
3462
3463  016146  000000                  INBUF:  0
3464          016210                  .=.+40
3465  016210  000000                  TEMP:   0
3466          016252                  .=.+40
3467  016252  000000                  MDATA:  0
3468          016314                  .=.+40
3469                                  .SBTTL  SCOPE HANDLER ROUTINE
3470
3471                                  ;;**********************************************************************
3472                                  ;*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
                                ;*SW11=1         INHIBIT ITERATIONS
                                ;*SW09=1         LOOP ON ERROR
                                ;*SW08=1         LOOP ON TEST IN SWR<7:0>
                                ;*CALL
                                ;*      SCOPE            ;;SCOPE=IOT

016314                          $SCOPE:

                                        ;SCOPE LOOP AND INTERATION HANDLER

016314                          .SCOPE:
016314  004767  174376                  JSR     PC,CKSWR
016320  005067  163072                  CLR     $ERRPC          ;CLEAR LAST ERROR PC
016324  022716  003376                  CMP     #TST1+2,(SP)    ;IS SCOPE AT BEGINING OF TEST 1?
016330  001422                          BEQ     $XTSTR          ;YES NO LOOP.

016332  032777  040000  163100  TTST:   BIT     #BIT14,@SWR     ;THIS CODE IS FOR TESTING FOR BIT 14
016340  001412                          BEQ     1$              ;ON LSI WHICH SYSMAC CANNOT HANDLE
016342  016767  163034  163036          MOV     $TSTNM,$LPADR
016350  000406                          BR      1$
016352  105777  163066                  TSTB    @$TKS           ;KEYBOARD DONE?
016356  100123                          BPL     $OVER           ;BR IF NO
016360  017766  163062  177776          MOV     @$TKB,-2(SP)    ;CLEAR DONE BIT
016366  032777  040000  163044  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
016374  001114                          BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
016376  000416                $XTSTR:   BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
016400  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
016404  012737  016424  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
016412  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
016416  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
016422  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
016424  022626                5$:       CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
016426  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
016432  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
016434                          6$: ;#####END OF CODE FOR THE XOR TESTER#####
016434  032777  000400  162776          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
016442  001404                          BEQ     2$              ;;BR IF NO
016444  127767  162770  162730          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
016452  001465                          BEQ     $OVER           ;;BR IF YES
016454  105767  162723                2$:       TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
016460  001421                          BEQ     3$              ;;BR IF NO
016462  126767  162727  162713          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
016470  101015                          BHI     3$              ;;BR IF NO
016472  032777  001000  162740          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
016500  001404                          BEQ     4$              ;;BR IF NO
016502  016767  162702  162676  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
016510  000446                          BR      $OVER
016512  105067  162665                4$:       CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
016516  005067  162770                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
016522  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
```

```
3529  016524  032777  004000  162706  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
3530  016532  001011                  BNE     1$              ;;BR IF YES
3531  016534  005767  162774          TST     $PASS           ;;IF FIRST PASS OF PROGRAM
3532  016540  001406                  BEQ     1$              ;;      INHIBIT ITERATIONS
3533  016542  005267  162636          INC     $ICNT           ;;INCREMENT ITERATION COUNT
3534  016546  026767  162740  162630  CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
3535  016554  002024                  BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
3536  016556  012767  000001  162620  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
3537  016564  016767  000056  162720          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
3538  016572  105267  162604          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
3539  016576  116767  162600  162726          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
3540  016604  011667  162576                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
3541  016610  011667  162574                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
3542  016614  005067  162674                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
3543  016620  112767  000001  162567          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3544  016626  016777  162550  162606  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
3545  016634  016716  162546                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
3546  016640  000002                  4$:     RTI
3547  016642  001407                  BRW:    1407
3548  016644  000432                  BRX:    432
3549  016646  000005                  $MXCNT: 5                       ;;MAX. NUMBER OF ITERATIONS
3550                                  .SBTTL  TRAP DECODER
3551
3552                                  ;;**************************************************************
3553                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3554                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3555                                  ;*OF THE DESIRED ROUTINE.  THEN USING THE ADDRESS OBTAINED IT WILL
3556                                  ;*GO TO THAT ROUTINE.
3557
3558  016650  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
3559  016652  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
3560  016656  005740                          TST     -(R0)           ;;BACKUP BY 2
3561  016660  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
3562  016662  006300                          ASL     R0              ;;POSITION FOR INDEXING
3563  016664  016000  016704                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
3564  016670  000200                          RTS     R0              ;;GO TO ROUTINE
3565
3566
3567                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
3568
3569  016672  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
3570  016674  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
3571  016702  000002                          RTI                     ;;RESTORE THE PSW
3572
3573                                  .SBTTL  TRAP TABLE
3574
3575                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3576                                  ;*BY THE "TRAP" INSTRUCTION.
3577
3578                                  ;       ROUTINE
3579                                  ;       -------
3580  016704  016672                  $TRPAD: .WORD   $TRAP2
3581  016706  013066                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
3582  016710  014604                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3583  016712  014560                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
3584  016714  014620                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
```

```
3585
3586
3587  016716  013044                          .SCOP1  ;;CALL=SCOP1    TRAP+5(104405)
3588  016720  013350                          .INSTR  ;;CALL=INSTR    TRAP+6(104406)
3589  016722  013456                          .INSTER ;;CALL=INSTER   TRAP+7(104407)
3590  016724  013466                          .PARAM  ;;CALL=PARAM    TRAP+10(104410)
3591  016726  013666                          .SAV05  ;;CALL=SAV05    TRAP+11(104411)
3592  016730  013726                          .RES05  ;;CALL=RES05    TRAP+12(104412)
3593  016732  013760                          .CONVRT ;;CALL=CONVRT   TRAP+13(104413)
3594  016734  014136                          .SETFLG ;;CALL=SETFLG   TRAP+14(104414)
3595                                  ;*****************************************************************
3596                                  ;UTILITIES
3597                                  ;*****************************************************************
3598
3599                                  ;THIS UTILITY CALCULATES PRIORITY LEVEL
3600  016736  006367  000044          DULEV:  ASL     DUPRT   ;SHIFT LEFT
3601  016742  006367  000040                  ASL     DUPRT   ;
3602  016746  006367  000034                  ASL     DUPRT   ;
3603  016752  006367  000030                  ASL     DUPRT   ;
3604  016756  006367  000024                  ASL     DUPRT   ;
3605  016762  016767  000020  000020         MOV     DUPRT,LESS1     ;MOVE THIS TO LESS1
3606  016770  162767  000001  000012         SUB     #1,LESS1        ;CREATE LESS1
3607  016776  042767  000037  000004         BIC     #37,LESS1       ;CLEAR TNZVC
3608  017004  000207                          RTS     PC
3609  017006  000240                  DUPRT:  PR5
3610  017010  000200                  LESS1:  PR4     ;LEVEL TO ALLOW INTERRUPTS
3611
3612                                          ;NEW DU ADDRESSES
3613  017012  016767  000126  162672  DUADDR: MOV     DUBASE,RXCSR    ;XXX0
3614  017020  005267  000120                  INC     DUBASE
3615  017024  016767  000114  162662         MOV     DUBASE,HRXCSR   ;XXX1
3616  017032  005267  000106                  INC     DUBASE
3617  017036  016767  000102  162652         MOV     DUBASE,RXDBUF   ;XXX2
3618  017044  016767  000074  162650         MOV     DUBASE,PARCSR   ;XXX2
3619  017052  005267  000066                  INC     DUBASE
3620  017056  016767  000062  162634         MOV     DUBASE,HRXDBUF  ;XXX3
3621  017064  016767  000054  162632         MOV     DUBASE,HPARCSR  ;XXX3
3622  017072  005267  000046                  INC     DUBASE
3623  017076  016767  000042  162622         MOV     DUBASE,TXCSR    ;XXX4
3624  017104  005267  000034                  INC     DUBASE
3625  017110  016767  000030  162612         MOV     DUBASE,HTXCSR   ;XXX5
3626  017116  005267  000022                  INC     DUBASE
3627  017122  016767  000016  162602         MOV     DUBASE,TXDBUF   ;XXX6
3628  017130  005267  000010                  INC     DUBASE
3629  017134  016767  000004  162572         MOV     DUBASE,HTXDBUF  ;XXX7
3630  017142  000207                          RTS     PC
3631  017144  000000                  DUBASE: 0
3632
3633                                          ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3634                                          ;INFORMATION CONTAINED IN $TMP1 AND IT IS
3635                                          ;SHIFTED IN BY THE CONTENTS OF SHIFT
3636  017146  042777  040000  162552  RPOKE:  BIC     #MTDATA,@TXCSR
3637  017154  005067  162322                  CLR     $TMP2
3638  017160  006067  162314                  ROR     $TMP1   ;FORCE CARRY
3639  017164  006067  162312                  ROR     $TMP2   ;PICK UP CARRY IN BIT 15
3640  017170  006267  162306                  ASR     $TMP2   ;SHIFT INTO BIT 14
```

```
3641  017174  042767  100000  162300          BIC     #BIT15,$TMP2    ;CLR BIT 15
3642  017202  056777  162274  162516          BIS     $TMP2,@TXCSR    ;POKE MAINT DATA
3643  017210  042777  020000  162510          BIC     #CLK,@TXCSR     ;POKE CLK
3644  017216  052777  020000  162502          BIS     #CLK,@TXCSR     ;
3645  017224  005367  161672                  DEC     SHIFT
3646  017230  001346                          BNE     RPOKE
3647  017232  000207                          RTS     PC
3648
3649                                          ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3650  017234  016767  162240  162240  ODD8:   MOV     $TMP1,$TMP2     ;SAVE TEMP1
3651  017242  005067  162236                  CLR     $TMP3
3652  017246  012727  000010                  MOV     #8.,(PC)+
3653  017252  000000                  4$:     0
3654  017254  006067  162222          1$:     ROR     $TMP2
3655  017260  005567  162220                  ADC     $TMP3
3656  017264  005367  177762                  DEC     4$
3657  017270  001371                          BNE     1$
3658  017272  006067  162206                  ROR     $TMP3
3659  017276  103404                          BCS     2$
3660  017300  052767  000400  162172          BIS     #BIT8,$TMP1     ;SET ODD PARITY
3661  017306  000403                          BR      3$
3662  017310  042767  000400  162162  2$:     BIC     #BIT8,$TMP1     ;CLR EVEN PARITY
3663                                          ;$TMP1 NOW HAS ODD PARITY CHARACTER
3664  017316  000207                  3$:     RTS     PC
3665
3666                                          ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3667  017320  016767  162154  162154  EVEN8:  MOV     $TMP1,$TMP2     ;SAVE TEMP1
3668  017326  005067  162152                  CLR     $TMP3
3669  017332  012727  000010                  MOV     #8.,(PC)+
3670  017336  000000                  4$:     0
3671  017340  006067  162136          1$:     ROR     $TMP2
3672  017344  005567  162134                  ADC     $TMP3
3673  017350  005367  177762                  DEC     4$
3674  017354  001371                          BNE     1$
3675  017356  006067  162122                  ROR     $TMP3
3676  017362  103004                          BCC     2$
3677  017364  052767  000400  162106          BIS     #BIT8,$TMP1     ;SET EVEN PARITY
3678  017372  000403                          BR      3$
3679  017374  042767  000400  162076  2$:     BIC     #BIT8,$TMP1     ;CLR ODD PARITY
3680                                          ;$TMP1 NOW HAS EVEN PARITY CHARACTER
3681  017402  000207                  3$:     RTS     PC
3682  017404  062716  000002          TRPREG: ADD     #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3683                                                          ;IN MAIN PART OF THE PROGRAM
3684  017410  000002                          RTI
3685          000001                  .END
```

```
AAA     003200            1294#
ABASE = 000000             877     918
ACDW1 = 000000             877     920
ACDW2 = 000000             877     921
ACPUOP= 000000             877     892
ACTREG  001166             736#   1250*   1264*   1265*   1272*   2796    2799    2809
ADDW0 = 000000             877     922
ADDW1 = 000000             877     923
ADDW10= 000000             877     932
ADDW11= 000000             877     933
ADDW12= 000000             877     934
ADDW13= 000000             877     935
ADDW14= 000000             877     936
ADDW15= 000000             877     937
ADDW2 = 000000             877     924
ADDW3 = 000000             877     925
ADDW4 = 000000             877     926
ADDW5 = 000000             877     927
ADDW6 = 000000             877     928
ADDW7 = 000000             877     929
ADDW8 = 000000             877     930
ADDW9 = 000000             877     931
ADEVCT= 000000             877     883
ADEVM = 000000             877     919
ADRCNT= 013665            3011*   3050*   3057#
AENV  = 000000             877     888
AENVM = 000000             877     889
AFATAL= 000000             877     880
AMADR1= 000000             877     905
AMADR2= 000000             877     909
AMADR3= 000000             877     912
AMADR4= 000000             877     915
AMAMS1= 000000             877     899
AMAMS2= 000000             877     907
AMAMS3= 000000             877     910
AMAMS4= 000000             877     913
AMSGAD= 000000             877     885
AMSGLG= 000000             877     886
AMSGTY= 000000             877     879
AMTYP1= 000000             877     900
AMTYP2= 000000             877     908
AMTYP3= 000000             877     911
AMTYP4= 000000             877     914
APASS = 000000             877     882
APRIOR= 000000             877
APTCSU= 000040             537#   2926
APTENV= 000001             537#   2919    3173
APTSIZ= 000200             537#   1169
APTSPO= 000100             537#   2921
ASWREG= 000000             877     890
ATESTN= 000000             877     881
AUNIT = 000000             877     884
AUSWR = 000000             877     891
AVECT1= 000000             877     916
AVECT2= 000000             877     917
BASEAD  001154             731#   1232*   1269*   1270    1276*   1278*   2803*   2815*   2819
```

```
BASEIV  001162          734#   1242*  2804*  2816*  2821   2822*  2823   2824*  2825   2826*  2827   2828*
BBB     003026         1249    1253#
BINWRD  014134         3092*   3094   3124#
BITW  = 002000          803#    996#  1792   1853   1914   1975   2036   2097   2158   2219   2277   2330   2383
                       2436    2490  2543   2596   2649   2704   2757
BIT0  = 000001          666#    779   810    972   1003
BIT00 = 000001          656#    666
BIT01 = 000002          655#    665
BIT02 = 000004          654#    664
BIT03 = 000010          653#    663
BIT04 = 000020          652#    662
BIT05 = 000040          651#    661
BIT06 = 000100          650#    660
BIT07 = 000200          649#    659
BIT08 = 000400          648#    658   3514
BIT09 = 001000          647#    657   3183   3522
BIT1  = 000002          665#    778   971
BIT10 = 002000          646#    769   803    962    996   3161
BIT11 = 004000          645#    768   961    3529
BIT12 = 010000          644#    767   784    960    977
BIT13 = 020000          643#    766   783    802    959   976    995    3168
BIT14 = 040000          642#    765   782    801    958   975    994    3493   3500
BIT15 = 100000          641#    764   781    800    957   974    993    3641
BIT2  = 000004          664#    777   970
BIT3  = 000010          663#    776   809    969   1002
BIT4  = 000020          662#    775   808    968   1001
BIT5  = 000040          661#    774   807    967   1000
BIT6  = 000100          660#    773   806    966    999
BIT7  = 000200          659#    772   805    965    998
BIT8  = 000400          658#    771   787    804    964   980    997    3660   3662   3677   3679
BIT9  = 001000          657#    770   786    963    979
BPTVEC= 000014          673#
BREAK = 000001          810#   1003#  1359   1405   1498   1594   1690
BRW     016642         3547#
BRX     016644         3548#
CARDET= 010000          767#    960#
CCC     012622         2795    2818  2835#
CHRCNT  014132         3090*   3095  3108*  3122#  3123
CKSWR   012716         2789    2852  2859#  2891   3488
CLK   = 020000          802#    995#  1359   1365   1366   1368   1369   1405   1413   1414   1416   1417   1498
                       1506    1507  1509   1510   1594   1602   1603   1605   1606   1690   1698   1699   1701
                       1702    1787  1788   1794   1795   1805   1848   1849   1855   1856   1866   1909   1910
                       1916    1917  1927   1970   1971   1977   1978   1988   2031   2032   2038   2039   2049
                       2092    2093  2099   2100   2110   2153   2154   2160   2161   2171   2214   2215   2221
                       2222    2232  2272   2273   2279   2280   2290   2325   2326   2332   2333   2343   2378
                       2379    2385  2386   2396   2431   2432   2438   2439   2449   2485   2486   2492   2493
                       2503    2538  2539   2545   2546   2556   2591   2592   2598   2599   2609   2644   2645
                       2651    2652  2662   2699   2700   2706   2707   2717   2752   2753   2759   2760   2770
                       3643    3644
CNTLU   012772         1199    2870#
CONVRT= 104413         2792    2872  3354   3593#
COUNT   001124          711#   1371* 1378*  1410*  1425*  1436*  1450*  1452   1454   1503*  1518*  1529*  1546*
                       1548    1550  1599*  1614*  1625*  1642*  1644   1646   1695*  1710*  1721*  1738*  1740
                       1742
CR    = 000015          581#    2965  2975
CRLF  = 000200          582#    2936  2975
```

```
CTS    = 020000            766#    959#
DDISP  = 177570            588#    844     1157
DEVADR   013662           3009*   3047    3055#
DEVICE   015325           2793    3384#
DF1      002132           1028    1032    1036    1040    1086#
DH1      002067           1026    1030    1034    1075#
DISPLA   001442            844#   1157*   1165*   1190*   3160*   3544*
DISPRE   000174            685#   1165    1190
DNA    = 100000            800#    993#
DNAINT=  000040            807#   1000#
DSC    = 100000            764#    957#
DSINTE=  000040            774#    967#
DSR    = 001000            770#    963#
DSWR   = 177570            587#    843     1156
DTR    = 000002            778#    971#
DT1      002116           1027    1031    1035    1081#
DT4      002126           1039    1084#
DUADDR   017012           1231    2820    3613#
DUBASE   017144           1227    1230    2819*   3613    3614*   3615    3616*   3617    3618    3619*   3620    3621    3622*
                          3623    3624*   3625    3626*   3627    3628*   3629    3631#
DULEV    016736           1289    3600#
DUPRT    017006           1288*   3600*   3601*   3602*   3603*   3604*   3605    3609#
DURIS    001740           1055#   2823*
DURIV    001736           1054#   1238    1241    1242    2821*   2828
DUTIS    001744           1057#   2827*
DUTIV    001742           1056#   2825*
EIGHT  = 006000            795#    988#   1362    1408    1964    2208
EMTVEC=  000030            676#   1141*   1142*
EM1      001762           1025    1061#
EM2      002022           1029    1067#
EM3      002043           1033    1070#
EM4      001746           1037    1059#
ERRCNT   001114            704#
ERRVEC=  000004            669#   1154    1155*   1166*   3505    3506*   3508*   3511*
EVEN8    017320           3667#
EVEPAR=  001400            798#    991#   1408    1501    1597    1693    2266    2372    2479    2585    2693
EVPAR  = 000400            787#    980#
FIVE   = 000000            792#    985#   1693    1781    2025    2266    2319    2372    2425
FRMERR=  020000            783#    976#   1462    1465    1468    1558    1561    1564    1654    1657    1660    1750    1753
                          1756
GNS    = ****** U         3581    3582    3583    3584    3587    3588    3589    3590    3591    3592    3593    3594
HDXEN  = 000010            809#   1002#
HILIM    013660           3008*   3038    3054#
HLD0     001130            713#   1081    3191*
HLD1     001132            714#   1081    3192*
HLD2     001134            715#   3193*
HLD3     001136            716#   3194*
HLD4     001140            717#   3195*
HLD5     001142            718#   3196*
HLD6     001144            719#   3197*
HOLD     001120            709#
HPARCS   001724           1048#   3621*
HRXCSR   001714           1044#   3615*
HRXDBU   001720           1046#   3620*
HT     = 000011            579#   2934    2975
HTXCSR   001730           1050#   3625*
```

```
HTXDBU  001734          1052#   3629*
INBUF   016146          1297    1301    2984    3014    3132    3136    3463#
INIFLG  001172           744#   1200    1203*
INSTER= 104407          1305    3033    3141    3589#
INSTR = 104406          1222    1233    1243    1254    1279    1295    1308    1312    1316    1320    1331    2874    3588#
INSTR2  013464          2981    2993    3002#
IOTVEC= 000020           674#   1139*   1140*
ISYMOD= 000000           791#    984#   1401    1408    1494    1501    1590    1597    1686    1693    2364    2372    2417
                        2425    2577    2585    2630    2638
JMRBY   001153           727#   1208*   1323
KEEPAD  001156           732#   1230*   1232    1276    1278    2815
KEEPIV  001164           735#   1241*   2816
LASTAD  001160           733#   1259    1270    1284
LESS1   017010          3605*   3606*   3607*   3610#
LF    = 000012           580#   2969    2975
LIGHTS  001102           694#   2840*
LIMITS  013612          3024    3038#
LOBITS  013664          3010*   3042    3056#   3057
LOCK    001110           702#   2894
LOKFLG  001174           746#
LOLIM   013656          3007*   3040    3053#
MCNTG   016113          2869    3453#
MCOW    015337          2798    3386#
MCRLF   015437          3086    3399#
MDATA   016252          3106    3117    3467#
MEPASS  015506          2791    3406#
MEXT  = 010000           814#   1007#
MEXTJ   016053          1321    3447#
MINT  = 004000           813#   1006#   1359    1405    1498    1594    1690    1778    1839    1900    1961    2022    2083
                        2144    2205    2263    2316    2369    2422    2476    2529    2582    2635    2690    2743
MLASTD  015273          1255    3379#
MLOCK   015552          3413#
MMNEW   016134          2874    3457#
MMSWR   016123          2871    3455#
MMULT   015244          1244    3375#
MPFAIL  015442          3357    3400#
MQM     015433          3000    3398#
MR      015533          1342    3410#
MRANGE  015536?         1280    3391#
MREGAD  015214          1223    3371#
MRESET= 000400           804#    997#   1354    1356    1400    1402    1493    1495    1589    1591    1685    1687    1772
                        1774    1833    1835    1894    1896    1955    1957    2016    2018    2077    2079    2138    2140
                        2199    2201    2257    2259    2310    2312    2363    2365    2416    2418    2470    2472    2523
                        2525    2576    2578    2629    2631    2684    2686    2737    2739
MSYNC   015605          1296    3418#
MTDATA= 040000           801#    994#   1359    1405    1498    1594    1690    3636
MTITLE  015146          1202    3364#
MTSTPC  015537          1332    3411#
MULTD   001152           726#   1246    1247    2794
MVECTO  015201          1234    3369#
MWIRE4  015774          1317    3439#
MWIRE5  015724          1313    3432#
MWIRE6  015653          1309    3425#
NEXT    001106           701#
NOPAR = 000000           796#    989#   1362    1781    1842    1903    1964    2025    2086    2147    2208
ODDPAR= 001000           797#    990#   2319    2425    2532    2638    2746
```

```
ODD8    017234          3650#
ONCE    002610          1201    1204#
OPTCLR  001151           725#   1319
OUT     013024          2860    2862    2864    2868    2880#
OUTCRY  012614          2792    2831#
OUTMUL  003166          1252    1277    1288#
OVRRUN= 040000           782#    975#   1462    1558    1654    1750
PARAM = 104410          1224    1235    1256    1281    1333    2875    3590#
PARAM1  013516          3013#   3034
PARCSR  001722          1047#   1355*   1362*   1401*   1408*   1494*   1501*   1590*   1597*   1686*   1693*   1773*   1781*
                        1834*   1842*   1895*   1903*   1956*   1964*   2017*   2025*   2078*   2086*   2139*   2147*   2200*
                        2208*   2258*   2266*   2311*   2319*   2364*   2372*   2417*   2425*   2471*   2479*   2524*   2532*
                        2577*   2585*   2630*   2638*   2685*   2693*   2738*   2746*   3618*
PAREN = 001000           786#    979#
PARER = 010000           784#    977#   1437    1462    1468    1530    1558    1564    1626    1654    1660    1722    1750
                        1756
PARERR  013572          3016    3018    3020    3029#   3039    3041    3043
PARTI   013654          3032    3052#
PASCNT  001112           703#   1177*   2839*   2840    2841
PFTAB   015140          3355    3361#
PIRQ  = 177772           586#
PIRQVE= 000240           680#
POPRO = 012600           759#    952#
POP1SP= 005726           757#    950#
POP2SP= 022626           761#    954#
PRO   = 000000           603#
PR1   = 000040           604#
PR2   = 000100           605#
PR3   = 000140           606#
PR4   = 000200           607#   3610
PR5   = 000240           608#   3609
PR6   = 000300           609#
PR7   = 000340           610#
PS    = 177776           583#    584
PSW   = 177776           584#
PUSHRO= 010046           758#    951#
PUSH1S= 005746           756#    949#
PUSH2S= 024646           760#    953#
PWRVEC= 000024           675#   1145*   1146*
RDSW    013032          2870*   2880*   2882#   3031
RECACT= 004000           768#    961#
REPLAY  012520          2811    2817    2819#
RESTAR  015044          3335    3341#
RESTRT  012700          2812    2843    2850#
RESVEC= 000010           670#
RESO5 = 104412          3592#
RETURN  001104           700#   1182*   1339*   1341*   1343    3360
RING  = 040000           765#    958#
RINTEN= 000100           773#    966#
ROTADD  001170           739#   1251*   1263*   1265    1267*   1272    1275*   2806*   2809    2813*
RPOKE   017146          1374    1421    1440    1459    1514    1533    1555    1610    1629    1651    1706    1725    1747
                        3636#   3646
RTS   = 000004           777#    970#
RUNA  = ****** U           1      51      141     514
RUNB  = ****** U           1      51      141     514
RUNC  = ****** U           1      51      141     514
```

```
RUND  = 000000            1     51    141    514
RUNE  = ****** U         21     57    144    519
RUNF  = ****** U         21     57    144    519
RUNIT   012424         2797   2803#  2810
RXCSR   001712         1043#  1363*  1370*  1375   1411*  1418*  1422   1441   1504*  1511*  1515   1534   1600*
                       1607*  1611   1630   1696*  1703*  1707   1726   2833   3613*
RXDBUF  001716         1045#  1409   1428   1444   1502   1521   1537   1598   1617   1633   1694   1713   1729
                       3617*
RXDONE= 000200          772#   965#
RXERR = 100000          781#   974#  1437   1462   1465   1468   1530   1558   1561   1564   1626   1654   1657
                       1660   1722   1750   1753   1756
SAVIT   014366         3170   3191#
SAVPC   001126          712#  3061*
SAVSP   001116          705#  3334*  3341
SAVOS = 104411         3591#
SCOP1 = 104405         3587#
SEND  = 000020          808#  1001#  1778   1839   1900   1961   2022   2083   2144   2205   2263   2316   2369
                       2422   2476   2529   2582   2635   2690   2743
SEREC   001150          724#  1315
SETFLG= 104414         1245   1310   1314   1318   1322   3594#
SEVEN = 004000          794#   987#  1501   1903   2147   2693   2746
SEXMIT  001147          723#  1311
SHIFT   001122          710#  1373*  1419*  1439*  1457*  1512*  1532*  1553*  1608*  1628*  1649*  1704*  1724*
                       1745*  1785*  1802*  1846*  1863*  1907*  1924*  1968*  1985*  2029*  2046*  2090*  2107*
                       2151*  2168*  2212*  2229*  2270*  2287*  2323*  2340*  2376*  2393*  2429*  2446*  2483*
                       2500*  2536*  2553*  2589*  2606*  2642*  2659*  2697*  2714*  2750*  2767*  3645*
SIX   = 002000          793#   986#  1597   1842   2086   2479   2532   2585   2638
SPACNT= 014133         3091*  3110   3113*  3123#
SRD   = 002000          769#   962#
STACK = 001100          574#  1137   1173   1327   3350
STD   = 000010          776#   969#
STFLG   001173          745#  1176*
STKLMT= 177774          585#
STPSYN= 000400          771#   964#  1370   1418   1511   1607   1703
SVOS    013674         3065#
SWR     001440          843#  1135   1156*  1158   1164*  1171*  1187   1189*  1194   1210   1329   2861   2892
                       3161   3168   3180   3183   3493   3500   3514   3516   3522   3529
SWREG   000176          686#  1164   1189   1194   2861   2878   2885
SWREGL  013034         2873   2883#
SWO   = 000001          638#
SWOO  = 000001          628#   638   1210
SW01  = 000002          627#   637   1329
SW02  = 000004          626#   636
SW03  = 000010          625#   635
SW04  = 000020          624#   634
SW05  = 000040          623#   633
SW06  = 000100          622#   632
SW07  = 000200          621#   631
SW08  = 000400          620#   630
SW09  = 001000          619#   629   2892
SW1   = 000002          637#
SW10  = 002000          618#
SW11  = 004000          617#
SW12  = 010000          616#
SW13  = 020000          615#
SW14  = 040000          614#
```

```
SW15  = 100000            613#
SW2   = 000004            636#
SW3   = 000010            635#
SW4   = 000020            634#
SW5   = 000040            633#
SW6   = 000100            632#
SW7   = 000200            631#
SW8   = 000400            630#
SW9   = 001000            629#
SYNCNO  001146            722#   1299*  1303*
SYNEXT= 020000            790#    983#  2017   2025   2078   2086   2139   2147   2200   2208
SYNINT= 030000            789#    982#  1355   1362   1773   1781   1834   1842   1895   1903   1956   1964   2258
                         2266    2311   2319   2471   2479   2524   2532   2685   2693   2738   2746
SYNSCH= 000020            775#    968#  1363   1411   1504   1600   1696
SYSTST= 014000            815#   1008#
TBITVE= 000014            671#
TEMP    016210           3096    3351*  3352*  3465#
TKVEC = 000060            678#
TPVEC = 000064            679#
TRAPVE= 000034            677#   1143*  1144*
TRPREG  017404           3682#
TRTVEC= 000014            672#
TST1    003374           1334    1341   1353#  2851   3490
TST10   006130           1893#
TST11   006346           1954#
TST12   006564           2015#
TST13   007002           2076#
TST14   007220           2137#
TST15   007436           2198#
TST16   007654           2256#
TST17   010062           2309#
TST2    003544           1399#
TST20   010270           2362#
TST21   010476           2415#
TST22   010704           2469#
TST23   011112           2522#
TST24   011320           2575#
TST25   011526           2628#
TST26   011734           2683#
TST27   012142           2736#
TST3    004132           1492#
TST4    004520           1588#
TST5    005106           1684#
TST6    005474           1771#
TST7    005712           1832#
TTST    016332           3493#
TXCSR   001726           1049#  1354*  1356*  1359*  1365*  1366*  1368*  1369*  1400*  1402*  1405*  1413*  1414*
                         1416*  1417*  1493*  1495*  1498*  1506*  1507*  1509*  1510*  1589*  1591*  1594*  1602*
                         1603*  1605*  1606*  1685*  1687*  1690*  1698*  1699*  1701*  1702*  1772*  1774*  1778*
                         1782   1787*  1788*  1794*  1795*  1796   1805*  1807   1814   1833*  1835*  1839*  1843
                         1848*  1849*  1855*  1856*  1857   1866*  1868   1875   1894*  1896*  1900*  1904   1909*
                         1910*  1916*  1917*  1918   1927*  1929   1936   1955*  1957*  1961*  1965   1970*  1971*
                         1977*  1978*  1979   1988*  1990   1997   2016*  2018*  2022*  2026   2031*  2032*  2038*
                         2039*  2040   2049*  2051   2058   2077*  2079*  2083*  2087   2092*  2093*  2099*  2100*
                         2101   2110*  2112   2119   2138*  2140*  2144*  2148   2153*  2154*  2160*  2161*  2162
                         2171*  2173   2180   2199*  2201*  2205*  2209   2214*  2215*  2221*  2222*  2223   2232*
```

```
                        2234    2241    2257*   2259*   2263*   2267    2272*   2273*   2279*   2280*   2281    2290*   2292
                        2310*   2312*   2316*   2320    2325*   2326*   2332*   2333*   2334    2343*   2345    2363*   2365*
                        2369*   2373    2378*   2379*   2385*   2386*   2387    2396*   2398    2416*   2418*   2422*   2426
                        2431*   2432*   2438*   2439*   2440    2449*   2451    2470*   2472*   2476*   2480    2485*   2486*
                        2492*   2493*   2494    2503*   2505    2523*   2525*   2529*   2533    2538*   2539*   2545*   2546*
                        2547    2556*   2558    2576*   2578*   2582*   2586    2591*   2592*   2598*   2599*   2600    2609*
                        2611    2629*   2631*   2635*   2639    2644*   2645*   2651*   2652*   2653    2662*   2664    2684*
                        2686*   2690*   2694    2699*   2700*   2706*   2707*   2708    2717*   2719    2737*   2739*   2743*
                        2747    2752*   2753*   2759*   2760*   2761    2770*   2772    3623*   3636*   3642*   3643*   3644*
TXDBUF  001732          1051#   1783*   1844*   1905*   1966*   2027*   2088*   2149*   2210*   2268*   2321*   2374*   2427*
                        2481*   2534*   2587*   2640*   2695*   2748*   3627*
TXDONE= 000200           805#    998#
TXINTE= 000100           806#    999#
TYPE  = 104401          1202    1342    2790    2793    2798    2869    2871    2939    2982    2999    3085    3116    3163
                        3171    3207    3224    3226    3229    3231    3235    3242    3306    3356    3581#
TYPOC = 104402          3215    3239    3582#
TYPON = 104404          3584#
TYPOS = 104403          3583#
USER  = 000000           812#   1005#
VOID  = 000001           779#    972#
WRDCNT  014130          3089*   3118*   3121#
ZERO    012432          2804#
$APTHD  002136          1108    1114#
$ASTAT= ****** U         537
$ATYC   000024R          537#
$ATY1   000000R          537#
$ATY3   000006R          537#   2924
$ATY4   000016R          537#   3176
$AUTOB  001434           840#
$BASE   001602           918#
$BDADR  001422           835#
$BDDAT  001426           837#
$BELL   001516           869#   3163    3191
$CDW1   001606           920#
$CDW2   001610           921#
$CHARC  013344          2941*   2951*   2958    2967*   2972#
$CKSWR= ****** U        3587
$CMTAG  001400           823#   1132    1133    1141    1147    1148
$CM1  = 000006           855#    856#    857#    858#    859#    860#    861#
$CM2  = 000014           855#    856#    857#    858#    859#    860#    861#
$CM3  = 000006           853#    855
$CM4  = 000006           861#    862#    863#    864#    865#    866#    867#
$CPUOP  001554           892#
$CRLF   001523           871#   2940    2975    3171    3191    3207    3226    3231    3235
$DDW0   001612           922#
$DDW1   001614           923#
$DDW10  001636           932#
$DDW11  001640           933#
$DDW12  001642           934#
$DDW13  001644           935#
$DDW14  001646           936#
$DDW15  001650           937#
$DDW2   001616           924#
$DDW3   001620           925#
$DDW4   001622           926#
$DDW5   001624           927#
```

```
$DDW6   001626            928#
$DDW7   001630            929#
$DDW8   001632            930#
$DDW9   001634            931#
$DEVCT  001536            883#
$DEVM   001604            919#
$E    = 000002            535#
$ENDAD  012670           1095    2846#
$ENV    001546            537     888#   1204    2919    2980    3173
$ENVM   001547            537     889#   1169    2921    2926
$ERFLG  001403            826#   1178*   2838*   3158*   3191    3358*   3474    3518    3520    3526*   3550
$ERMAX  001415            832#   1149*   3520    3543*   3550
$ERROR  014204           1141    3157#
$ERRPC  001416            833#   1081    1084    1180*   2837*   3165*   3166*   3167    3191    3213    3359*   3489*
$ERRTB  001652           1023#   3221
$ERRTY  014424           3206#
$ERTTL  001412            830#   1179*   3164*   3191
$ESCAP  001514            868#   1148*   3186    3188    3191    3542*
$ETABL  001546            887#
$ETEND  001652            940#   1120
$FATAL  001530            537*    880#
$FFLG   000244R           537#*
$FILLC  001456            851#   2944    2975
$FILLS  001455            850#   2975
$GDADR  001420            834#
$GDDAT  001424            836#
$GTSWR= ****** U         3586
$HIBTS  002136           1115#
$ICNT   001404            827#   3533*   3534    3536*   3549
$INTAG  001435            841#
$ITEMB  001414            831#   3167*   3175    3191    3210
$LF     001524            872#   2975    3191
$LFLG   000243R           537#*
$LPADR  001406            828#   1150*   2851*   3495*   3524*   3540*   3545    3549
$LPERR  001410            829#   1151*   3185    3524    3541*   3549
$MADR1  001560            905#
$MADR2  001564            909#
$MADR3  001570            912#
$MADR4  001574            915#
$MAIL   001526            878#   1116    1120    1168    2919    3173    3539
$MAMS1  001556            899#
$MAMS2  001562            907#
$MAMS3  001566            910#
$MAMS4  001572            913#
$MBADR  002140           1116#
$MFLG   000242R           537#*
$MSGAD  001542            537*    885#
$MSGLG  001544            537*    886#
$MSGTY  001526            537*    879#
$MTYP1  001557            900#
$MTYP2  001563            908#
$MTYP3  001567            911#
$MTYP4  001573            914#
$MXCNT  016646           3537    3549#
$N    = 000000            535#   2782#
$NULL   001454            849#   2946    2975
```

```
$NWTST= 000000          1352#  1398#  1491#  1587#  1683#  1770#  1831#  1892#  1953#  2014#  2075#  2136#  2197#
                        2255#  2308#  2361#  2414#  2468#  2521#  2574#  2627#  2682#  2735#
$OCNT   015002          3278*  3307*  3320#
$OMODE  015004          3273*  3277*  3282   3285*  3296*  3322#
$OVER   016626          3498   3501   3517   3525   3535   3544#
$PASS   001534           882#  1168*  2841*  3531   3550
$PASTM  002144          1118#
$PWRDN  015006          1145   3326#
$QUES   001522           870#  2975   3191
$RDCHR= ****** U        3587
$RDDEC= ****** U        3587
$RDLIN= ****** U        3587
$RDOCT= ****** U        3587
$REGAD  001460           853#
$REG0   001462           855#  3070*  3075
$REG1   001464           856#  3069*  3076
$REG2   001466           857#  3068*  3077
$REG3   001470           858#  3067*  3078
$REG4   001472           859#  3066*  3079
$REG5   001474           860#  3065*  3080
$R2A  = ****** U        3587
$SAVRE= ****** U        3587
$SCOPE  016314          1139   3483#
$SETUP= 000017          1121#  1138   1139   1141   1143   1145   1147   1148   1150   3158   3183   3190   3484
$STUP = 177777          1121#
$SVLAD  016572          3509   3538#
$SVPC = 002136          1093#  1098
$SWR  = 177400           526#   867    868    869   1147   1148   1150   1151   1354   1400   1493   1589   1685
                        1772   1833   1894   1955   2016   2077   2138   2199   2257   2310   2363   2416   2470
                        2523   2576   2629   2684   2737   3149   3150   3151   3152   3153   3161   3168   3180
                        3183   3191   3475   3476   3477   3478   3479   3500   3512   3514   3515   3518   3519
                        3520   3527   3528   3529   3541   3544   3549
$SWREG  001550           890#  1171
$SWRMK= 000000          3479   3480   3516
$TESTN  001532           881#  3539*
$TIMES  001512           867#  1147*  3527*  3534   3537*  3549
$TKB    001446           846#  2865   2988   2996   3499
$TKS    001444           845#  2863   2986   3497
$TMP0   001476           861#
$TMP1   001500           862#  1372*  1420*  1438*  1458*  1461*  1464*  1467*  1513*  1531*  1554*  1557*  1560*
                        1563*  1609*  1627*  1650*  1653*  1656*  1659*  1705*  1723*  1746*  1749*  1752*  1755*
                        1784*  1790*  1845*  1851*  1906*  1912*  1967*  1973*  2028*  2034*  2089*  2095*  2150*
                        2156*  2211*  2217*  2269*  2275*  2322*  2328*  2375*  2381*  2428*  2434*  2482*  2488*
                        2535*  2541*  2588*  2594*  2641*  2647*  2696*  2702*  2749*  2755*  3638*  3650   3660*
                        3662*  3667   3677*  3679*
$TMP2   001502           863#  3637*  3639*  3640*  3641*  3642   3650*  3654*  3667*  3671*
$TMP3   001504           864#  3651*  3655*  3658*  3668*  3672*  3675*
$TMP4   001506           865#
$TMP5   001510           866#
$TN   = 000030           537#  1352   1354#  1398   1400#  1491   1493#  1587   1589#  1683   1685#  1770   1772#
                        1831   1833#  1892   1894#  1953   1955#  2014   2016#  2075   2077#  2136   2138#  2197
                        2199#  2255   2257#  2308   2310#  2361   2363#  2414   2416#  2468   2470#  2521   2523#
                        2574   2576#  2627   2629#  2682   2684#  2735   2737#
$TPB    001452           848#  2964*  2975   2996*
$TPFLG  001457           852#  2913   2975
$TPS    001450           847#  2962   2975   2994
```

```
$TRAP   016650           1143    3558#
$TRAP2  016672           3569#   3580
$TRP  = 000015           3573#   3582#   3583#   3584#   3585#   3587    3588#   3589#   3590#   3591#   3592#   3593#   3594#
                         3595#
$TRPAD  016704           3563    3580#
$TSTM   002142           1117#
$TSTNM  001402            825#   1181*   1336    1339    2836*   3160    3191    3197    3474    3495    3516    3538*   3539
                         3544    3550
$TYPBN= ****** U         3585
$TYPDS= ****** U         3585
$TYPE   013066            537    2913#   3573    3581
$TYPEC  013300           2943    2950    2957    2962#   2963
$TYPEX  013346           2968    2970    2973#
$TYPOC  014604           3276#   3582
$TYPON  014620           3275    3278#   3584
$TYPOS  014560           3271#   3583
$UNIT   001540            884#
$UNITM  002146           1119#
$USWR   001552            891#   1206
$VECT1  001576            916#
$VECT2  001600            917#
$XTSTR  016376           3491    3503#
$OFILL  015003           3272*   3276*   3286    3321#
$40CAT= ****** U         3170    3500
      = 017412            537#    571#    684#    687#    692#    747#    748#    822#    873    1079#   1093    1094#   1096#
                         1098#   1104    1105#   1107#   1109#   1136    1150    1151    1376    1423    1433    1442    1446
                         1516    1526    1535    1539    1612    1622    1631    1635    1708    1718    1727    1731    1799
                         1810    1815    1860    1871    1876    1921    1932    1937    1982    1993    1998    2043    2054
                         2059    2104    2115    2120    2165    2176    2181    2226    2237    2242    2284    2295    2337
                         2348    2390    2401    2443    2454    2497    2508    2550    2561    2603    2614    2656    2667
                         2711    2722    2764    2775    2975    3191    3245#   3337    3353    3464#   3466#   3468#   3549
                         3550
.BEGIN  003310           1198    1209    1212    1327#   2853
.CONVR  013760           3085#   3593
.EOP    012350           2788#
.INSTE  013456           2999#   3589
.INSTR  013350           2978#   3588
.INST1  013370           2982#   2991    3001
.MSG    013372           2865*   2866*   2867    2978*   2983#
.PARAM  013466           3006#   3590
.PFAIL  015006           1175    3327#   3348
.RESO5  013726           3075#   3592
.SAVO5  013666           3061#   3591
.SCOPE  016314           3487#
.SCOP1  013044           2891#   3587
.SETFL  014136           3131#   3142    3594
.START  002152            688    1130#   1182    2802
.$ASTA= ****** U          537
.$X   = 002136           1104#   1109
```

```
COMMEN    681#
ENDCOM    681#
ERROR     575#  1377  1424  1434  1443  1447  1517  1527  1536  1540  1613  1623  1632  1636  1709
         1719  1728  1732  1800  1811  1816  1861  1872  1877  1922  1933  1938  1983  1994  1999
         2044  2055  2060  2105  2116  2121  2166  2177  2182  2227  2238  2243  2285  2296  2338
         2349  2391  2402  2444  2455  2498  2509  2551  2562  2604  2615  2657  2668  2712  2723
         2765  2776
ESCAPE    681#
GETPRI    681#
GETSWR    681#
MULT      681#
NEWTST    681#  1352  1398  1491  1587  1683  1770  1831  1892  1953  2014  2075  2136  2197  2255
         2308  2361  2414  2468  2521  2574  2627  2682  2735
POP       537   681#
PRGEND    559#  2781
PRGFRT    538#   560
PUSH      537   681#
PUSSYF    556#
REPORT    526#   681#
RSETUP    543#  1354  1400  1493  1589  1685
SC        535#  3484
SCOPE     576#  1353  1399  1492  1588  1684  1771  1832  1893  1954  2015  2076  2137  2198  2256
         2309  2362  2415  2469  2522  2575  2628  2683  2736  2788
SC1       535#  3546
SETPRI    681#
SETTRA    526#  3573#  3582  3583  3584  3587  3588  3589  3590  3591  3592  3593  3594
SETUP     681#  1130
SKIP      681#
SLASH     681#
SPACE     681#
STARS     537   681#   818   873   876  1091  1101  1103  1110  1352  1398  1491  1587  1683  1770
         1831  1892  1953  2014  2075  2136  2197  2255  2308  2361  2414  2468  2521  2574  2627
         2682  2735  2898  3145  3201  3248  3471  3552
SWRSU     681#  1152#
TRMTRP   3573#
TSETUP    544#  1772  1833  1894  1955  2016  2077  2138  2199  2257  2310  2363  2416  2470  2523
         2576  2629  2684  2737
TYPBIN    681#
TYPDEC    681#
TYPNAM    681#
TYPNUM    681#
TYPOCS    681#
TYPOCT    681#  3213  3237
TYPTXT    681#
$BEGIN    535#  1324
$BINAR    555#
$BUFFE    537#  3460
$CABLE    557#
$CATCH    529#   571
$CK       536#  2854
$CLRVE    535#  1204
$CONVR    537#  3082
$DNA      553#  1759  1820  1881  1942  2003  2064  2125  2186
$EOP      536#  2781
$GETFL    535#  1243  1308  1312  1316  1320
$GETPA    535#  1222  1233  1253  1279  1331
```

```
$GETSY    535#   1290
$HEADE    526#    560
$INSTR    537#   2975
$ISOB     551#
$MATCH    548#
$MRR      542#
$MRRW     540#
$MRW      541#
$MSG      537#   3364
$NEED     558#   3191
$PARAM    537#   3003
$PFAIL    537#   3323
$POKE     546#   1787   1793   1848   1854   1909   1915   1970   1976   2031   2037   2092   2098   2153   2159
          2214   2220   2272   2278   2325   2331   2378   2384   2431   2437   2485   2491   2538   2544   2591
          2597   2644   2650   2699   2705   2752   2758
$POKER    547#   1367   1415   1508   1604   1700
$RCNET    558#   1381   1471   1567   1663
$RECAC    546#
$REG      537#   3058
$RESET    539#   1354   1356   1400   1402   1493   1495   1589   1591   1685   1687   1772   1774   1833   1835
          1894   1896   1955   1957   2016   2018   2077   2079   2138   2140   2199   2201   2257   2259   2310
          2312   2363   2365   2416   2418   2470   2472   2523   2525   2576   2578   2629   2631   2684   2686
          2737   2739
$RXACT    545#
$SCOPE    535#   3469
$SCOP1    536#   2888
$SETFL    535#   3125
$SETVE    530#    681
$START    535#   1122
$STRIP    552#   1345
$SYMBO    526#    749    942
$SYNCR    548#   1363   1411   1504   1600   1696
$TRPAR    554#   2247   2300   2353   2406   2460   2513   2566   2619   2674   2727
$TSTNO    535#   1352   1398   1491   1587   1683   1770   1831   1892   1953   2014   2075   2136   2197   2255
          2308   2361   2414   2468   2521   2574   2627   2682   2735
$UNIBU    538#
$VARIA    533#    691
$WORDF    549#
$WORDO    548#
$WORDP    550#
$$CMRE    816#    855    856    857    858    859    860
$$CMTM    816#    861    862    863    864    865    866
$$ESCA    681#
$$NEWT    681#   1352   1398   1491   1587   1683   1770   1831   1892   1953   2014   2075   2136   2197   2255
          2308   2361   2414   2468   2521   2574   2627   2682   2735
$$SET    3573#  3582   3583   3584   3587   3588   3589   3590   3591   3592   3593   3594
$$SETM   1168#
$$SKIP    681#
.EQUAT    526#    571
.HEADE    526#
.SETUP    526#   1121
.$ACT1    526#   1089
.$APTB    526#    874#
.$APTH    526#   1099
.$APTY    526#    537
.$CATC    526#
```

```
.$CMTA    526#    816
.$EOP     526#
.$ERRO    526#   3143
.$ERRT    526#   3199
.$POWE    526#
.$SCOP    526#   3469
.$TRAP    526#   3550
.$TYPE    526#   2896
.$TYPO    526#   3246


. ABS.   017412     000


 ERRORS DETECTED:   0

 DZDUTB,DZDUTB/SOL/CRF=DZDUT1/EQ:RUND,DZDUT2,DZDUTB
 RUN-TIME:  21 12 1 SECONDS
 RUN-TIME RATIO:  255/35=7.1
 CORE USED:   30K  (59 PAGES)
```