DU11

OFF-LINE RECEIVER TESTS

CZDUBD0

AH-8681D-MC

COPYRIGHT 73–78

FICHE 1 OF 1

JAN 1979

digital

MADE IN USA

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

PRODUCT CODE: AC-8680D-MC

PRODUCT NAME: CZDUBD0 DU11 OFFLINE RECEIVER TESTS

RELEASE DATE: JUN 1978

MAINTAINER : DIAGNOSTICS

GENERAL DESCRIPTION

THIS DIAGNOSTIC CAN CHAIN 16 DU11'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.

1.   THE DU11 OFFLINE RECEIVER TESTS VERIFY THAT THE RECEIVER
        CHIP/LOGIC WORKS PROPERLY

2.   REQUIREMENTS

        PDP-11 FAMILY STANDARD COMPUTER WITH OR WITHOUT HARDWARE SWITCH
        REGISTER (LOC. 177570)

        DU11 SYNCRONOUS/ISOCRONOUS OPTION

        ONE CONSOLE TELETYPE OR EQUIVALENT

2.2   STORAGE

        THE PROGRAM LOADS AND RUNS IN 8K OF MEMORY.

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

+++

        NOTE: BEFORE PROCEEDING IT IS IMPORTANT TO
                                TO REALIZE IF ONE DOESNOT
                                HAVE THE DU11 SET UP TO THE
                                DEFAULT PARAMETERS (SEE SECTION
                                8 OF THIS DOCUMENT) , THEN ONE MUST

SET SW00 = 1, AND ANSWER THE PARAMETER
QUESTION ROUTINE.

++

4.1   CONTROL SWITCH SETTINGS

        NOTE: SOFTWARE SWITCH REGISTER IS DEFINED AS LOC. 176, WHILE
              THE SOFTWARE DISPLAY REGISTER IS DEFINED AS LOC. 174.

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)

        SW02=1
        NOTE1: IN GENERAL SW01 WILL BE USED WHEN SW02=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 LOAD ADDRESS 000200

        4.3.1.3 CLEAR CONSOLE SWITCHES

        4.3.1.4 PRESS START

        4.3.1.5 THE PROGRAM WILL TYPE 'DU11 CZDUB-D TAPE B' (ONCE ONLY)

        NOTE:IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
             WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:

SWR=XXXXXX NEW=   (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

4.3.1.7 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 LOAD ADDRESS   000200

4.3.3.2 SET SW00=1

4.3.3.3 PRESS START

NOTE:IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
     WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
     SWR=XXXXXX NEW=   (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

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 DU11 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 DU11 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 DU11 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 DU11 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 ''DU PRIORITY LEVEL-'' AND
WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.13   TYPE IN THE APPROPRIATE DEVICE PRIORITY LEVEL OF THE
DU11 OR DU11'S TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>
(NOTE THAT ALL MULTIPLE DEVICES MUST BE AT THE SAME PRIORITY
LEVEL).   IE ''5''

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

4.3.3.14   THE PROGRAM WILL TYPE ''# OF SYNC CHARS
SELECTED (1 OR 2)-'' AND WAIT FOR AN INPUT FROM THE TELETYPE
KEYBOARD

4.3.3.15   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 4.3.3.14

4.3.3.16   THE PROGRAM WILL TYPE '' IS SEC XMIT JUMPER #6 IN ? (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 THAT ALL MULTIPLE DEVICES
MUST BE THE SAME)

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

4.3.3.18   THE PROGRAM WILL TYPE ''IS SEC REC JUMPER # 5 IN ?
(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 ''IS OPT CLR ENABLE JUMPER
4 IN ? (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 ''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

4.3.3.23   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 4.3.3.22

4.3.3.24   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 LOAD 000200

4.3.4.2 SET SW01=1

4.3.4.3 PRESS START
NOTE:IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
     WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
     SWR=XXXXXX NEW=   (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

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 SW02 =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 LOAD ADDRESS 000200

       4.3.5.2 SET SW02 =1
            NOTE: IT MAY BE ADVANTAGEOUS TO SET SW01=1 (OPTIONAL)

       4.3.5.3 PRESS START

       NOTE:IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
            WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
            SWR=XXXXXX NEW=   (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

       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

4.4   STATUS MAP

THE STATUS MAP IS AN AREA OF THE DU11 DIAGNOSTICS,
WHICH WILL ALLOW THE TRANSFER OF PARAMETERS BETWEEN DIAGNOSTICS.
IF YOU WISH TO TEST A DU11 , WHICH IS NOT AT THE DEFAULT
VALUES ,YOU NEED ONLY GO THROUGH THE TEDIOUS QUESTIONING
AND ANSWERING ROUTINE ONCE.

THE FOLLOWING COMBINATIONS OF SWITCH REGISTER SETTINGS WILL
ALLOW YOU ACCESS TO THE STATUS MAP.

1) SW07=1

2) START AT 200

3) THE DIAGNOSTIC WILL GO TO THE STATUS MAP
   AND BYPASS ALL OF THE QUESTIONING ROUTINE.

NOTE: IT IS EXTREMELY IMPORTANT THAT EITHER YOU HAVE
JUST ANSWERED THESE QUESTIONS DURING A PRIOR DIAGNOSTIC
OR THAT YOU HAVE MANUALLY ENTERED THE CORRECT VALUES
FOR VECTOR ADDRESSES ETC.,IN THE AREA DESIGNATED
FOR THE STATUS MAP. IT IS IMPORTANT THAT THIS BE PERFORMED
BEFORE STARTING AT 200.

THE DIAGNOSTIC HAS NO METHOD TO DETERMINE
THAT THE STATUS MAP HAS INDEED BEEN LOADED CORRECTLY.
THE DIAGNOSTIC ASSUMES THAT WHEN SW07=1 THE VALUES IN
THE STATUS MAP ARE THE VALUES TO BE USED .
THESE VALUES CAN BE THE WRONG VALUES , BUT THE
DIAGNOSTIC WILL NOT REALIZE THAT A MISTAKE HAS BEEN MADE.

IF BOTH SW07 AND SW00 (SWITCH REGISTER SWITCHES)
ARE SET ,(EQUAL TO 1), THE PROGRAM WILL IGNORE SW00 AND
SEEING SW07 SET, THE VALUES FROM THE STATUS MAP WILL BE
USED. TO USE THE DEFAULT VALUES FOR THE DU11'S
THE OPERATOR MUST SET SW00=0 AND SW07=0. THE USE OF SW00
IS EXPLAINED IN GREATER DETAIL IN SECTION   4.3
OF THIS DOCUMENT.

THE FIRST TIME A PROGRAM IS LOADED OR THE FIRST TIME
A PROGRAM IS ALTERED VIA THE PARAMETER RESELECTION QUESTION
 AND ANSWER ROUTINE,   A PARTIAL STATUS MAP WILL BE PRINTED.
THIS MAP WILL BE PRINTED ONCE FOR ANY COMBINATION OF SWITCHES
EXCEPT SW01. RESTARTING THE PROGRAM WILLNOT PRINT OUT A
MAP UNLESS THE PROGRAM PARAMETERS ARE BEING RESELECTED
BY PUTTING SW00=1.(ON)

THE MAP WILL LOOK LIKE:

STATUS MAP

1300/ 177777

1302/ 000000

1304/ 177777

THE BYTES ARE DEFINED AS FOLLOWS:

1300   THE NUMBER OF SYNCHRONOUS CHARACTERS REQUIRED FOR
        SYNCHRONIZATION.
1301   SEC TRANSMIT JUMPER
1302   SEC RECEIVER JUMPER
1303   OPTIONAL JUMPER
1304   MULTIPLE DEVICES (NO=0 , YES=   1)
1305   EXTERNAL MODEM BYPASS? (NO=0 ,YES=   1)

IF THE BYTE IS 0 , THE JUMPER IS NOT CONNECTED
AND IF THE BYTE IS 377 ETC. THE JUMPER SHOULD BE CONNECTED.

5.   OPERATING PROCEDURE

        IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH
        REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS
        THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER.
        IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES
        AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH
        REGISTER (LOC. 176) IS USED.

        CONTROL:

        THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH
        REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY
        DOING THE FOLLOWING:

        1) TYPE CONTROL G <^G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO
           LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
        2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS
           OF THE SOFTWARE SWITCH REGISTER.)

        3) AFTER THE ''NEW='' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
           OF THE FOLLOWING AT THE TTY:

           A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>.
              (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS
              WILL BE ALLOWED)
              IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH
              REGISTER CONTENTS WILL NOT BE CHANGED.

           B) IF A CONTROL U <^U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU
              BACK TO STEP 2.

5.1   OPERATIONAL SWITCH SETTINGS
        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

```
              SW08 =1           LOOP ON ERROR
              SW07 =1           USE STATUS MAP PARAMETERS
              SW02 =1           LOCK ON TEST
              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
        THERE ARE FOUR DISTINCT ERROR TYPEOUTS

        NOTE: IF THE SOFTWARE SWITCH REGISTER IS TO BE CHANGED AFTER A HALT
        THE THE OPERATOR IS REQUIRED TO TYPE A <^G> BEFORE DEPRESSING CONTINUE.
        THE FOLLOWING WILL BE TYPED:
        SWR=XXXXXX NEW=   (REFER TO SECTION 5. FOR OPERATOR OPTION)

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...LOAD 000200 IN SWR AND DEPRESS START....
                 THE PROGRAM WILL CONTINUE WITH THE DEVICE IT WAS IN BEFORE HALTING.

         7.2.2.2 ....OR .....LOAD 000200 WITH SW00 =1 AND DEPRESS START....
                 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.......LOAD & START AT 0002C0

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 11/20 MACHINE.
         IF RUNNING ON AN 11/40 OR AN 11/45 'HOLD:'' MUST BE PATCHED TO 40   .

         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: 160040

         VECTOR ADDRESS-   DURIV: 770

         ARE YOU RUNNING MULTIPLE DEVICES ?- NO   MULTD: 0

         LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- LASTADD: 0

         DU PRIORITY LEVEL- LEVEL 5   DUPRT: LEVEL 5

         # OF SYNC CHARS SELECTED - 2   SYNCNO: 377

         IS SEC XMIT JUMPER # 6 IN ?- YES   SEXMIT: 377

         IS SEC REC JUMPER # 5 IN ?- YES   SEREC: 377

         IS OPT CLR ENABLE JUMPER # 4 IN ?- YES OPTCLR: 377

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

9.            PROGRAM DESCRIPTION

9.1           THIS PROGRAM PERFORMS THE OFFLINE RECEIVER SECTION TESTING
              OF THE DEVICE
              SEE LISTING FOR DETAILS


10.           FLOW CHARTS: RECEIVER FLOW,TRANSMITTER FLOW,TRANSMITTER & RECEIVER FLOW

11.           LISTINGS
```

```
 674
 675  000000' 000000G              D
```

```
676                              .ENABLE ABS
677
678                              ;DU11 DZDUB-C TAPE B
679                              ;COPYRIGHT 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
680
681                                      ;STARTING PROCEDURE
682                                      ;LOAD PROGRAM
683                                      ;PRESS START
684                                      ;PROGRAM WILL TYPE 'DU11 DZDUB-C TAPE B ''
685                                      ;PROGRAM WILL TYPE 'R'' TO INDICATE THAT TESTING HAS STARTED
686                                      ;AT THE END OF A PASS, PROGRAM WILL TYPE 'END OF PASS TAPE B''
687                                      ;AND THEN RESUME TESTING
688
689
690                              ;SWITCH REGISTER OPTIONS
691
692        100000                SW15=100000            ;=1,HALT ON ERROR
693        040000                SW14=40000             ;=1,LOOP ON CURRENT TEST
694        020000                SW13=20000             ;=1,INHIBIT ERROR TYPEOUT
695        010000                SW12=10000
696        004000                SW11=4000              ;=1,INHIBIT ITERATIONS
697        002000                SW10=2000              ;=1,ESCAPE TO NEXT TEST ON ERROR
698        001000                SW09=1000              ;=1,LOOP WITH CURRENT DATA
699        000400                SW08=400               ;=1,LOOP ON ERROR
700        000200                SW07=200               ;+  =1, USE STATUS MAP
701        000100                SW06=100
702        000040                SW05=40
703        000020                SW04=20
704        000010                SW03=10
705        000004                SW02=4                 ;LOCK ON TEST SELECT
706        000002                SW01=2                 ;RESTART PROGRAM AT SELECTED TEST
707        000001                SW00=1                 ;RESELECT VECTOR AND CONTROL REGISTER
708                                                     ;ADDRESS AFTER PROGRAM RESTART
709
```

```
710
711                                    ;REGISTER DEFINITIONS
712
713            000000                  R0=%0                 ;GENERAL REGISTER
714            000001                  R1=%1                 ;GENERAL REGISTER
715            000002                  R2=%2                 ;GENERAL REGISTER
716            000003                  R3=%3                 ;GENERAL REGISTER
717            000004                  R4=%4                 ;GENERAL REGISTER
718            000005                  R5=%5                 ;GENERAL REGISTER
719            000006                  SP=%6                 ;PROCESSOR STACK POINTER
720            000007                  PC=%7                 ;PROGRAM COUNTER
721
722                                    ;LOCATION EQUIVALENCIES
723
724            177570                  DSWR=177570           ;HARDWARE SWITCH REGISTER LOC.
725            177570                  DLIGHTS=177570        ;HARDWARE DISPLAY REGISTER LOC.
726            177776                  PS=177776             ;PROCESSOR STATUS WORD
727            001100                  STACK=1100            ;START OF PROCESSOR STACK
728
729                                    ;INSTRUCTION DEFINITIONS
730
731            005746                  PUSH1SP=5746          ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
732            005726                  POP1SP=5726           ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
733            010046                  PUSHR0=10046          ;SAVE R0 ON STACK =MOV R0,-(SP)
734            012600                  POPR0=12600           ;RESTORE R0 FROM STACK =MOV (SP)+,R0
735            024646                  PUSH2SP=24646         ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
736            022626                  POP2SP=22626          ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
737                                    .EQUIV  EMT,HLT ;BASIC DEFINITION OF ERROR CALL
738
739
740            100000                  BIT15=100000
741            040000                  BIT14=40000
742            020000                  BIT13=20000
743            010000                  BIT12=10000
744            004000                  BIT11=4000
745            002000                  BIT10=2000
746            001000                  BIT9=1000
747            000400                  BIT8=400
748            000200                  BIT7=200
749            000100                  BIT6=100
750            000040                  BIT5=40
751            000020                  BIT4=20
752            000010                  BIT3=10
753            000004                  BIT2=4
754            000002                  BIT1=2
755            000001                  BIT0=1
756
757                                    ;PROCESSER LEVELS
758            000340                  LEVEL7=340
759            000300                  LEVEL6=300
760            000240                  LEVEL5=240
761            000200                  LEVEL4=200
762            000140                  LEVEL3=140
763            000100                  LEVEL2=100
764            000040                  LEVEL1=040
765            000000                  LEVEL0=000
```

```
766                                    ;REGISTER DEFINITIONS
767                                         ;RXCSR BIT DEFINITIONS
768          100000            DSC=BIT15              ;DATA SET CHANGE
769          040000            RING=BIT14            ;RING
770          020000            CTS=BIT13            ;CLR TO SEND
771          010000            CARDET=BIT12          ;CARRIER DETECT
772          004000            RECACT=BIT11          ;REC ACTIVE
773          002000            SRD=BIT10            ;SEC REC DATA
774          001000            DSR=BIT9            ;DATA SET RDY
775          000400            STPSYN=BIT8          ;STRIP SYNC
776          000200            RXDONE=BIT7          ;REC DONE
777          000100            RINTEN=BIT6          ;REC INTR ENABLE
778          000040            DSINTE=BIT5          ;DSC INTR ENABLE
779          000020            SYNSCH=BIT4          ;SYNC SEARCH
780          000010            STD=BIT3            ;SEC XMIT DATA
781          000004            RTS=BIT2            ;REQ TO SEND
782          000002            DTR=BIT1            ;DATA TERM RDY
783          000001            VOID=BIT0
784                                         ;RXDBUF BIT DEFINITIONS
785          100000            RXERR=BIT15          ;REC ERROR
786          040000            OVRRUN=BIT14          ;OVERRUN
787          020000            FRMERR=BIT13          ;FRAME ERROR
788          010000            PARER=BIT12          ;PARITY ERROR
789                                         ;PARCSR BIT DEFINITIONS
790          001000            PAREN=BIT9            ;PARITY ENABLE
791          000400            EVPAR=BIT8            ;EVEN PARITY SENSE
792                                         ;PARCSR WRD DEFINITIONS
793          030000            SYNINT=30000          ;SYNC EXTERNAL MODE
794          020000            SYNEXT=20000          ;SYNC INTERNAL MODE
795          000000            ISYMOD=0            ;ISOC MODE
796          000000            FIVE=0              ;WORD LENGTH 5 BITS
797          002000            SIX=2000            ;WORD LENGTH 6 BITS
798          004000            SEVEN=4000          ;WORD LENGTH 7 BITS
799          006000            EIGHT=6000          ;WORD LENGTH 8 BITS
800          000000            NOPAR=0            ;NO PARITY
801          001000            ODDPAR=1000          ;ODD PARITY
802          001400            EVEPAR=1400          ;EVEN PARITY
803                                         ;TXCSR BIT DEFINITIONS
804          100000            DNA=BIT15            ;DATA NOT AVAILABLE
805          040000            MTDATA=BIT14          ;MAINT DATA
806          020000            CLK=BIT13            ;CLK
807          002000            BITW=BIT10          ;BIT WINDOW
808          000400            MRESET=BIT8          ;MASTER RESET
809          000200            TXDONE=BIT7          ;XMIT DONE
810          000100            TXINTE=BIT6          ;XMIT INTR ENABLE
811          000040            DNAINTE=BIT5          ;DNA INTR ENAB
812          000020            SEND=BIT4            ;SEND
813          000010            HDXEN=BIT3            ;HDX/FDX
814          000001            BREAK=BIT0          ;BREAK
815                                         ;TXCSR WRD DEFINITIONS
816          000000            USER=0              ;USER MODE
817          004000            MINT=4000            ;MAINT INT MODE
818          010000            MEXT=10000          ;MAINT EXT MODE
819          014000            SYSTST=14000          ;SYSTEM TEST MODE
820                                    ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
```

```
821                                              ;STANDARD INTERRUPT VECTORS
822
823
824            000024                            .=24
825   000024   016104                                    .PFAIL                  ;POWER FAIL HANDLER
826   000026   000340                                    340                     ;SERVICE AT LEVEL 7
827   000030   015634                                    .HLT                    ;ERROR HANDLER
828   000032   000340                                    340                     ;SERVICE AT LEVEL 7
829   000034   015602                                    .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
830   000036   000340                                    340                     ;SERVICE AT LEVEL 7
831
832                                              ;SOFTWARE SWITCH REGISTER
833
834            000174                            .=174
835   000174   000000                            DISPREG:.WORD 0                  ;SOFTWARE DISPLAY REG.
836   000176   000000                            SWREG:  .WORD 0                  ;SOFTWARE SWITCH REGISTER
837   000200   000167  001214                    JMP     .START                  ;GO TO START OF PROGRAM
838
839
840
841            001100                            .=1100
842
843                                              ;INDIRECT POINTERS
844
845   001100   177570                            SWR:    177570                   ;SWITCH REGISTER POINTER
846   001102   177570                            LIGHTS:177570                    ;DISPLAY REGISTER POINTER
847   001104   177560                            TKCSR:  177560                   ;TELETYPE KEYBOARD CONTROL REGISTER
848   001106   177562                            TKDBR:  177562                   ;TELETYPE KEYBOARD DATA BUFFER
849   001110   177564                            TPCSR:  177564                   ;TELEPRINTER CONTROL REGISTER
850   001112   177566                            TPDBR:  177566                   ;TELEPRINTER DATA BUFFER
851
852                                              ;PROGRAM CONTROL PARAMETERS
853
854   001114   000000                            RTRN:   0                        ;SCOPE ADDRESS FOR LOOP ON TEST
855   001116   000000                            NEXT:   0                        ;ADDRESS OF NEXT TEST TO BE EXECUTED
856   001120   000000                            LOCK:   0                        ;ADDRESS FOR LOCK ON CURRENT DATA
857   001122   000000                            ICOUNT: 0                        ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
858   001124   000000                            LPCNT:  0                        ;NUMBER OF ITEREATIONS COMPLETED
859   001126   000000                            TSTNO:  0                        ;NUMBER OF TEST IN PROGRESS
860   001130   000000                            PASCNT: 0                        ;NUMBER OF PASSES COMPLETED
861   001132   000000                            ERRCNT: 0                        ;TOTAL NUMBER OF ERRORS
862   001134   000000                            LSTERR: 0                        ;PC OF LAST ERROR CALL
863
864                                              ;PROGRAM VARIABLES
865
866   001136   000020                            HOLD:   20                       ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
867   001140   000000                            SHIFT:  0                        ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
868   001142   000000                            COUNT:  0                        ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
869   001144   000000                            TEMP1:  0                        ;TEMPORARY STORAGE
870   001146   000000                            TEMP2:  0                        ;TEMPORARY STORAGE
871   001150   000000                            TEMP3:  0                        ;TEMPORARY STORAGE
872   001152   000000                            TEMP4:  0                        ;TEMPORARY STORAGE
873   001154   000000                            TEMP5:  0                        ;TEMPORARY STORAGE
874   001156   000000                            SAVR0:  0                        ;R0 STORAGE
875   001160   000000                            SAVR1:  0                        ;R1 STORAGE
876   001162   000000                            SAVR2:  0                        ;R2 STORAGE
```

```
877  001164  000000           SAVR3:  0              ;R3 STORAGE
878  001166  000000           SAVR4:  0              ;R4 STORAGE
879  001170  000000           SAVR5:  0              ;R5 STORAGE
880  001172  000000           SAVSP:  0              ;STACK POINTER STORAGE
881  001174  000000           SAVPC:  0              ;PROGRAM COUNTER STORAGE
```

```
882                                      ;PROGRAM CONVERSATIONAL PARAMETERS
883   001176      377         SYNCNO: .BYTE   377      ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
884   001177      377         SEXMIT: .BYTE   377      ;SEC XMIT JUMPER "IN"
885   001200      377         SEREC:  .BYTE   377      ;SEC REC JUMPER "IN"
886   001201      377         OPTCLR: .BYTE   377      ;OPTIONAL JUMPER CLR "IN"
887   001202      000         MULTD:  .BYTE   0        ;NO MULTIPLE DEVICE FLAG
888   001203      377         JMRBY:  .BYTE   377      ;EXTERNAL MODEM BYPASS JUMPER "IN"
889                           .EVEN
890
891                                      ;PROGRAM MULTIPLE DEVICE PARAMETERS
892   001204      000000      BASEADD:        0        ;PROG CONTROLLED 1ST DEVICE ADDR
893   001206      000000      KEEPADD:        0        ;SAVED 1ST DEVICE ADDR
894   001210      000000      LASTADD:        0        ;LAST DEVICE RXCSR ADDR
895   001212      000000      BASEIV:         0        ;PROG CONTROLLED IV
896   001214      000000      KEEPIV:         0        ;SAVED INTR VECTOR
897   001216      000000      ACTREG:         0        ;ACTIVE REGISTER ...MODIFY THIS
898                                                    ;LOCATION TO DISQUALIFY OR QUALIFY
899                                                    ;DEVICES (1= RUN,..0= DON'T RUN)
900   001220      000000      ROTADD:         0        ;ROTATING POINTER FOR ACTREG..POINTS
901                                                    ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DE
902                           ;*********************************************
903
904                           ; THESE ARE STORAGE FOR THE STATUS MAP PRINT OUT
905   001222      000000            FLAG:0       ; FLAGS FOR STATUS MAP PRINT OUT (SSP)
906   001224      000000            HOLD0: 0     ; HOLDS  R0 IN STATUS MAP PRINT
907   001226      000000            HOLD1:0      ; R1 ETC.
908   001230      000000            COUNT1:0     ; FOR COUNTING 3 WORDS
909   001232      000002            TABLE  : 2   ; FOR CONVRT ROUTINE
910   001234      003006                    3006
911   001236      000000                  0
912   001240      003006                    3006
913   001242      000000                      0
914                           ;*********************************
915                           ;PROGRAM CONTROL FLAGS
916
917
918   001244      000         INIFLG: .BYTE   0        ;PROGRAM INITIALIZATION FLAG
919   001245      000         STFLG:  .BYTE   0        ;TEST START FLAG
920   001246      000         ERRFLG: .BYTE   0        ;ERROR OCCURED FLAG
921   001247      000         LOKFLG: .BYTE   0        ;LOCK ON CURRENT TEST FLAG
922
923                           .EVEN
924
925                           ;*******    STATUS  MAP    *******
926
927          001300          .=1300
928   001300      000001      STATUS: NOSYNC: .BLKB 1   ;SYNC CHARS
929   001301      000001              MITSEX: .BLKB 1   ;XMIT JUMPER
930   001302      000001              RESEC:  .BLKB 1   ;REC SEC JUMPER
931   001303      000001              CLROPT: .BLKB 1   ;OPTIONAL JUMPER
932   001304      000001              DMULT:  .BLKB 1   ;MULTIPLE DEVICE FLAG
933   001305      000001              BYJMR:  .BLKB 1   ;EXTERNAL MODEM
934
935                           ; MULTIPLE    DEVICE PARAMETERS
936
937   001306      000001              ADDBASE: .BLKW 1   ;PROG CONTROLLED 1ST DEVICE ADDR
```

```
938   001310   000001        ADDKEEP: .BLKW 1        ;SAVED 1ST DEVICE ADDR
939   001312   000001        ADDLAST: .BLKW 1        ;LAST DEVICE  RXCSR  ADDR
940   001314   000001        IVBASE:  .BLKW 1        ;PROG CONTROLLED IV
941   001316   000001        IVKEEP:  .BLKW 1        ;SAVED INTR VECTOR
942   001320   000001        REGACT:  .BLKW 1        ;ACTIVE REGISTER
943   001322   000001        ADDROT:  .BLKW 1        ;ROTATING POINTER
944   001324   000001        PRTDU:   .BLKW 1        ;DU11 PRIORITY
945   001326   000001        RIVDU:   .BLKW 1        ;DU11 REC INTR VECTOR
946   001330   000001        TIVDU:   .BLKW 1        ;DU11 XMIT INTR VECTOR
947   001332   000001        TISDU:   .BLKW 1        ;DU11 XMIT INTR STATUS
948   001334   000001        RISDU:   .BLKW 1        ;DU11  REC INTR STATUS
949   001336   000001        L1ESS: .BLKW 1          ;PRIORITY TO ALLOW INTR
950   001340   000001        CSRRX:  .BLKW 1         ; DEFAULT OR ALTERED PARAMETERS
951   001342   000001        CSRHRX: .BLKW 1         ;
952   001344   000001        BUFRXD: .BLKW 1
953   001346   000001        BUFHRXD: .BLKW 1
954   001350   000001        CSRPAR: .BLKW  1
955   001352   000001        CSRHPAR:.BLKW  1
956   001354   000001        CSRTX: .BLKW 1
957   001356   000001        CSRHTX: .BLKW 1
958   001360   000001        BUFTXD: .BLKW 1
959   001362   000001        BUFHTXD:.BLKW 1
960   001364   000001        BASEDU: .BLKW 1         ;DU11 RXCSR BASE ADDR
961                          .EVEN
962
963                          ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
964                          ;POINTERS TO SUBROUTINES CAN BE FOUND
965                          ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
966
967   001366               .TRPTAB:
968                        ;***********************************************************************
969                        ;***********************************************************************
970           104400        SCOPE=TRAP+0     ;CALL TO SCOPE LOOP AND ITERATION HANDLER
971   001366  014366      .SCOPE
972           104401        SCOP1=TRAP+1     ;CALL TO LOOP ON CURRENT DATA HANDLER
973   001370  014552      .SCOP1
974           104402        TYPE=TRAP+2      ;CALL TO TELETYPE OUTPUT ROUTINE
975   001372  014572      .TYPE
976           104403        INSTR=TRAP+3     ;CALL TO ASCII STRING INPUT ROUTINE
977   001374  014632      .INSTR
978           104404        INSTER=TRAP+4    ;CALL TO INPUT ERROR HANDLER
979   001376  014750      .INSTER
980           104405        PARAM=TRAP+5     ;CALL TO NUMERICAL DATA INPUT ROUTINE
981   001400  015002      .PARAM
982           104406        SAV05=TRAP+6     ;CALL TO REGISTER SAVE ROUTINE
983   001402  015216      .SAV05
984           104407        RES05=TRAP+7     ;CALL TO REGISTER RESTORE ROUTINE
985   001404  015256      .RES05
986           104410        CONVRT=TRAP+10   ;CALL TO DATA OUTPUT ROUTINE
987   001406  015310      .CONVRT
988           104411        CNVRT=TRAP+11    ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF
989   001410  015314      .CNVRT
990           104412        SETFLG=TRAP+12   ;CALL TO FLAG SET ROUTINE
991   001412  015534      .SETFLG
992           104413        CKSWR=TRAP+13    ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
993   001414  016250      .CKSWR
```

```
 994            104414                           CNTLU=TRAP+14    ;CALL TO ALLOW LOADING  OF SWREG FROM TTY
 995   001416  016324                   .CNTLU
 996                                     ;**********************************************************************
 997                                     ;**********************************************************************
 998
 999                                             ;PROGRAM INITIALIZATION
1000                                             ;LOCK OUT INTERRUPTS
1001                                             ;SET UP PROCESSOR STACK
1002                                             ;SET UP POWER FAIL VECTOR
1003                                             ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1004                                             ;TYPE TITLE MESSAGE
1005
1006   001420  012767  000340  176350   .START:  MOV     #340,PS          ;LOCK OUT INTERRUPTS
1007   001426  012706  001100                     MOV     #STACK,SP        ;SET UP STACK
1008   001432  012737  016104  000024             MOV     #.PFAIL,@#24     ;SET UP POWER FAIL VECTOR
1009   001440  005067  177460                     CLR     LPCNT            ;CLEAR # OF ITERATION COMPLETED LOCATION
1010   001444  105067  177575                     CLRB    STFLG            ;CLEAR START FLAG
1011   001450  005067  177454                     CLR     PASCNT           ;CLEAR PASS COUNT
1012   001454  105067  177566                     CLRB    ERRFLG           ;CLEAR ERROR FLAG
1013   001460  005067  177446                     CLR     ERRCNT           ;CLEAR ERROR COUNT
1014   001464  005067  177444                     CLR     LSTERR           ;CLEAR LAST ERROR POINTER
1015   001470  012767  000001  177430             MOV     #1,TSTNO         ;SET UP FOR TEST 1
1016   001476  012767  001420  177410             MOV     #.START,RTRN     ;SET UP FOR POWER FAIL BEFORE
1017                                                                       ;TESTING STARTS
1018   001504  105767  177534                     TSTB    INIFLG           ;HAS INITIALIZATION BEEN PERFORMED
1019   001510  001004                             BNE     ONCE
1020   001512  104402  016424                     TYPE    .MTITLE          ;TYPE TITLE MESSAGE
1021   001516  105167  177522                     COMB    INIFLG           ;IF NOT SET FLAG AND DO
1022   001522  012767  177570  177350   ONCE:     MOV     #DSWR,SWR        ;RELOAD HARDWARE SWITCH REGISTER INTO POINTER
1023   001530  012767  177570  177344             MOV     #DLIGHTS,LIGHTS  ;RELOAD HARDWARE DISPLAY REGISTER INTO POINTER
1024   001536  013746  000006                     MOV     @#6,-(SP)        ;SAVE VECTORS
1025   001542  013746  000004                     MOV     @#4,-(SP)
1026   001546  012737  001566  000004             MOV     #64$,@#4         ;SET UP FOR TIMEOUT
1027   001554  022777  177777  177316             CMP     #-1,@SWR         ;REFERENCE HARDWARE SWITCH REGISTER
1028   001562  001402                             BEQ     65$
1029   001564  000407                             BR      66$
1030   001566  022626                   64$:      CMP     (SP)+,(SP)+      ;ADJUST STACK
1031   001570  012767  000176  177302   65$:      MOV     #SWREG,SWR       ;POINT TO SOFTWARE SWITCH REG
1032   001576  012767  000174  177276             MOV     #DISPREG,LIGHTS  ;POINT TO SOFT DISPLAY REG
1033   001604  012637  000004           66$:      MOV     (SP)+,@#4        ;RESTORE VECTORS
1034   001610  012637  000006                     MOV     (SP)+,@#6
1035   001614  005737  000042                     TST     @#42             ;UNDER MONITOR
1036   001620  001005                             BNE     MAP
1037   001622  022767  000176  177250             CMP     #SWREG,SWR       ;IS SWREG USED
1038   001630  001001                             BNE     MAP              ; BRANCH TO CHECK FOR STATUS MAP
1039   001632  104414                             CNTLU
1040                                     ;************************************************************
1041                                     ; CODE FOR STATUS MAP
1042                                     ; CODE ADDED FOR REV. E OF DIAGNOSTICS
1043                                     ; IF SW07= 1 ,THEN YOU USE THE STATUS MAP PREVIOUSLY
1044                                     ; SETUP, OR REENTER QUESTIONING ROUTINE
1045
1046   001634  032777  000200  177236   MAP:      BIT     #SW07, @SWR      ; IS SW07=1?
1047   001642  001537                             BEQ     $67              ; IF NOT, GO TO TEST FOR SW00=1
1048                                     ; NOW SET UP MAP VALUES FOR PROGRAM
1049                                     ; THESE VALUES FROM THE STATUS MAP WILL BE USED IN THE
```

```
1050                                          ; OPERATION OF THIS  PROGRAM.
1051  001644  116767  177430  177324          MOVB NOSYNC       ,SYNCNO          ; SYNC CHAR
1052  001652  116767  177423  177317          MOVB MITSEX       ,SEXMIT          ; XMIT   JUMPER
1053  001660  116767  177416  177312          MOVB RESEC        ,SEREC           ; SEC REC JUMPER
1054  001666  116767  177411  177305          MOVB CLROPT       ,OPTCLR          ;OPTIONAL JUMPER
1055  001674  116767  177404  177300          MOVB DMULT        ,MULTD           ;MULTIPLE DEVICE
1056  001702  116767  177377  177273          MOVB BYJMR        ,JMRBY           ;EXTERNAL MODEM
1057  001710  016767  177372  177266          MOV  ADDBASE      ,BASEADD         ;PROG 1ST DEVICE ADDR
1058  001716  016767  177366  177262          MOV  ADDKEEP      ,KEEPADD         ; SAVED 1ST DEVICE ADDR
1059  001724  016767  177362  177256          MOV   ADDLAST     ,LASTADD         ; LAST DEVICE RXCSR ADDR
1060  001732  016767  177360  177254          MOV  IVKEEP       ,KEEPIV          ; SAVED INTR VECTOR
1061  001740  016767  177354  177250          MOV  REGACT       ,ACTREG          ;ACTIVE REGISTER
1062  001746  016767  177350  177244          MOV  ADDROT       ,ROTADD          ;ROTATING POINTER
1063  001754  016767  177334  177230          MOV  IVBASE,      BASEIV           ; BASE INTR VECTOR
1064  001762  016767  177220  177214          MOV  KEEPADD      , BASEADD        :RELOAD BASEADD
1065  001770  016767  177332  016572          MOV  RIVDU,       DURIV            ; REC INTR VECTOR
1066  001776  016767  177332  016566          MOV  RISDU,       DURIS            ; REC INTR STATUS
1067  002004  016767  177320  016562          MOV  TIVDU,       DUTIV            ; XMIT INTR VECTOR
1068  002012  016767  177314  016556          MOV  TISDU,       DUTIS            ; XMIT INTR STATUS
1069  002020  016767  177312  016064          MOV  L1ESS,       LESS1            ; PRIORITY TO ALLOW INTR
1070  002026  013737  001324  020110          MOV  a#PRTDU,     a#DUPRT          ; PRIORITY RELOADED
1071  002034  016767  177324  016204          MOV       BASEDU, DUBASE           ;'
1072  002042  016767  177272  016474          MOV  CSRRX,       RXCSR
1073  002050  016767  177266  016470          MOV  CSRHRX,      HRXCSR
1074  002056  016767  177262  016464          MOV       BUFRXD, RXDBUF
1075  002064  016767  177256  016460          MOV       BUFHRXD,          HRXDBUF
1076  002072  016767  177252  016454          MOV       CSRPAR,           PARCSR
1077  002100  016767  177246  016450          MOV       CSRHPAR,          HPARCSR
1078  002106  016767  177242  016444          MOV       CSRTX,  TXCSR
1079  002114  016767  177236  016440          MOV       CSRHTX,           HTXCSR
1080  002122  016767  177232  016434          MOV  BUFTXD,      TXDBUF
1081  002130  016767  177226  016430          MOV  BUFHTXD,     HTXDBUF
1082  002136  000167  000466                  JMP .BEGIN                         ;BRANCH TO BEGIN TESTING
1083                                  :**************************************************
1084  002142  032777  000001  176730  $67:    BIT     #SW00     ,aSWR            ; RESELECT VECTOR $ CONTROL REG?
1085  002150  001002                          BNE     1$                         ; BRANCH  TO QUESTIONING
1086  002152  000167  000452                  JMP .BEGIN                         ; GO TO  LOAD STATUS MAP ETC.
1087  002156  005037  001222          1$:     CLR a#FLAG                         ; CLEAR FLAG SO STATUS MAP PRINTS OUT
1088  002162  012700  000300                  MOV     #300,R0                    ;RESTORE VECTOR AREA TO TRAPCATCHER
1089  002166  012701  000302                  MOV     #302,R1                    ;START AT LOCATION 300
1090  002172  012702  000004                  MOV     #4,R2
1091  002176  010110          2$:             MOV     R1,(R0)
1092  002200  005011                          CLR     (R1)
1093  002202  060200                          ADD     R2,R0
1094  002204  060201                          ADD     R2,R1
1095  002206  022701  001000                  CMP     #1000,R1                   ;END AT LOCATION 776
1096  002212  002771                          BLT     2$
1097  002214  104403                          INSTR                              ;OUTPUT MESSAGE & GET INPUT STRING
1098  002216  016500                          MREGAD                             ;MESSAGE
1099  002220  104405                          PARAM                              ;CONVERT STRING
1100  002222  160000                          160000                             ;LOW LIMIT
1101  002224  167776                          167776                             ;HIGH LIMIT
1102  002226  020246                          DUBASE                             ;STORE AT THIS LOCATION
1103  002230  001             .BYTE   1                                          ;MASK
1104  002231  001             .BYTE   1                                          ;HOW MANY TIMES + 2
1105  002232  016767  016010  176746          MOV       DUBASE,KEEPADD  ;SAVE
```

```
1106  002240  004767  015650                   JSR     PC,DUADDR
1107  002244  016767  176736  176732           MOV     KEEPADD,BASEADD  ;RESTORE FOR ROTATION
1108  002252  104403                            INSTR                    ;OUTPUT MESSAGE & GET INPUT STRING
1109  002254  016456                             MVECTO                  ;MESSAGE
1110  002256  104405                             PARAM                   ;CONVERT STRING
1111  002260  000300                             300                     ;LOW LIMIT
1112  002262  000776                             776                     ;HIGH LIMIT
1113  002264  020570                             DURIV                   ;STORE AT THIS LOCATION
1114  002266     001              .BYTE   1                              ;MASK
1115  002267     004              .BYTE   4                              ;HOW MANY TIMES + 2
1116  002270  016767  016274  176716           MOV     DURIV,KEEPIV     ;SAVE
1117  002276  016767  016266  176706           MOV     DURIV,BASEIV     ;SET UP FOR ROTATION
1118  002304  104403                            INSTR                    ;OUTPUT MESSAGE & GET INPUT STRING
1119  002306  016561                             MMULT                   ;MESSAGE
1120  002310  104412                             SETFLG                  ;SET FLAG BASED UPON INPUT STRING
1121  002312  001202                             MULTD                   ;THIS FLAG
1122  002314  105767  176662                   TSTB    MULTD   ;ARE THERE MULTIPLE DEVICES
1123                                                    ;ON THE SYSTEM ?
1124  002320  100406                           BMI     BBB     ;YES,ASK NEXT QUESTION
1125  002322  005067  176670                   CLR     ACTREG
1126  002326  005067  176666                   CLR     ROTADD
1127  002332  000167  000140                   JMP     OUTMUL  ;JUMP AROUND NEXT QUESTION
1128  002336                           BBB:
1129  002336  104403                            INSTR                    ;OUTPUT MESSAGE & GET INPUT STRING
1130  002340  016640                             MLASTD                  ;MESSAGE
1131  002342  104405                             PARAM                   ;CONVERT STRING
1132  002344  160000                             160000                  ;LOW LIMIT
1133  002346  167776                             167776                  ;HIGH LIMIT
1134  002350  001210                             LASTADD                 ;STORE AT THIS LOCATION
1135  002352     001              .BYTE   1                              ;MASK
1136  002353     001              .BYTE   1                              ;HOW MANY TIMES + 2
1137                                  ;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1138  002354  012767  000001  176636  1$:      MOV     #1,ROTADD        ;SET UP POINTER
1139  002362  005067  176630                   CLR     ACTREG  ;CLR ACTIVE REGISTER
1140  002366  056767  176626  176622  2$:      BIS     ROTADD,ACTREG    ;MAKE THIS DEVICE ACTIVE
1141  002374  000241                           CLC
1142  002376  006167  176616                   ROL     ROTADD  ;SET UP POINTER
1143  002402  103421                           BCS     3$      ;ARE YOU OUT OF RANGE ?
1144  002404  062767  000010  176572           ADD     #10,BASEADD     ;SET UP BASE ADDRESS
1145  002412  026767  176572  176564           CMP     LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
1146  002420  101362                           BHI     2$      ;NO DO IT AGAIN
1147  002422  056767  176572  176566           BIS     ROTADD,ACTREG   ;THIS ASSUMES THAT THERE ARE AT
1148                                                    ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1149                                                    ;MULTIPLE DEVICE QUESTION
1150  002430  012767  000001  176562  4$:      MOV     #1,ROTADD        ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1151  002436  016767  176544  176540           MOV     KEEPADD,BASEADD ;DITTO
1152  002444  000414                           BR      OUTMUL  ;CONTINUE QUESTIONS
1153  002446  016767  176534  176530  3$:      MOV     KEEPADD,BASEADD ;RESTORE
1154  002454  104403                            INSTR                    ;OUTPUT MESSAGE & GET INPUT STRING
1155  002456  017023                             MRANGE                  ;MESSAGE
1156  002460  104405                             PARAM                   ;CONVERT STRING
1157  002462  160000                             160000                  ;LOW LIMIT
1158  002464  167776                             167776                  ;HIGH LIMIT
1159  002466  001210                             LASTADD                 ;STORE AT THIS LOCATION
1160  002470     001              .BYTE   1                              ;MASK
1161  002471     001              .BYTE   1                              ;HOW MANY TIMES + 2
```

```
1162  002472  000167  177656              JMP    1$        ;DO IT AGAIN
1163  002476                     OUTMUL:
1164  002476  104403                       INSTR                    ;OUTPUT MESSAGE & GET INPUT STRING
1165  002500  017307                       MLEVEL                   ;MESSAGE
1166  002502  104405                       PARAM                    ;CONVERT STRING
1167  002504  000004                       4                        ;LOW LIMIT
1168  002506  000007                       7                        ;HIGH LIMIT
1169  002510  020110                       DUPRT                    ;STORE AT THIS LOCATION
1170  002512  000         .BYTE   0                        ;MASK
1171  002513  001         .BYTE   1                        ;HOW MANY TIMES + 2
1172  002514  004767  015320              JSR    PC,DULEV
1173                           ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1174                           ;BUFFER TO THE CHARACTERS '1' AND '2'.
1175                           ;IF THE CHARACTER IS '1' CLEAR THE FLAG
1176                           ;IF THE CHARACTER IS '2' SET THE FLAG
1177  002520                     AAA:
1178  002520  104403                       INSTR                    ;OUTPUT MESSAGE & GET INPUT STRING
1179  002522  017334                       MSYNC                    ;MESSAGE
1180  002524  122767  000061  015146  3$:  CMPB   #'1,INBUF         ;IS IT '1' ?
1181  002532  001003                       BNE    1$
1182  002534  105067  176436              CLRB   SYNCNO            ;000
1183  002540  000412                       BR     4$
1184  002542  122767  000062  015130  1$:  CMPB   #'2,INBUF         ;IS IT '2' ?
1185  002550  001004                       BNE    2$
1186  002552  112767  177777  176416      MOVB   #-1,SYNCNO                ;377
1187  002560  000402                       BR     4$
1188  002562  104404                  2$:  INSTER                   ;RETRY
1189  002564  000757                       BR     3$
1190  002566  000240                  4$:  NOP
1191  002570  104403                       INSTR                    ;OUTPUT MESSAGE & GET INPUT STRING
1192  002572  017402                       MWIRE6                   ;MESSAGE
1193  002574  104412                       SETFLG                   ;SET FLAG BASED UPON INPUT STRING
1194  002576  001177                       SEXMIT                   ;THIS FLAG
1195  002600  104403                       INSTR                    ;OUTPUT MESSAGE & GET INPUT STRING
1196  002602  017450                       MWIRE5                   ;MESSAGE
1197  002604  104412                       SETFLG                   ;SET FLAG BASED UPON INPUT STRING
1198  002606  001200                       SEREC                    ;THIS FLAG
1199  002610  104403                       INSTR                    ;OUTPUT MESSAGE & GET INPUT STRING
1200  002612  017515                       MWIRE4                   ;MESSAGE
1201  002614  104412                       SETFLG                   ;SET FLAG BASED UPON INPUT STRING
1202  002616  001201                       OPTCLR                   ;THIS FLAG
1203  002620  104403                       INSTR                    ;OUTPUT MESSAGE & GET INPUT STRING
1204  002622  017571                       MEXTJ                    ;MESSAGE
1205  002624  104412                       SETFLG                   ;SET FLAG BASED UPON INPUT STRING
1206  002626  001203                       JMRBY                    ;THIS FLAG
1207
1208
1209                           ;TEST START AND RESTART
1210
1211  002630  012767  000340  175140  .BEGIN: MOV   #340,PS                   ;LOCK OUT INTERRUPTS
1212                           ; ******  LOAD STATUS MAP ******
1213                           ;THE VALUES NOW BEING LOADED INTO THE STATUS MAP WILL BE
1214                           ; USED IN THIS PROGRAM AND    WILL BE  PASSED  TO ANY
1215                           ;OTHER DU11 PROGRAMS LOADED IMMEDIATELY FOLLOWING THIS PROG.
1216  002636  032777  000200  176234      BIT    #SW07        .aSWR  ; SW07 SET , IF YES BRANCH
1217  002644  001132                       BNE    HEREU
```

```
1218  002646  116767  176324  176424        MOVB SYNCNO   ,NOSYNC      ;SYNC CHARS
1219  002654  116767  176317  176417        MOVB SEXMIT   ,MITSEX      ;XMIT JUMPER
1220  002662  116767  176312  176412        MOVB SEREC    ,RESEC       ;SEC REC JUMPER
1221  002670  116767  176305  176405        MOVB OPTCLR   ,CLROPT      ;OPTIONAL JUMPER
1222  002676  116767  176300  176400        MOVB MULTD    ,DMULT       ; MULTIPLE DEVICES
1223  002704  116767  176273  176373        MOVB JMRBY    ,BYJMR       ;EXTERNAL MODEM
1224  002712  016767  176266  176366        MOV BASEADD   ,ADDBASE     ;PROG CONTROLLED 1ST ADDR
1225  002720  016767  176262  176362        MOV KEEPADD   ,ADDKEEP     ;SAVED 1ST DEVICE ADDR
1226  002726  016767  176256  176356        MOV LASTADD   ,ADDLAST     ;LAST DEVICE RXCSR ADDR
1227  002734  016767  176254  176354        MOV KEEPIV    ,IVKEEP      ;SAVED INTR VECTOR
1228  002742  016767  176244  176344        MOV BASEIV    ,IVBASE      ; RELOAD BASE INTR VECTOR
1229  002750  016767  176242  176342        MOV ACTREG    ,REGACT      ;ACTIVE REGISTER
1230  002756  016767  176236  176336        MOV ROTADD    ,ADDROT      ;ROTATING POINTER
1231  002764  013737  020110  001324        MOV @#DUPRT   ,@#PRTDU             ;DU11 PRIORITY
1232  002772  015572  176326             MOV DURIV     ,RIVDU       ;REC INTR VECTOR
1233  003000  016767  015566  176326        MOV DURIS     ,RISDU       ; REC INTR STATUS
1234  003006  016767  015562  176314        MOV DUTIV     ,TIVDU       ; XMIT INTR    VECTOR
1235  003014  016767  015556  176310        MOV DUTIS     ,TISDU       ; XMIT  INTR   STATUS
1236  003022  016767  015064  176306        MOV LESS1     ,L1ESS       ; PRIORITY TO ALLOW INTR
1237  003030  016767  015212  176326        MOV DUBASE    ,BASEDU      ;RXCSR BASE ADDRESS
1238  003036  016767  015502  176274        MOV RXCSR,    CSRRX
1239  003044  016767  015476  176270        MOV HRXCSR,   CSRHRX
1240  003052  016767  015472  176264        MOV RXDBUF,   BUFRXD
1241  003060  016767  015466  176260        MOV HRXDBUF,  BUFHRXD
1242  003066  016767  015462  176254        MOV PARCSR,   CSRPAR
1243  003074  016767  015456  176250        MOV HPARCSR,  CSRHPAR
1244  003102  016767  015452  176244        MOV TXCSR,    CSRTX
1245  003110  016767  015446  176240        MOV HTXCSR,   CSRHTX
1246  003116  016767  015442  176234        MOV TXDBUF,   BUFTXD
1247  003124  016767  015436  176230        MOV HTXDBUF,  BUFHTXD
1248                                      ;***************************************
1249                                      ;  THE FOLLOWING CODE WILL PRINT
1250                                      ;  THE CONVERSATIONALLY SET JUMPER
1251                                      ;  SETTINGS FROM THE STATUS MAP
1252                                      ;  ON THE FIRST PASS OF
1253                                      ;  THIS DIAGNOSTIC OR
1254                                      ;  JUST AFTER THE QUESTIONING
1255                                      ;  AND ANSWERING .
1256                                      ;***************************************
1257
1258  003132  005737  001222        HEREU:        TST @#FLAG          ; TEST IF 1ST PASS
1259  003136  001402                               BEQ    SETFG        ; IF FIRST PASS SET FLAG/PRINT
1260  003140  000167  000116                       JMP THRU            ; AROUND IF PASS > 1
1261  003144  104402  017646        SETFG:        TYPE   ,MSTATUS      ; PRINT 'STATUS MAP'
1262  003150  062737  000001  001222              ADD #1, @#FLAG       ; SET FLAG ON 1ST PASS
1263  003156  010067  176042                       MOV RO, HOLD0       ; SAVE RO
1264  003162  010167  176040                       MOV R1, HOLD1       ; SAVE R1
1265  003166  012767  000003  176034              MOV #3, COUNT1       ;COUNTER FOR WORDS PRINTED
1266  003174  012700  000002'                      MOV #BUFF1, RO
1267  003200  012701  001300                       MOV #STATUS, R1     ; (BUFF1)=STATUS  ETC.
1268  003204  010120                FILBUF:  MOV R1, (RO)+            ; LOAD BUFF AS ABOVE
1269  003206  062701  000002                       ADD #2 ,R1          ; PREPARE STATUS ADDRESS
1270  003212  020127  001306                       CMP R1, #STATUS+6   ; CHECK IF 3 WORDS LOADED
1271  003216  001372                               BNE FILBUF          ; BACK  TO LOAD NEXT ADDRESS
1272  003220  012700  000002'                      MOV #BUFF1 ,RO      ; LOAD FOR PRINT OUT
1273  003224  010067  176006        UP:      MOV RO , TABLE+4         ;LOAD ADDRESS TO PRINT
```

```
1274  003230  012067  176006              MOV    (R0)+, TABLE+10         ; LOAD CONTENTS
1275  003234  104410  001232              CONVRT , TABLE          ; PRINT ADDRESS/CONTENTS PAIR
1276  003240  104402  017110              TYPE, MCRLF             ; CR AND LF
1277  003244  005367  175760              DEC COUNT1              ;COUNT WORDS PRINTED
1278  003250  001365                      BNE UP                  ; GO PRINT NEXT ADDRESS/CONTENTS
1279  003252  016700  175746              MOV HOLD0,R0
1280  003256  016701  175744              MOV HOLD1, R1
1281                                       ;*********************************
1282
1283  003262  012706  001100      THRU:   MOV    #STACK,SP        ;SET UP STACK
1284  003266  005737  000042              TST    @#42             ;IS PROGRAM UNDER MONITOR CONTROL
1285  003272  001056                      BNE    3$
1286  003274  105767  175702              TSTB   MULTD   ;DON'T ALLOW LOCK ON TEST IF RUNNING
1287                                       ;MULTIPLE DEVICES
1288  003300  001407                      BEQ    5$               ;IF NO,TEST FOR LOCK ON TEST
1289  003302  016767  011240  011140      MOV    BRW,TTST         ;RESTORE NORMAL SCOPE LOOP
1290  003310  016767  011234  011134      MOV    BRX,TTST+2       ;DITTO
1291  003316  000444                      BR     3$               ;JUMP AROUND IF YES
1292  003320  032777  000004  175552  5$: BIT    #BIT2,@SWR       ;CHECK FOR LOCK ON TEST
1293  003326  001416                      BEQ    1$
1294  003330  104403                      INSTR                   ;OUTPUT MESSAGE & GET INPUT STRING
1295  003332  017244                      MLOCK                   ;MESSAGE
1296  003334  104412                      SETFLG                  ;SET FLAG BASED UPON INPUT STRING
1297  003336  001247                      LOKFLG                  ;THIS FLAG
1298  003340  105767  175703              TSTB   LOKFLG           ;IS LOCK ON TEST OPTION SELECTED
1299  003344  001407                      BEQ    1$
1300  003346  012767  000240  011074      MOV    #NOP,TTST
1301  003354  012767  000240  011070      MOV    #NOP,TTST+2      ;SET UP TO LOCK
1302  003362  000406                      BR     2$
1303  003364  016767  011156  011056  1$: MOV    BRW,TTST
1304  003372  016767  011152  011052      MOV    BRX,TTST+2       ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1305  003400  032777  000002  175472  2$: BIT    #SW01,@SWR       ;IF SW01=1, GET STARTING PC
1306  003406  001410                      BEQ    3$
1307  003410  104403                      INSTR                   ;OUTPUT MESSAGE & GET INPUT STRING
1308  003412  017231                      MTSTPC                  ;MESSAGE
1309  003414  104405                      PARAM                   ;CONVERT STRING
1310  003416  003446                      TST1                    ;LOW LIMIT
1311  003420  013610                      TLAST                   ;HIGH LIMIT
1312  003422  001114                      RTRN                    ;STORE AT THIS LOCATION
1313  003424    001               .BYTE   1                       ;MASK
1314  003425    001               .BYTE   1                       ;HOW MANY TIMES + 2
1315  003426  000403                      BR     4$
1316  003430  012767  003446  175456  3$: MOV    #TST1,RTRN       ;START AT TEST 1
1317  003436  104402  017225          4$: TYPE   ,MR              ;TYPE R
1318  003442  000177  175446              JMP    @RTRN            ;START TESTING
1319
1320                                       ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1321                                       ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1322                                       ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1323                                       ;;(OVRRUN,RXERR)
1324                                       ;;MODE:ISYMOD
1325                                       ;;LENGTH:SIX
1326                                       ;;CHAR:25
1327                                       ;;
1328  003446  012767  000001  175452  TST1: MOV  #1,TSTNO         ;SAVE THIS
1329  003454  012767  003710  175434      MOV    #TST2,NEXT       ;GO TO THIS TEST WHEN THRU
```

```
1330   003462   052777   000400   015070        BIS      #MRESET,aTXCSR  ;MASTER RESET
1331   003470   012777   000000   015056        MOV      #ISYMOD,aPARCSR ;SET THE MODE
1332   003476   052777   000400   015054        BIS      #MRESET,aTXCSR  ;MASTER RESET
1333
1334                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1335   003504   012777   064001   015046        MOV      #MTDATA!CLK!MINT!BREAK,aTXCSR
1336
1337                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1338   003512   012777   002000   015034        MOV      #ISYMOD!SIX!NOPAR!0,aPARCSR
1339   003520   052777   000020   015016        BIS      #SYNSCH,aRXCSR  ;SET SYNC SEARCH
1340                                    ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1341   003526   042777   020000   015024        BIC      #CLK,aTXCSR     ;POKE CLK DOWN
1342   003534   052777   020000   015016        BIS      #CLK,aTXCSR     ;POKE CLK UP
1343                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1344   003542   042777   020000   015010        BIC      #CLK,aTXCSR     ;POKE CLK DOWN
1345   003550   052777   020000   015002        BIS      #CLK,aTXCSR     ;POKE CLK UP
1346   003556   016703   014766                 MOV      RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1347   003562   012700   000025                 MOV      #25,R0   ;EXPECTED
1348   003566   012767   000010   175344        MOV      #8.,SHIFT       ;# OF SHIFTS
1349   003574   012767   000252   175342        MOV      #252,TEMP1      ;DATA CHAR
1350   003602   004767   014442                 JSR      PC,RPOKE        ;SHIFT IN THIS CHAR
1351   003606   105777   014732                 TSTB     aRXCSR   ;RXDONE ?
1352   003612   100401                          BMI      64$
1353   003614   104000                          HLT      ;RXDONE SHOULD BE SET
1354   003616
1355   003616   017701   014726        64$:     MOV      aRXDBUF,R1      ;ACTUAL
1356   003622   020001                          CMP      R0,R1    ;COMPARE EXPECTED VS. ACTUAL
1357   003624   001401                          BEQ      65$
1358   003626   104002                          HLT      2        ;RECEIVED DATA DID NOT MATCH
1359                                                               ;EXPECTED DATA - CHECK MAINT DATA
1360                                                               ;OR RECEIVER LOGIC
1361   003630                          65$:
1362   003630   012767   000010   175302        MOV      #8.,SHIFT       ;# OF SHIFTS
1363   003636   012767   000252   175300        MOV      #252,TEMP1      ;DATA CHAR
1364   003644   004767   014400                 JSR      PC,RPOKE        ;SHIFT IN THIS CHAR
1365                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1366   003650   012767   000010   175262        MOV      #8.,SHIFT       ;# OF SHIFTS
1367   003656   012767   000252   175260        MOV      #252,TEMP1      ;DATA CHAR
1368   003664   004767   014360                 JSR      PC,RPOKE        ;SHIFT IN THIS CHAR
1369   003670   012700   140025                 MOV      #140000!25,R0   ;EXPECTED DATA PLUS
1370                                                               ;RXERR & OVRRUN
1371   003674   017701   014650                 MOV      aRXDBUF,R1      ;ACTUAL
1372   003700   020001                          CMP      R0,R1    ;COMPARE EXP VS. ACT
1373   003702   001401                          BEQ      66$
1374   003704   104002                          HLT      2        ;SPECIFICALLY LOOK AT RXERR &
1375                                                               ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1376   003706                          66$:
1377   003706   104400                          SCOPE
1378                                    ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1379                                    ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1380                                    ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1381                                    ;;(OVRRUN,RXERR)
1382                                    ;;MODE:ISYMOD
1383                                    ;;LENGTH:SIX
1384                                    ;;CHAR:52
1385                                    ;;
```

```
1386  003710  012767  000002  175210  TST2:   MOV     #2,TSTNO          :SAVE THIS
1387  003716  012767  004152  175172          MOV     #TST3,NEXT              :GO TO THIS TEST WHEN THRU
1388  003724  052777  000400  014626          BIS     #MRESET,aTXCSR  :MASTER RESET
1389  003732  012777  000000  014614          MOV     #ISYMOD,aPARCSR :SET THE MODE
1390  003740  052777  000400  014612          BIS     #MRESET,aTXCSR  :MASTER RESET
1391
1392                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1393  003746  012777  064001  014604          MOV     #MTDATA!CLK!MINT!BREAK,aTXCSR
1394
1395                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1396  003754  012777  002000  014572          MOV     #ISYMOD!SIX!NOPAR!0,aPARCSR
1397  003762  052777  000020  014554          BIS     #SYNSCH,aRXCSR  :SET SYNC SEARCH
1398                                          :POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1399  003770  042777  020000  014562          BIC     #CLK,aTXCSR       :POKE CLK DOWN
1400  003776  052777  020000  014554          BIS     #CLK,aTXCSR       :POKE CLK UP
1401                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1402  004004  042777  020000  014546          BIC     #CLK,aTXCSR       :POKE CLK DOWN
1403  004012  052777  020000  014540          BIS     #CLK,aTXCSR       :POKE CLK UP
1404  004020  016703  014524                  MOV     RXDBUF,R3         :SET UP FOR ERROR MESSAGE
1405  004024  012700  000052                  MOV     #52,R0  :EXPECTED
1406  004030  012767  000010  175102          MOV     #8.,SHIFT         :# OF SHIFTS
1407  004036  012767  000324  175100          MOV     #324,TEMP1        :DATA CHAR
1408  004044  004767  014200                  JSR     PC,RPOKE          :SHIFT IN THIS CHAR
1409  004050  105777  014470                  TSTB    aRXCSR  :RXDONE ?
1410  004054  100401                          BMI     64$
1411  004056  104000                          HLT     :RXDONE SHOULD BE SET
1412  004060                          64$:
1413  004060  017701  014464                  MOV     aRXDBUF,R1        :ACTUAL
1414  004064  020001                          CMP     R0,R1   :COMPARE EXPECTED VS. ACTUAL
1415  004066  001401                          BEQ     65$
1416  004070  104002                          HLT     2                 :RECEIVED DATA DID NOT MATCH
1417                                                                    :EXPECTED DATA - CHECK MAINT DATA
1418                                                                    :OR RECEIVER LOGIC
1419  004072                          65$:
1420  004072  012767  000010  175040          MOV     #8.,SHIFT         :# OF SHIFTS
1421  004100  012767  000324  175036          MOV     #324,TEMP1        :DATA CHAR
1422  004106  004767  014136                  JSR     PC,RPOKE          :SHIFT IN THIS CHAR
1423                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1424  004112  012767  000010  175020          MOV     #8.,SHIFT         :# OF SHIFTS
1425  004120  012767  000324  175016          MOV     #324,TEMP1        :DATA CHAR
1426  004126  004767  014116                  JSR     PC,RPOKE          :SHIFT IN THIS CHAR
1427  004132  012700  140052                  MOV     #140000!52,R0     :EXPECTED DATA PLUS
1428                                                                    :RXERR & OVRRUN
1429  004136  017701  014406                  MOV     aRXDBUF,R1        :ACTUAL
1430  004142  020001                          CMP     R0,R1   :COMPARE EXP VS. ACT
1431  004144  001401                          BEQ     66$
1432  004146  104002                          HLT     2                 :SPECIFICALLY LOOK AT RXERR &
1433                                                                    :OVRRUN BITS...THEY BOTH SHOULD BE SET
1434  004150                          66$:
1435  004150  104400                          SCOPE
1436                                          ::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1437                                          ::RECEIVER SECTION,IT USES THE ERROR FLAGS
1438                                          ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1439                                          ::(OVRRUN,RXERR)
1440                                          ::MODE:ISYMOD
1441                                          ::LENGTH:SIX
```

```
1442                                              ;;CHAR:77
1443                                              ;;
1444   004152  012767  000003  174746    TST3:    MOV     #3,TSTNO          ;SAVE THIS
1445   004160  012767  004414  174730             MOV     #TST4,NEXT              ;GO TO THIS TEST WHEN THRU
1446   004166  052777  000400  014364             BIS     #MRESET,@TXCSR  ;MASTER RESET
1447   004174  012777  000000  014352             MOV     #ISYMOD,@PARCSR ;SET THE MODE
1448   004202  052777  000400  014350             BIS     #MRESET,@TXCSR  ;MASTER RESET
1449
1450                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1451   004210  012777  064001  014342             MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1452
1453                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1454   004216  012777  002000  014330             MOV     #ISYMOD!SIX!NOPAR!0,@PARCSR
1455   004224  052777  000020  014312             BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1456                                              ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1457   004232  042777  020000  014320             BIC     #CLK,@TXCSR       ;POKE CLK DOWN
1458   004240  052777  020000  014312             BIS     #CLK,@TXCSR       ;POKE CLK UP
1459                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1460   004246  042777  020000  014304             BIC     #CLK,@TXCSR       ;POKE CLK DOWN
1461   004254  052777  020000  014276             BIS     #CLK,@TXCSR       ;POKE CLK UP
1462   004262  016703  014262                      MOV     RXDBUF,R3         ;SET UP FOR ERROR MESSAGE
1463   004266  012700  000077                      MOV     #77,R0  ;EXPECTED
1464   004272  012767  000010  174640             MOV     #8.,SHIFT         ;# OF SHIFTS
1465   004300  012767  000376  174636             MOV     #376,TEMP1        ;DATA CHAR
1466   004306  004767  013736                      JSR     PC,RPOKE          ;SHIFT IN THIS CHAR
1467   004312  105777  014226                      TSTB    @RXCSR  ;RXDONE ?
1468   004316  100401                              BMI     64$
1469   004320  104000                              HLT     ;RXDONE SHOULD BE SET
1470   004322                           64$:
1471   004322  017701  014222                      MOV     @RXDBUF,R1        ;ACTUAL
1472   004326  020001                              CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1473   004330  001401                              BEQ     65$
1474   004332  104002                              HLT     2         ;RECEIVED DATA DID NOT MATCH
1475                                                                 ;EXPECTED DATA - CHECK MAINT DATA
1476                                                                 ;OR RECEIVER LOGIC
1477   004334                           65$:
1478   004334  012767  000010  174576             MOV     #8.,SHIFT         ;# OF SHIFTS
1479   004342  012767  000376  174574             MOV     #376,TEMP1        ;DATA CHAR
1480   004350  004767  013674                      JSR     PC,RPOKE          ;SHIFT IN THIS CHAR
1481                                     ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1482   004354  012767  000010  174556             MOV     #8.,SHIFT         ;# OF SHIFTS
1483   004362  012767  000376  174554             MOV     #376,TEMP1        ;DATA CHAR
1484   004370  004767  013654                      JSR     PC,RPOKE          ;SHIFT IN THIS CHAR
1485   004374  012700  140077                      MOV     #140000!77,R0     ;EXPECTED DATA PLUS
1486                                                                 ;RXERR & OVRRUN
1487   004400  017701  014144                      MOV     @RXDBUF,R1        ;ACTUAL
1488   004404  020001                              CMP     R0,R1   ;COMPARE EXP VS. ACT
1489   004406  001401                              BEQ     66$
1490   004410  104002                              HLT     2         ;SPECIFICALLY LOOK AT RXERR &
1491                                                                 ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1492   004412                           66$:
1493   004412  104400                              SCOPE
1494                                              ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1495                                              ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1496                                              ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1497                                              ;;(OVRRUN,RXERR)
```

```
1498                                           ;;MODE:ISYMOD
1499                                           ;;LENGTH:SIX
1500                                           ;;CHAR:0
1501                                           ;;
1502   004414  012767  000004  174504  TST4:   MOV     #4,TSTNO        ;SAVE THIS
1503   004422  012767  004656  174466          MOV     #TST5,NEXT          ;GO TO THIS TEST WHEN THRU
1504   004430  052777  000400  014122          BIS     #MRESET,@TXCSR  ;MASTER RESET
1505   004436  012777  000000  014110          MOV     #ISYMOD,@PARCSR ;SET THE MODE
1506   004444  052777  000400  014106          BIS     #MRESET,@TXCSR  ;MASTER RESET
1507
1508                                           ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1509   004452  012777  064001  014100          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1510
1511                                           ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1512   004460  012777  002000  014066          MOV     #ISYMOD!SIX!NOPAR!0,@PARCSR
1513   004466  052777  000020  014050          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1514                                           ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1515   004474  042777  020000  014056          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1516   004502  052777  020000  014050          BIS     #CLK,@TXCSR     ;POKE CLK UP
1517                                           ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1518   004510  042777  020000  014042          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1519   004516  052777  020000  014034          BIS     #CLK,@TXCSR     ;POKE CLK UP
1520   004524  016703  014020                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1521   004530  012700  000000                  MOV     #0,R0   ;EXPECTED
1522   004534  012767  000010  174376          MOV     #8.,SHIFT       ;# OF SHIFTS
1523   004542  012767  000200  174374          MOV     #200,TEMP1      ;DATA CHAR
1524   004550  004767  013474                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1525   004554  105777  013764                  TSTB    @RXCSR  ;RXDONE ?
1526   004560  100401                          BMI     64$
1527   004562  104000                          HLT     ;RXDONE SHOULD BE SET
1528                                    64$:
1529   004564  017701  013760                  MOV     @RXDBUF,R1      ;ACTUAL
1530   004570  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1531   004572  001401                          BEQ     65$
1532   004574  104002                          HLT     2       ;RECEIVED DATA DID NOT MATCH
1533                                                           ;EXPECTED DATA - CHECK MAINT DATA
1534                                                           ;OR RECEIVER LOGIC
1535   004576                           65$:
1536   004576  012767  000010  174334          MOV     #8.,SHIFT       ;# OF SHIFTS
1537   004604  012767  000200  174332          MOV     #200,TEMP1      ;DATA CHAR
1538   004612  004767  013432                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1539                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1540   004616  012767  000010  174314          MOV     #8.,SHIFT       ;# OF SHIFTS
1541   004624  012767  000200  174312          MOV     #200,TEMP1      ;DATA CHAR
1542   004632  004767  013412                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1543   004636  012700  140000                  MOV     #140000!0,R0    ;EXPECTED DATA PLUS
1544                                                           ;RXERR & OVRRUN
1545   004642  017701  013702                  MOV     @RXDBUF,R1      ;ACTUAL
1546   004646  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
1547   004650  001401                          BEQ     66$
1548   004652  104002                          HLT     2       ;SPECIFICALLY LOOK AT RXERR &
1549                                                           ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1550   004654                           66$:
1551   004654  104400                          SCOPE
1552                                           ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1553                                           ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
```

```
1554                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1555                                          ;;(OVRRUN,RXERR)
1556                                          ;;MODE:ISYMOD
1557                                          ;;LENGTH:SEVEN
1558                                          ;;CHAR:125
1559                                          ;;
1560   004656  012767  000005  174242  TST5:  MOV     #5,TSTNO          ;SAVE THIS
1561   004664  012767  005120  174224         MOV     #TST6,NEXT            ;GO TO THIS TEST WHEN THRU
1562   004672  052777  000400  013660         BIS     #MRESET,@TXCSR   ;MASTER RESET
1563   004700  012777  000000  013646         MOV     #ISYMOD,@PARCSR  ;SET THE MODE
1564   004706  052777  000400  013644         BIS     #MRESET,@TXCSR   ;MASTER RESET
1565
1566                                          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1567   004714  012777  064001  013636         MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1568
1569                                          ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1570   004722  012777  004000  013624         MOV     #ISYMOD!SEVEN!NOPAR!0,@PARCSR
1571   004730  052777  000020  013606         BIS     #SYNSCH,@RXCSR   ;SET SYNC SEARCH
1572                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1573   004736  042777  020000  013614         BIC     #CLK,@TXCSR         ;POKE CLK DOWN
1574   004744  052777  020000  013606         BIS     #CLK,@TXCSR         ;POKE CLK UP
1575                                          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1576   004752  042777  020000  013600         BIC     #CLK,@TXCSR         ;POKE CLK DOWN
1577   004760  052777  020000  013572         BIS     #CLK,@TXCSR         ;POKE CLK UP
1578   004766  016703  013556                 MOV     RXDBUF,R3           ;SET UP FOR ERROR MESSAGE
1579   004772  012700  000125                 MOV     #125,R0 ;EXPECTED
1580   004776  012767  000011  174134         MOV     #9.,SHIFT        ;# OF SHIFTS
1581   005004  012767  000652  174132         MOV     #652,TEMP1       ;DATA CHAR
1582   005012  004767  013232                 JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
1583   005016  105777  013522                 TSTB    @RXCSR  ;RXDONE ?
1584   005022  100401                         BMI     64$
1585   005024  104000                         HLT     ;RXDONE SHOULD BE SET
1586   005026                          64$:
1587   005026  017701  013516                 MOV     @RXDBUF,R1       ;ACTUAL
1588   005032  020001                         CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1589   005034  001401                         BEQ     65$
1590   005036  104002                         HLT     2                ;RECEIVED DATA DID NOT MATCH
1591                                                                   ;EXPECTED DATA - CHECK MAINT DATA
1592                                                                   ;OR RECEIVER LOGIC
1593   005040                          65$:
1594   005040  012767  000011  174072         MOV     #9.,SHIFT        ;# OF SHIFTS
1595   005046  012767  000652  174070         MOV     #652,TEMP1       ;DATA CHAR
1596   005054  004767  013170                 JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
1597                                          ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1598   005060  012767  000011  174052         MOV     #9.,SHIFT        ;# OF SHIFTS
1599   005066  012767  000652  174050         MOV     #652,TEMP1       ;DATA CHAR
1600   005074  004767  013150                 JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
1601   005100  012700  140125                 MOV     #140000!125,R0   ;EXPECTED DATA PLUS
1602                                                                   ;RXERR & OVRRUN
1603   005104  017701  013440                 MOV     @RXDBUF,R1       ;ACTUAL
1604   005110  020001                         CMP     R0,R1   ;COMPARE EXP VS. ACT
1605   005112  001401                         BEQ     66$
1606   005114  104002                         HLT     2                ;SPECIFICALLY LOOK AT RXERR &
1607                                                                   ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1608   005116                          66$:
1609   005116  104400                         SCOPE
```

```
1610                                          ::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1611                                          ::RECEIVER SECTION,IT USES THE ERROR FLAGS
1612                                          ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1613                                          :: (OVRRUN,RXERR)
1614                                          ::MODE:ISYMOD
1615                                          ::LENGTH:SEVEN
1616                                          ::CHAR:52
1617                                          ::
1618    005120  012767  000006  174000  TST6:    MOV     #6,TSTNO         ;SAVE THIS
1619    005126  012767  005362  173762           MOV     #TST7,NEXT              ;GO TO THIS TEST WHEN THRU
1620    005134  052777  000400  013416           BIS     #MRESET,@TXCSR   ;MASTER RESET
1621    005142  012777  000000  013404           MOV     #ISYMOD,@PARCSR  ;SET THE MODE
1622    005150  052777  000400  013402           BIS     #MRESET,@TXCSR   ;MASTER RESET
1623
1624                                      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1625    005156  012777  064001  013374           MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1626
1627                                      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1628    005164  012777  004000  013362           MOV     #ISYMOD!SEVEN!NOPAR!0,@PARCSR
1629    005172  052777  000020  013344           BIS     #SYNSCH,@RXCSR   ;SET SYNC SEARCH
1630                                      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1631    005200  042777  020000  013352           BIC     #CLK,@TXCSR      ;POKE CLK DOWN
1632    005206  052777  020000  013344           BIS     #CLK,@TXCSR      ;POKE CLK UP
1633                                      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1634    005214  042777  020000  013336           BIC     #CLK,@TXCSR      ;POKE CLK DOWN
1635    005222  052777  020000  013330           BIS     #CLK,@TXCSR      ;POKE CLK UP
1636    005230  016703  013314                    MOV     RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
1637    005234  012700  000052                    MOV     #52,R0   ;EXPECTED
1638    005240  012767  000011  173672           MOV     #9.,SHIFT        ;# OF SHIFTS
1639    005246  012767  000524  173670           MOV     #524,TEMP1       ;DATA CHAR
1640    005254  004767  012770                    JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
1641    005260  105777  013260                    TSTB    @RXCSR   ;RXDONE ?
1642    005264  100401                            BMI     64$
1643    005266  104000                            HLT      ;RXDONE SHOULD BE SET
1644    005270                          64$:
1645    005270  017701  013254                    MOV     @RXDBUF,R1       ;ACTUAL
1646    005274  020001                            CMP     R0,R1    ;COMPARE EXPECTED VS. ACTUAL
1647    005276  001401                            BEQ     65$
1648    005300  104002                            HLT     2       ;RECEIVED DATA DID NOT MATCH
1649                                                                ;EXPECTED DATA - CHECK MAINT DATA
1650                                                                ;OR RECEIVER LOGIC
1651    005302                          65$:
1652    005302  012767  000011  173630           MOV     #9.,SHIFT        ;# OF SHIFTS
1653    005310  012767  000524  173626           MOV     #524,TEMP1       ;DATA CHAR
1654    005316  004767  012726                    JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
1655                                      ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1656    005322  012767  000011  173610           MOV     #9.,SHIFT        ;# OF SHIFTS
1657    005330  012767  000524  173606           MOV     #524,TEMP1       ;DATA CHAR
1658    005336  004767  012706                    JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
1659    005342  012700  140052                    MOV     #140000!52,R0    ;EXPECTED DATA PLUS
1660                                                                        ;RXERR & OVRRUN
1661    005346  017701  013176                    MOV     @RXDBUF,R1       ;ACTUAL
1662    005352  020001                            CMP     R0,R1    ;COMPARE EXP VS. ACT
1663    005354  001401                            BEQ     66$
1664    005356  104002                            HLT     2       ;SPECIFICALLY LOOK AT RXERR &
1665                                                                ;OVRRUN BITS...THEY BOTH SHOULD BE SET
```

```
1666  005360                    66$:
1667  005360  104400                    SCOPE
1668                                     ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1669                                     ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1670                                     ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1671                                     ;;(OVRRUN,RXERR)
1672                                     ;;MODE:ISYMOD
1673                                     ;;LENGTH:SEVEN
1674                                     ;;CHAR:177
1675                                     ;;
1676  005362  012767  000007  173536  TST7:  MOV     #7,TSTNO        ;SAVE THIS
1677  005370  012767  005624  173520         MOV     #TST8,NEXT              ;GO TO THIS TEST WHEN THRU
1678  005376  052777  000400  013154         BIS     #MRESET,@TXCSR  ;MASTER RESET
1679  005404  012777  000000  013142         MOV     #ISYMOD,@PARCSR ;SET THE MODE
1680  005412  052777  000400  013140         BIS     #MRESET,@TXCSR  ;MASTER RESET
1681
1682                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1683  005420  012777  064001  013132         MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1684
1685                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1686  005426  012777  004000  013120         MOV     #ISYMOD!SEVEN!NOPAR!0,@PARCSR
1687  005434  052777  000020  013102         BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1688                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1689  005442  042777  020000  013110         BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1690  005450  052777  020000  013102         BIS     #CLK,@TXCSR     ;POKE CLK UP
1691                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1692  005456  042777  020000  013074         BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1693  005464  052777  020000  013066         BIS     #CLK,@TXCSR     ;POKE CLK UP
1694  005472  016703  013052                 MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1695  005476  012700  000177                 MOV     #177,R0 ;EXPECTED
1696  005502  012767  000011  173430         MOV     #9.,SHIFT       ;# OF SHIFTS
1697  005510  012767  000776  173426         MOV     #776,TEMP1      ;DATA CHAR
1698  005516  004767  012526                 JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1699  005522  105777  013016                 TSTB    @RXCSR  ;RXDONE ?
1700  005526  100401                         BMI     64$
1701  005530  104000                         HLT     ;RXDONE SHOULD BE SET
1702  005532                    64$:
1703  005532  017701  013012                 MOV     @RXDBUF,R1      ;ACTUAL
1704  005536  020001                         CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1705  005540  001401                         BEQ     65$
1706  005542  104002                         HLT     2       ;RECEIVED DATA DID NOT MATCH
1707                                                          ;EXPECTED DATA - CHECK MAINT DATA
1708                                                          ;OR RECEIVER LOGIC
1709  005544                    65$:
1710  005544  012767  000011  173366         MOV     #9.,SHIFT       ;# OF SHIFTS
1711  005552  012767  000776  173364         MOV     #776,TEMP1      ;DATA CHAR
1712  005560  004767  012464                 JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1713                                   ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1714  005564  012767  000011  173346         MOV     #9.,SHIFT       ;# OF SHIFTS
1715  005572  012767  000776  173344         MOV     #776,TEMP1      ;DATA CHAR
1716  005600  004767  012444                 JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1717  005604  012700  140177                 MOV     #140000!177,R0  ;EXPECTED DATA PLUS
1718                                                                  ;RXERR & OVRRUN
1719  005610  017701  012734                 MOV     @RXDBUF,R1      ;ACTUAL
1720  005614  020001                         CMP     R0,R1   ;COMPARE EXP VS. ACT
1721  005616  001401                         BEQ     66$
```

```
1722  005620  104002                        HLT     2           ;SPECIFICALLY LOOK AT RXERR &
1723                                                             ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1724  005622                        66$:
1725  005622  104400                        SCOPE
1726                                         ::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1727                                         ::RECEIVER SECTION,IT USES THE ERROR FLAGS
1728                                         ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1729                                         ::(OVRRUN,RXERR)
1730                                         ::MODE:ISYMOD
1731                                         ::LENGTH:SEVEN
1732                                         ::CHAR:0
1733                                         ::
1734  005624  012767  000010  173274  TST8:  MOV     #8,TSTNO         ;SAVE THIS
1735  005632  012767  006066  173256         MOV     #TST9,NEXT                ;GO TO THIS TEST WHEN THRU
1736  005640  052777  000400  012712         BIS     #MRESET,@TXCSR   ;MASTER RESET
1737  005646  012777  000000  012700         MOV     #ISYMOD,@PARCSR  ;SET THE MODE
1738  005654  052777  000400  012676         BIS     #MRESET,@TXCSR   ;MASTER RESET
1739
1740                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1741  005662  012777  064001  012670         MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1742
1743                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1744  005670  012777  004000  012656         MOV     #ISYMOD!SEVEN!NOPAR!0,@PARCSR
1745  005676  052777  000020  012640         BIS     #SYNSCH,@RXCSR   ;SET SYNC SEARCH
1746                                  ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1747  005704  042777  020000  012646         BIC     #CLK,@TXCSR      ;POKE CLK DOWN
1748  005712  052777  020000  012640         BIS     #CLK,@TXCSR      ;POKE CLK UP
1749                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1750  005720  042777  020000  012632         BIC     #CLK,@TXCSR      ;POKE CLK DOWN
1751  005726  052777  020000  012624         BIS     #CLK,@TXCSR      ;POKE CLK UP
1752  005734  016703  012610                 MOV     RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
1753  005740  012700  000000                 MOV     #0,R0    ;EXPECTED
1754  005744  012767  000011  173166         MOV     #9.,SHIFT        ;# OF SHIFTS
1755  005752  012767  000400  173164         MOV     #400,TEMP1       ;DATA CHAR
1756  005760  004767  012264                 JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
1757  005764  105777  012554                 TSTB    @RXCSR   ;RXDONE ?
1758  005770  100401                         BMI     64$
1759  005772  104000                         HLT     ;RXDONE SHOULD BE SET
1760  005774                          64$:
1761  005774  017701  012550                 MOV     @RXDBUF,R1       ;ACTUAL
1762  006000  020001                         CMP     R0,R1    ;COMPARE EXPECTED VS. ACTUAL
1763  006002  001401                         BEQ     65$
1764  006004  104002                         HLT     2           ;RECEIVED DATA DID NOT MATCH
1765                                                              ;EXPECTED DATA - CHECK MAINT DATA
1766                                                              ;OR RECEIVER LOGIC
1767  006006                          65$:
1768  006006  012767  000011  173124         MOV     #9.,SHIFT        ;# OF SHIFTS
1769  006014  012767  000400  173122         MOV     #400,TEMP1       ;DATA CHAR
1770  006022  004767  012222                 JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
1771                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1772  006026  012767  000011  173104         MOV     #9.,SHIFT        ;# OF SHIFTS
1773  006034  012767  000400  173102         MOV     #400,TEMP1       ;DATA CHAR
1774  006042  004767  012202                 JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
1775  006046  012700  140000                 MOV     #140000!0,R0     ;EXPECTED DATA PLUS
1776                                                                  ;RXERR & OVRRUN
1777  006052  017701  012472                 MOV     @RXDBUF,R1       ;ACTUAL
```

```
1778  006056  020001                      CMP    R0,R1    ;COMPARE EXP VS. ACT
1779  006060  001401                      BEQ    66$
1780  006062  104002                      HLT    2        ;SPECIFICALLY LOOK AT RXERR &
1781                                                       ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1782  006064                       66$:
1783  006064  104400                      SCOPE
1784                                       ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1785                                       ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1786                                       ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1787                                       ;;(OVRRUN,RXERR)
1788                                       ;;MODE:ISYMOD
1789                                       ;;LENGTH:EIGHT
1790                                       ;;CHAR:125
1791                                       ;;
1792  006066  012767  000011  173032  TST9:  MOV  #9,TSTNO         ;SAVE THIS
1793  006074  012767  006330  173014      MOV    #TST10,NEXT             ;GO TO THIS TEST WHEN THRU
1794  006102  052777  000400  012450      BIS    #MRESET,@TXCSR   ;MASTER RESET
1795  006110  012777  000000  012436      MOV    #ISYMOD,@PARCSR  ;SET THE MODE
1796  006116  052777  000400  012434      BIS    #MRESET,@TXCSR   ;MASTER RESET
1797
1798                                       ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1799  006124  012777  064001  012426      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1800
1801                                       ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1802  006132  012777  006000  012414      MOV    #ISYMOD!EIGHT!NOPAR!0,@PARCSR
1803  006140  052777  000020  012376      BIS    #SYNSCH,@RXCSR   ;SET SYNC SEARCH
1804                                       ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1805  006146  042777  020000  012404      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1806  006154  052777  020000  012376      BIS    #CLK,@TXCSR      ;POKE CLK UP
1807                                       ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1808  006162  042777  020000  012370      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1809  006170  052777  020000  012362      BIS    #CLK,@TXCSR      ;POKE CLK UP
1810  006176  016703  012346              MOV    RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
1811  006202  012700  000125              MOV    #125,R0  ;EXPECTED
1812  006206  012767  000012  172724      MOV    #10.,SHIFT       ;# OF SHIFTS
1813  006214  012767  001252  172722      MOV    #1252,TEMP1      ;DATA CHAR
1814  006222  004767  012022              JSR    PC,RPOKE         ;SHIFT IN THIS CHAR
1815  006226  105777  012312              TSTB   @RXCSR   ;RXDONE ?
1816  006232  100401                      BMI    64$
1817  006234  104000                      HLT    ;RXDONE SHOULD BE SET
1818  006236                       64$:
1819  006236  017701  012306              MOV    @RXDBUF,R1       ;ACTUAL
1820  006242  020001                      CMP    R0,R1    ;COMPARE EXPECTED VS. ACTUAL
1821  006244  001401                      BEQ    65$
1822  006246  104002                      HLT    2        ;RECEIVED DATA DID NOT MATCH
1823                                                       ;EXPECTED DATA - CHECK MAINT DATA
1824                                                       ;OR RECEIVER LOGIC
1825  006250                       65$:
1826  006250  012767  000012  172662      MOV    #10.,SHIFT       ;# OF SHIFTS
1827  006256  012767  001252  172660      MOV    #1252,TEMP1      ;DATA CHAR
1828  006264  004767  011760              JSR    PC,RPOKE         ;SHIFT IN THIS CHAR
1829                                       ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1830  006270  012767  000012  172642      MOV    #10.,SHIFT       ;# OF SHIFTS
1831  006276  012767  001252  172640      MOV    #1252,TEMP1      ;DATA CHAR
1832  006304  004767  011740              JSR    PC,RPOKE         ;SHIFT IN THIS CHAR
1833  006310  012700  140125              MOV    #140000!125,R0   ;EXPECTED DATA PLUS
```

```
1834                                                        ;RXERR & OVRRUN
1835    006314  017701  012230              MOV    @RXDBUF,R1        ;ACTUAL
1836    006320  020001                      CMP    R0,R1    ;COMPARE EXP VS. ACT
1837    006322  001401                      BEQ    66$
1838    006324  104002                      HLT    2         ;SPECIFICALLY LOOK AT RXERR &
1839                                                         ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1840    006326                      66$:
1841    006326  104400                      SCOPE
1842                                ::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1843                                ::RECEIVER SECTION,IT USES THE ERROR FLAGS
1844                                ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1845                                ::(OVRRUN,RXERR)
1846                                ::MODE:ISYMOD
1847                                ::LENGTH:EIGHT
1848                                ::CHAR:252
1849                                ::
1850    006330  012767  000012  172570  TST10:  MOV    #10,TSTNO          ;SAVE THIS
1851    006336  012767  006572  172552      MOV    #TST11,NEXT            ;GO TO THIS TEST WHEN THRU
1852    006344  052777  000400  012206      BIS    #MRESET,@TXCSR  ;MASTER RESET
1853    006352  012777  000000  012174      MOV    #ISYMOD,@PARCSR ;SET THE MODE
1854    006360  052777  000400  012172      BIS    #MRESET,@TXCSR  ;MASTER RESET
1855
1856                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1857    006366  012777  064001  012164      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1858
1859                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1860    006374  012777  006000  012152      MOV    #ISYMOD!EIGHT!NOPAR!0,@PARCSR
1861    006402  052777  000020  012134      BIS    #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1862                                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1863    006410  042777  020000  012142      BIC    #CLK,@TXCSR       ;POKE CLK DOWN
1864    006416  052777  020000  012134      BIS    #CLK,@TXCSR       ;POKE CLK UP
1865                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1866    006424  042777  020000  012126      BIC    #CLK,@TXCSR       ;POKE CLK DOWN
1867    006432  052777  020000  012120      BIS    #CLK,@TXCSR       ;POKE CLK UP
1868    006440  016703  012104              MOV    RXDBUF,R3         ;SET UP FOR ERROR MESSAGE
1869    006444  012700  000252              MOV    #252,R0 ;EXPECTED
1870    006450  012767  000012  172462      MOV    #10.,SHIFT        ;# OF SHIFTS
1871    006456  012767  001524  172460      MOV    #1524,TEMP1       ;DATA CHAR
1872    006464  004767  011560              JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
1873    006470  105777  012050              TSTB   @RXCSR  ;RXDONE ?
1874    006474  100401                      BMI    64$
1875    006476  104000                      HLT    ;RXDONE SHOULD BE SET
1876    006500                      64$:
1877    006500  017701  012044              MOV    @RXDBUF,R1        ;ACTUAL
1878    006504  020001                      CMP    R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1879    006506  001401                      BEQ    65$
1880    006510  104002                      HLT    2         ;RECEIVED DATA DID NOT MATCH
1881                                                         ;EXPECTED DATA - CHECK MAINT DATA
1882                                                         ;OR RECEIVER LOGIC
1883    006512                      65$:
1884    006512  012767  000012  172420      MOV    #10.,SHIFT        ;# OF SHIFTS
1885    006520  012767  001524  172416      MOV    #1524,TEMP1       ;DATA CHAR
1886    006526  004767  011516              JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
1887                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1888    006532  012767  000012  172400      MOV    #10.,SHIFT        ;# OF SHIFTS
1889    006540  012767  001524  172376      MOV    #1524,TEMP1       ;DATA CHAR
```

```
1890  006546  004767  011476         JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
1891  006552  012700  140252         MOV    #140000!252,R0    ;EXPECTED DATA PLUS
1892                                                          ;RXERR & OVRRUN
1893  006556  017701  011766         MOV    @RXDBUF,R1        ;ACTUAL
1894  006562  020001                 CMP    R0,R1    ;COMPARE EXP VS. ACT
1895  006564  001401                 BEQ    66$
1896  006566  104002                 HLT    2                 ;SPECIFICALLY LOOK AT RXERR &
1897                                                          ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1898  006570                  66$:
1899  006570  104400                 SCOPE
1900                                  ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1901                                  ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1902                                  ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1903                                  ;;(OVRRUN,RXERR)
1904                                  ;;MODE:ISYMOD
1905                                  ;;LENGTH:EIGHT
1906                                  ;;CHAR:377
1907                                  ;;
1908  006572  012767  000013  172326 TST11: MOV #11,TSTNO     ;SAVE THIS
1909  006600  012767  007034  172310        MOV #TST12,NEXT        ;GO TO THIS TEST WHEN THRU
1910  006606  052777  000400  011744        BIS #MRESET,@TXCSR ;MASTER RESET
1911  006614  012777  000000  011732        MOV #ISYMOD,@PARCSR ;SET THE MODE
1912  006622  052777  000400  011730        BIS #MRESET,@TXCSR ;MASTER RESET
1913
1914                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1915  006630  012777  064001  011722        MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1916
1917                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1918  006636  012777  006000  011710        MOV #ISYMOD!EIGHT!NOPAR!0,@PARCSR
1919  006644  052777  000020  011672        BIS #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1920                                         ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1921  006652  042777  020000  011700        BIC #CLK,@TXCSR    ;POKE CLK DOWN
1922  006660  052777  020000  011672        BIS #CLK,@TXCSR    ;POKE CLK UP
1923                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1924  006666  042777  020000  011664        BIC #CLK,@TXCSR    ;POKE CLK DOWN
1925  006674  052777  020000  011656        BIS #CLK,@TXCSR    ;POKE CLK UP
1926  006702  016703  011642               MOV RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1927  006706  012700  000377               MOV #377,R0 ;EXPECTED
1928  006712  012767  000012  172220        MOV #10.,SHIFT     ;# OF SHIFTS
1929  006720  012767  001776  172216        MOV #1776,TEMP1    ;DATA CHAR
1930  006726  004767  011316               JSR PC,RPOKE       ;SHIFT IN THIS CHAR
1931  006732  105777  011606               TSTB @RXCSR ;RXDONE ?
1932  006736  100401                       BMI 64$
1933  006740  104000                       HLT ;RXDONE SHOULD BE SET
1934  006742                  64$:
1935  006742  017701  011602               MOV @RXDBUF,R1     ;ACTUAL
1936  006746  020001                       CMP R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1937  006750  001401                       BEQ 65$
1938  006752  104002                       HLT 2              ;RECEIVED DATA DID NOT MATCH
1939                                                          ;EXPECTED DATA - CHECK MAINT DATA
1940                                                          ;OR RECEIVER LOGIC
1941  006754                  65$:
1942  006754  012767  000012  172156        MOV #10.,SHIFT    ;# OF SHIFTS
1943  006762  012767  001776  172154        MOV #1776,TEMP1   ;DATA CHAR
1944  006770  004767  011254               JSR PC,RPOKE       ;SHIFT IN THIS CHAR
1945                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
```

```
1946  006774  012767  000012  172136          MOV     #10.,SHIFT        ;# OF SHIFTS
1947  007002  012767  001776  172134          MOV     #1776,TEMP1       ;DATA CHAR
1948  007010  004767  011234                  JSR     PC,RPOKE          ;SHIFT IN THIS CHAR
1949  007014  012700  140377                  MOV     #140000!377,R0    ;EXPECTED DATA PLUS
1950                                                                     ;RXERR & OVRRUN
1951  007020  017701  011524                  MOV     @RXDBUF,R1        ;ACTUAL
1952  007024  020001                          CMP     R0,R1    ;COMPARE EXP VS. ACT
1953  007026  001401                          BEQ     66$
1954  007030  104002                          HLT     2                 ;SPECIFICALLY LOOK AT RXERR &
1955                                                                     ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1956  007032                          66$:
1957  007032  104400                          SCOPE
1958                                           ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1959                                           ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1960                                           ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1961                                           ;;(OVRRUN,RXERR)
1962                                           ;;MODE:ISYMOD
1963                                           ;;LENGTH:EIGHT
1964                                           ;;CHAR:0
1965                                           ;;
1966  007034  012767  000014  172064  TST12:  MOV     #12,TSTNO         ;SAVE THIS
1967  007042  012767  007276  172046          MOV     #TST13,NEXT              ;GO TO THIS TEST WHEN THRU
1968  007050  052777  000400  011502          BIS     #MRESET,@TXCSR  ;MASTER RESET
1969  007056  012777  000000  011470          MOV     #ISYMOD,@PARCSR ;SET THE MODE
1970  007064  052777  000400  011466          BIS     #MRESET,@TXCSR  ;MASTER RESET
1971
1972                                           ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1973  007072  012777  064001  011460          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1974
1975                                           ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1976  007100  012777  006000  011446          MOV     #ISYMOD!EIGHT!NOPAR!0,@PARCSR
1977  007106  052777  000020  011430          BIS     #SYNSCH,@RXCSR   ;SET SYNC SEARCH
1978                                           ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1979  007114  042777  020000  011436          BIC     #CLK,@TXCSR      ;POKE CLK DOWN
1980  007122  052777  020000  011430          BIS     #CLK,@TXCSR      ;POKE CLK UP
1981                                           ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1982  007130  042777  020000  011422          BIC     #CLK,@TXCSR      ;POKE CLK DOWN
1983  007136  052777  020000  011414          BIS     #CLK,@TXCSR      ;POKE CLK UP
1984  007144  016703  011400                  MOV     RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
1985  007150  012700  000000                  MOV     #0,R0    ;EXPECTED
1986  007154  012767  000012  171756          MOV     #10.,SHIFT        ;# OF SHIFTS
1987  007162  012767  001000  171754          MOV     #1000,TEMP1       ;DATA CHAR
1988  007170  004767  011054                  JSR     PC,RPOKE          ;SHIFT IN THIS CHAR
1989  007174  105777  011344                  TSTB    @RXCSR   ;RXDONE ?
1990  007200  100401                          BMI     64$
1991  007202  104000                          HLT     ;RXDONE SHOULD BE SET
1992  007204                          64$:
1993  007204  017701  011340                  MOV     @RXDBUF,R1        ;ACTUAL
1994  007210  020001                          CMP     R0,R1    ;COMPARE EXPECTED VS. ACTUAL
1995  007212  001401                          BEQ     65$
1996  007214  104002                          HLT     2                 ;RECEIVED DATA DID NOT MATCH
1997                                                                     ;EXPECTED DATA - CHECK MAINT DATA
1998                                                                     ;OR RECEIVER LOGIC
1999  007216                          65$:
2000  007216  012767  000012  171714          MOV     #10.,SHIFT        ;# OF SHIFTS
2001  007224  012767  001000  171712          MOV     #1000,TEMP1       ;DATA CHAR
```

```
2002   007232   004767   011012              JSR       PC,RPOKE        ;SHIFT IN THIS CHAR
2003                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2004   007236   012767   000012   171674     MOV       #10.,SHIFT      ;# OF SHIFTS
2005   007244   012767   001000   171672     MOV       #1000,TEMP1     ;DATA CHAR
2006   007252   004767   010772              JSR       PC,RPOKE        ;SHIFT IN THIS CHAR
2007   007256   012700   140000              MOV       #140000!0,R0    ;EXPECTED DATA PLUS
2008                                                                   ;RXERR & OVRRUN
2009   007262   017701   011262              MOV       @RXDBUF,R1      ;ACTUAL
2010   007266   020001                       CMP       R0,R1    ;COMPARE EXP VS. ACT
2011   007270   001401                       BEQ       66$
2012   007272   104002                       HLT       2               ;SPECIFICALLY LOOK AT RXERR &
2013                                                                   ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2014   007274                       66$:
2015   007274   104400                       SCOPE
2016                                ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2017                                ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2018                                ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2019                                ;;(OVRRUN,RXERR)
2020                                ;;MODE:SYNEXT
2021                                ;;LENGTH:FIVE
2022                                ;;CHAR:25
2023                                ;;
2024   007276   012767   000015   171622   TST13:  MOV   #13,TSTNO        ;SAVE THIS
2025   007304   012767   007524   171604     MOV       #TST14,NEXT                ;GO TO THIS TEST WHEN THRU
2026   007312   052777   000400   011240     BIS       #MRESET,@TXCSR  ;MASTER RESET
2027   007320   012777   020000   011226     MOV       #SYNEXT,@PARCSR ;SET THE MODE
2028   007326   052777   000400   011224     BIS       #MRESET,@TXCSR  ;MASTER RESET
2029
2030                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2031   007334   012777   064001   011216     MOV       #MTDATA!CLK!MINT!BREAK,@TXCSR
2032
2033                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2034   007342   012777   020000   011204     MOV       #SYNEXT!FIVE!NOPAR!0,@PARCSR
2035   007350   052777   000020   011166     BIS       #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2036                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2037   007356   042777   020000   011174     BIC       #CLK,@TXCSR     ;POKE CLK DOWN
2038   007364   052777   020000   011166     BIS       #CLK,@TXCSR     ;POKE CLK UP
2039   007372   016703   011152              MOV       RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2040   007376   012700   000025              MOV       #25,R0   ;EXPECTED
2041   007402   012767   000005   171530     MOV       #5,SHIFT        ;# OF SHIFTS
2042   007410   012767   000025   171526     MOV       #25,TEMP1       ;DATA CHAR
2043   007416   004767   010626              JSR       PC,RPOKE        ;SHIFT IN THIS CHAR
2044   007422   105777   011116              TSTB      @RXCSR   ;RXDONE ?
2045   007426   100401                       BMI       64$
2046   007430   104000                       HLT       ;RXDONE SHOULD BE SET
2047   007432                       64$:
2048   007432   017701   011112              MOV       @RXDBUF,R1      ;ACTUAL
2049   007436   020001                       CMP       R0,R1    ;COMPARE EXPECTED VS. ACTUAL
2050   007440   001401                       BEQ       65$
2051   007442   104002                       HLT       2               ;RECEIVED DATA DID NOT MATCH
2052                                                                   ;EXPECTED DATA - CHECK MAINT DATA
2053                                                                   ;OR RECEIVER LOGIC
2054   007444                       65$:
2055   007444   012767   000005   171466     MOV       #5,SHIFT        ;# OF SHIFTS
2056   007452   012767   000025   171464     MOV       #25,TEMP1       ;DATA CHAR
2057   007460   004767   010564              JSR       PC,RPOKE        ;SHIFT IN THIS CHAR
```

```
2058                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2059    007464  012767  000005  171446          MOV     #5,SHIFT        ;# OF SHIFTS
2060    007472  012767  000025  171444          MOV     #25,TEMP1       ;DATA CHAR
2061    007500  004767  010544                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2062    007504  012700  140025                  MOV     #140000!25,R0   ;EXPECTED DATA PLUS
2063                                                                    ;RXERR & OVRRUN
2064    007510  017701  011034                  MOV     @RXDBUF,R1      ;ACTUAL
2065    007514  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
2066    007516  001401                          BEQ     66$
2067    007520  104002                          HLT     2               ;SPECIFICALLY LOOK AT RXERR &
2068                                                                    ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2069    007522                          66$:
2070    007522  104400                          SCOPE
2071                                    ::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2072                                    ::RECEIVER SECTION,IT USES THE ERROR FLAGS
2073                                    ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2074                                    ::(OVRRUN,RXERR)
2075                                    ::MODE:SYNEXT
2076                                    ::LENGTH:FIVE
2077                                    ::CHAR:12
2078                                    ::
2079    007524  012767  000016  171374  TST14:  MOV     #14,TSTNO       ;SAVE THIS
2080    007532  012767  007752  171356          MOV     #TST15,NEXT             ;GO TO THIS TEST WHEN THRU
2081    007540  052777  000400  011012          BIS     #MRESET,@TXCSR  ;MASTER RESET
2082    007546  012777  020000  011000          MOV     #SYNEXT,@PARCSR ;SET THE MODE
2083    007554  052777  000400  010776          BIS     #MRESET,@TXCSR  ;MASTER RESET
2084
2085                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2086    007562  012777  064001  010770          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2087
2088                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2089    007570  012777  020000  010756          MOV     #SYNEXT!FIVE!NOPAR!0,@PARCSR
2090    007576  052777  000020  010740          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2091                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2092    007604  042777  020000  010746          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2093    007612  052777  020000  010740          BIS     #CLK,@TXCSR     ;POKE CLK UP
2094    007620  016703  010724                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2095    007624  012700  000012                  MOV     #12,R0  ;EXPECTED
2096    007630  012767  000005  171302          MOV     #5,SHIFT        ;# OF SHIFTS
2097    007636  012767  000012  171300          MOV     #12,TEMP1       ;DATA CHAR
2098    007644  004767  010400                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2099    007650  105777  010670                  TSTB    @RXCSR  ;RXDONE ?
2100    007654  100401                          BMI     64$
2101    007656  104000                          HLT     ;RXDONE SHOULD BE SET
2102    007660                          64$:
2103    007660  017701  010664                  MOV     @RXDBUF,R1      ;ACTUAL
2104    007664  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2105    007666  001401                          BEQ     65$
2106    007670  104002                          HLT     2               ;RECEIVED DATA DID NOT MATCH
2107                                                                    ;EXPECTED DATA - CHECK MAINT DATA
2108                                                                    ;OR RECEIVER LOGIC
2109    007672                          65$:
2110    007672  012767  000005  171240          MOV     #5,SHIFT        ;# OF SHIFTS
2111    007700  012767  000012  171236          MOV     #12,TEMP1       ;DATA CHAR
2112    007706  004767  010336                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2113                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
```

```
2114  007712  012767  000005  171220          MOV    #5,SHIFT           ;# OF SHIFTS
2115  007720  012767  000012  171216          MOV    #12,TEMP1          ;DATA CHAR
2116  007726  004767  010316                  JSR    PC,RPOKE           ;SHIFT IN THIS CHAR
2117  007732  012700  140012                  MOV    #140000!12,R0      ;EXPECTED DATA PLUS
2118                                                                     ;RXERR & OVRRUN
2119  007736  017701  010606                  MOV    @RXDBUF,R1         ;ACTUAL
2120  007742  020001                          CMP    R0,R1    ;COMPARE EXP VS. ACT
2121  007744  001401                          BEQ    66$
2122  007746  104002                          HLT    2                  ;SPECIFICALLY LOOK AT RXERR &
2123                                                                     ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2124  007750                          66$:
2125  007750  104400                          SCOPE
2126                                    ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2127                                    ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2128                                    ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2129                                    ;;(OVRRUN,RXERR)
2130                                    ;;MODE:SYNEXT
2131                                    ;;LENGTH:FIVE
2132                                    ;;CHAR:37
2133                                    ;;
2134  007752  012767  000017  171146  TST15:  MOV    #15,TSTNO          ;SAVE THIS
2135  007760  012767  010200  171130          MOV    #TST16,NEXT                ;GO TO THIS TEST WHEN THRU
2136  007766  052777  000400  010564          BIS    #MRESET,@TXCSR     ;MASTER RESET
2137  007774  012777  020000  010552          MOV    #SYNEXT,@PARCSR    ;SET THE MODE
2138  010002  052777  000400  010550          BIS    #MRESET,@TXCSR     ;MASTER RESET
2139
2140                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2141  010010  012777  064001  010542          MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2142
2143                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2144  010016  012777  020000  010530          MOV    #SYNEXT!FIVE!NOPAR!0,@PARCSR
2145  010024  052777  000020  010512          BIS    #SYNSCH,@RXCSR    ;SET SEARCH SYNC
2146                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2147  010032  042777  020000  010520          BIC    #CLK,@TXCSR        ;POKE CLK DOWN
2148  010040  052777  020000  010512          BIS    #CLK,@TXCSR        ;POKE CLK UP
2149  010046  016703  010476                  MOV    RXDBUF,R3          ;SET UP FOR ERROR MESSAGE
2150  010052  012700  000037                  MOV    #37,R0    ;EXPECTED
2151  010056  012767  000005  171054          MOV    #5,SHIFT           ;# OF SHIFTS
2152  010064  012767  000037  171052          MOV    #37,TEMP1          ;DATA CHAR
2153  010072  004767  010152                  JSR    PC,RPOKE           ;SHIFT IN THIS CHAR
2154  010076  105777  010442                  TSTB   @RXCSR   ;RXDONE ?
2155  010102  100401                          BMI    64$
2156  010104  104000                          HLT    ;RXDONE SHOULD BE SET
2157  010106                          64$:
2158  010106  017701  010436                  MOV    @RXDBUF,R1         ;ACTUAL
2159  010112  020001                          CMP    R0,R1    ;COMPARE EXPECTED VS. ACTUAL
2160  010114  001401                          BEQ    65$
2161  010116  104002                          HLT    2                  ;RECEIVED DATA DID NOT MATCH
2162                                                                     ;EXPECTED DATA - CHECK MAINT DATA
2163                                                                     ;OR RECEIVER LOGIC
2164  010120                          65$:
2165  010120  012767  000005  171012          MOV    #5,SHIFT           ;# OF SHIFTS
2166  010126  012767  000037  171010          MOV    #37,TEMP1          ;DATA CHAR
2167  010134  004767  010110                  JSR    PC,RPOKE           ;SHIFT IN THIS CHAR
2168                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2169  010140  012767  000005  170772          MOV    #5,SHIFT           ;# OF SHIFTS
```

```
2170  010146  012767  000037  170770          MOV     #37,TEMP1        ;DATA CHAR
2171  010154  004767  010070                  JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
2172  010160  012700  140037                  MOV     #140000!37,R0    ;EXPECTED DATA PLUS
2173                                                                   ;RXERR & OVRRUN
2174  010164  017701  010360                  MOV     @RXDBUF,R1       ;ACTUAL
2175  010170  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
2176  010172  001401                          BEQ     66$
2177  010174  104002                          HLT     2                ;SPECIFICALLY LOOK AT RXERR &
2178                                                                   ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2179  010176                          66$:
2180  010176  104400                          SCOPE
2181                                           ::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2182                                           ::RECEIVER SECTION,IT USES THE ERROR FLAGS
2183                                           ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2184                                           ::(OVRRUN,RXERR)
2185                                           ::MODE:SYNEXT
2186                                           ::LENGTH:FIVE
2187                                           ::CHAR:0
2188                                           ::
2189  010200  012767  000020  170720  TST16:  MOV     #16,TSTNO        ;SAVE THIS
2190  010206  012767  010426  170702          MOV     #TST17,NEXT              ;GO TO THIS TEST WHEN THRU
2191  010214  052777  000400  010336          BIS     #MRESET,@TXCSR   ;MASTER RESET
2192  010222  012777  020000  010324          MOV     #SYNEXT,@PARCSR  ;SET THE MODE
2193  010230  052777  000400  010322          BIS     #MRESET,@TXCSR   ;MASTER RESET
2194
2195                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2196  010236  012777  064001  010314          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2197
2198                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2199  010244  012777  020000  010302          MOV     #SYNEXT!FIVE!NOPAR!0,@PARCSR
2200  010252  052777  000020  010264          BIS     #SYNSCH,@RXCSR   ;SET SEARCH SYNC
2201                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2202  010260  042777  020000  010272          BIC     #CLK,@TXCSR      ;POKE CLK DOWN
2203  010266  052777  020000  010264          BIS     #CLK,@TXCSR      ;POKE CLK UP
2204  010274  016703  010250                  MOV     RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
2205  010300  012700  000000                  MOV     #0,R0    ;EXPECTED
2206  010304  012767  000005  170626          MOV     #5,SHIFT         ;# OF SHIFTS
2207  010312  012767  000000  170624          MOV     #0,TEMP1         ;DATA CHAR
2208  010320  004767  007724                  JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
2209  010324  105777  010214                  TSTB    @RXCSR  ;RXDONE ?
2210  010330  100401                          BMI     64$
2211  010332  104000                          HLT              ;RXDONE SHOULD BE SET
2212  010334                          64$:
2213  010334  017701  010210                  MOV     @RXDBUF,R1       ;ACTUAL
2214  010340  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2215  010342  001401                          BEQ     65$
2216  010344  104002                          HLT     2                ;RECEIVED DATA DID NOT MATCH
2217                                                                   ;EXPECTED DATA - CHECK MAINT DATA
2218                                                                   ;OR RECEIVER LOGIC
2219  010346                          65$:
2220  010346  012767  000005  170564          MOV     #5,SHIFT         ;# OF SHIFTS
2221  010354  012767  000000  170562          MOV     #0,TEMP1         ;DATA CHAR
2222  010362  004767  007662                  JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
2223                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2224  010366  012767  000005  170544          MOV     #5,SHIFT         ;# OF SHIFTS
2225  010374  012767  000000  170542          MOV     #0,TEMP1         ;DATA CHAR
```

```
2226  010402  004767  007642          JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2227  010406  012700  140000          MOV    #140000!0,R0    ;EXPECTED DATA PLUS
2228                                                          ;RXERR & OVRRUN
2229  010412  017701  010132          MOV    @RXDBUF,R1      ;ACTUAL
2230  010416  020001                  CMP    R0,R1  ;COMPARE EXP VS. ACT
2231  010420  001401                  BEQ    66$
2232  010422  104002                  HLT    2               ;SPECIFICALLY LOOK AT RXERR &
2233                                                          ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2234  010424                  66$:
2235  010424  104400                  SCOPE
2236                                   ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2237                                   ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2238                                   ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2239                                   ;;(OVRRUN,RXERR)
2240                                   ;;MODE:SYNEXT
2241                                   ;;LENGTH:SIX
2242                                   ;;CHAR:25
2243                                   ;;
2244  010426  012767  000021  170472  TST17: MOV    #17,TSTNO       ;SAVE THIS
2245  010434  012767  010654  170454         MOV    #TST18,NEXT             ;GO TO THIS TEST WHEN THRU
2246  010442  052777  000400  010110         BIS    #MRESET,@TXCSR  ;MASTER RESET
2247  010450  012777  020000  010076         MOV    #SYNEXT,@PARCSR ;SET THE MODE
2248  010456  052777  000400  010074         BIS    #MRESET,@TXCSR  ;MASTER RESET
2249
2250                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2251  010464  012777  064001  010066         MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2252
2253                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2254  010472  012777  022000  010054         MOV    #SYNEXT!SIX!NOPAR!0,@PARCSR
2255  010500  052777  000020  010036         BIS    #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2256                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2257  010506  042777  020000  010044         BIC    #CLK,@TXCSR     ;POKE CLK DOWN
2258  010514  052777  020000  010036         BIS    #CLK,@TXCSR     ;POKE CLK UP
2259  010522  016703  010022                 MOV    RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2260  010526  012700  000025                 MOV    #25,R0  ;EXPECTED
2261  010532  012767  000006  170400         MOV    #6,SHIFT        ;# OF SHIFTS
2262  010540  012767  000025  170376         MOV    #25,TEMP1       ;DATA CHAR
2263  010546  004767  007476                 JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2264  010552  105777  007766                 TSTB   @RXCSR  ;RXDONE ?
2265  010556  100401                         BMI    64$
2266  010560  104000                         HLT    ;RXDONE SHOULD BE SET
2267  010562                  64$:
2268  010562  017701  007762                 MOV    @RXDBUF,R1      ;ACTUAL
2269  010566  020001                         CMP    R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2270  010570  001401                         BEQ    65$
2271  010572  104002                         HLT    2               ;RECEIVED DATA DID NOT MATCH
2272                                                          ;EXPECTED DATA - CHECK MAINT DATA
2273                                                          ;OR RECEIVER LOGIC
2274  010574                  65$:
2275  010574  012767  000006  170336         MOV    #6,SHIFT        ;# OF SHIFTS
2276  010602  012767  000025  170334         MOV    #25,TEMP1       ;DATA CHAR
2277  010610  004767  007434                 JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2278                                   ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2279  010614  012767  000006  170316         MOV    #6,SHIFT        ;# OF SHIFTS
2280  010622  012767  000025  170314         MOV    #25,TEMP1       ;DATA CHAR
2281  010630  004767  007414                 JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
```

```
2282  010634  012700  140025           MOV    #140000!25,R0      ;EXPECTED DATA PLUS
2283                                                             ;RXERR & OVRRUN
2284  010640  017701  007704           MOV    @RXDBUF,R1        ;ACTUAL
2285  010644  020001                   CMP    R0,R1    ;COMPARE EXP VS. ACT
2286  010646  001401                   BEQ    66$
2287  010650  104002                   HLT    2                ;SPECIFICALLY LOOK AT RXERR &
2288                                                             ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2289  010652                    66$:
2290  010652  104400                   SCOPE
2291                                    ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2292                                    ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2293                                    ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2294                                    ;;(OVRRUN,RXERR)
2295                                    ;;MODE:SYNEXT
2296                                    ;;LENGTH:SIX
2297                                    ;;CHAR:52
2298                                    ;;
2299  010654  012767  000022  170244  TST18: MOV    #18,TSTNO         ;SAVE THIS
2300  010662  012767  011102  170226         MOV    #TST19,NEXT        ;GO TO THIS TEST WHEN THRU
2301  010670  052777  000022  007662         BIS    #MRESET,@TXCSR  ;MASTER RESET
2302  010676  012777  020000  007650         MOV    #SYNEXT,@PARCSR  ;SET THE MODE
2303  010704  052777  000400  007646         BIS    #MRESET,@TXCSR  ;MASTER RESET
2304
2305                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2306  010712  012777  064001  007640         MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2307
2308                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2309  010720  012777  022000  007626         MOV    #SYNEXT!SIX!NOPAR!0,@PARCSR
2310  010726  052777  000020  007610         BIS    #SYNSCH,@RXCSR   ;SET SEARCH SYNC
2311                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2312  010734  042777  020000  007616         BIC    #CLK,@TXCSR        ;POKE CLK DOWN
2313  010742  052777  020000  007610         BIS    #CLK,@TXCSR        ;POKE CLK UP
2314  010750  016703  007574                 MOV    RXDBUF,R3         ;SET UP FOR ERROR MESSAGE
2315  010754  012700  000052                 MOV    #52,R0   ;EXPECTED
2316  010760  012767  000006  170152         MOV    #6,SHIFT          ;# OF SHIFTS
2317  010766  012767  000052  170150         MOV    #52,TEMP1          ;DATA CHAR
2318  010774  004767  007250                 JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
2319  011000  105777  007540                 TSTB   @RXCSR  ;RXDONE ?
2320  011004  100401                         BMI    64$
2321  011006  104000                         HLT    ;RXDONE SHOULD BE SET
2322  011010                    64$:
2323  011010  017701  007534                 MOV    @RXDBUF,R1       ;ACTUAL
2324  011014  020001                         CMP    R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2325  011016  001401                         BEQ    65$
2326  011020  104002                         HLT    2                ;RECEIVED DATA DID NOT MATCH
2327                                                                  ;EXPECTED DATA - CHECK MAINT DATA
2328                                                                  ;OR RECEIVER LOGIC
2329  011022                    65$:
2330  011022  012767  000006  170110         MOV    #6,SHIFT          ;# OF SHIFTS
2331  011030  012767  000052  170106         MOV    #52,TEMP1          ;DATA CHAR
2332  011036  004767  007206                 JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
2333                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2334  011042  012767  000006  170070         MOV    #6,SHIFT          ;# OF SHIFTS
2335  011050  012767  000052  170066         MOV    #52,TEMP1          ;DATA CHAR
2336  011056  004767  007166                 JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
2337  011062  012700  140052                 MOV    #140000!52,R0    ;EXPECTED DATA PLUS
```

```
2338                                                    ;RXERR & OVRRUN
2339   011066  017701  007456              MOV    @RXDBUF,R1     ;ACTUAL
2340   011072  020001                      CMP    R0,R1    ;COMPARE EXP VS. ACT
2341   011074  001401                      BEQ    66$
2342   011076  104002                      HLT    2              ;SPECIFICALLY LOOK AT RXERR &
2343                                                              ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2344   011100                       66$:
2345   011100  104400                      SCOPE
2346                                        ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2347                                        ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2348                                        ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2349                                        ;;(OVRRUN,RXERR)
2350                                        ;;MODE:SYNEXT
2351                                        ;;LENGTH:SIX
2352                                        ;;CHAR:77
2353                                        ;;
2354   011102  012767  000023  170016  TST19:  MOV    #19,TSTNO       ;SAVE THIS
2355   011110  012767  011330  170000          MOV    #TST20,NEXT            ;GO TO THIS TEST WHEN THRU
2356   011116  052777  000400  007434          BIS    #MRESET,@TXCSR  ;MASTER RESET
2357   011124  012777  020000  007422          MOV    #SYNEXT,@PARCSR ;SET THE MODE
2358   011132  052777  000400  007420          BIS    #MRESET,@TXCSR  ;MASTER RESET
2359
2360                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2361   011140  012777  064001  007412          MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2362
2363                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2364   011146  012777  022000  007400          MOV    #SYNEXT!SIX!NOPAR!0,@PARCSR
2365   011154  052777  000020  007362          BIS    #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2366                                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2367   011162  042777  020000  007370          BIC    #CLK,@TXCSR     ;POKE CLK DOWN
2368   011170  052777  020000  007362          BIS    #CLK,@TXCSR     ;POKE CLK UP
2369   011176  016703  007346                  MOV    RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2370   011202  012700  000077                  MOV    #77,R0  ;EXPECTED
2371   011206  012767  000006  167724          MOV    #6,SHIFT        ;# OF SHIFTS
2372   011214  012767  000077  167722          MOV    #77,TEMP1       ;DATA CHAR
2373   011222  004767  007022                  JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2374   011226  105777  007312                  TSTB   @RXCSR  ;RXDONE ?
2375   011232  100401                          BMI    64$
2376   011234  104000                          HLT    ;RXDONE SHOULD BE SET
2377   011236                       64$:
2378   011236  017701  007306                  MOV    @RXDBUF,R1      ;ACTUAL
2379   011242  020001                          CMP    R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2380   011244  001401                          BEQ    65$
2381   011246  104002                          HLT    2               ;RECEIVED DATA DID NOT MATCH
2382                                                                  ;EXPECTED DATA - CHECK MAINT DATA
2383                                                                  ;OR RECEIVER LOGIC
2384   011250                       65$:
2385   011250  012767  000006  167662          MOV    #6,SHIFT        ;# OF SHIFTS
2386   011256  012767  000077  167660          MOV    #77,TEMP1       ;DATA CHAR
2387   011264  004767  006760                  JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2388                                        ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2389   011270  012767  000006  167642          MOV    #6,SHIFT        ;# OF SHIFTS
2390   011276  012767  000077  167640          MOV    #77,TEMP1       ;DATA CHAR
2391   011304  004767  006740                  JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2392   011310  012700  140077                  MOV    #170000!77,R0   ;EXPECTED DATA PLUS
2393                                                                  ;RXERR & OVRRUN
```

```
2394  011314  017701  007230            MOV    @RXDBUF,R1      ;ACTUAL
2395  011320  020001                    CMP    R0,R1   ;COMPARE EXP VS. ACT
2396  011322  001401                    BEQ    66$
2397  011324  104002                    HLT    2               ;SPECIFICALLY LOOK AT RXERR &
2398                                                            ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2399  011326                    66$:
2400  011326  104400                    SCOPE
2401                                     ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2402                                     ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2403                                     ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2404                                     ;;(OVRRUN,RXERR)
2405                                     ;;MODE:SYNEXT
2406                                     ;;LENGTH:SIX
2407                                     ;;CHAR:0
2408                                     ;;
2409  011330  012767  000024  167570 TST20: MOV  #20,TSTNO         ;SAVE THIS
2410  011336  012767  011556  167552    MOV    #TST21,NEXT              ;GO TO THIS TEST WHEN THRU
2411  011344  052777  000400  007206    BIS    #MRESET,@TXCSR  ;MASTER RESET
2412  011352  012777  020000  007174    MOV    #SYNEXT,@PARCSR ;SET THE MODE
2413  011360  052777  000400  007172    BIS    #MRESET,@TXCSR  ;MASTER RESET
2414
2415                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2416  011366  012777  064001  007164    MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2417
2418                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2419  011374  012777  022000  007152    MOV    #SYNEXT!SIX!NOPAR!0,@PARCSR
2420  011402  052777  000020  007134    BIS    #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2421                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2422  011410  042777  020000  007142    BIC    #CLK,@TXCSR      ;POKE CLK DOWN
2423  011416  052777  020000  007134    BIS    #CLK,@TXCSR      ;POKE CLK UP
2424  011424  016703  007120            MOV    RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2425  011430  012700  000000            MOV    #0,R0   ;EXPECTED
2426  011434  012767  000006  167476    MOV    #6,SHIFT         ;# OF SHIFTS
2427  011442  012767  000000  167474    MOV    #0,TEMP1         ;DATA CHAR
2428  011450  004767  006574            JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2429  011454  105777  007064            TSTB   @RXCSR  ;RXDONE ?
2430  011460  100401                    BMI    64$
2431  011462  104000                    HLT    ;RXDONE SHOULD BE SET
2432  011464                    64$:
2433  011464  017701  007060            MOV    @RXDBUF,R1      ;ACTUAL
2434  011470  020001                    CMP    R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2435  011472  001401                    BEQ    65$
2436  011474  104002                    HLT    2               ;RECEIVED DATA DID NOT MATCH
2437                                                            ;EXPECTED DATA - CHECK MAINT DATA
2438                                                            ;OR RECEIVER LOGIC
2439  011476                    65$:
2440  011476  012767  000006  167434    MOV    #6,SHIFT         ;# OF SHIFTS
2441  011504  012767  000000  167432    MOV    #0,TEMP1         ;DATA CHAR
2442  011512  004767  006532            JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2443                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2444  011516  012767  000006  167414    MOV    #6,SHIFT         ;# OF SHIFTS
2445  011524  012767  000000  167412    MOV    #0,TEMP1         ;DATA CHAR
2446  011532  004767  006512            JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2447  011536  012700  140000            MOV    #140000!0,R0    ;EXPECTED DATA PLUS
2448                                                            ;RXERR & OVRRUN
2449  011542  017701  007002            MOV    @RXDBUF,R1      ;ACTUAL
```

```
2450  011546  020001                        CMP     R0,R1    ;COMPARE EXP VS. ACT
2451  011550  001401                        BEQ     66$
2452  011552  104002                        HLT     2          ;SPECIFICALLY LOOK AT RXERR &
2453                                                           ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2454  011554                        66$:
2455  011554  104400                        SCOPE
2456                                  ::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2457                                  ::RECEIVER SECTION,IT USES THE ERROR FLAGS
2458                                  ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2459                                  ::(OVRRUN,RXERR)
2460                                  ::MODE:SYNEXT
2461                                  ::LENGTH:SEVEN
2462                                  ::CHAR:125
2463                                  ::
2464  011556  012767  000025  167342  TST21: MOV     #21,TSTNO        ;SAVE THIS
2465  011564  012767  012004  167324         MOV     #TST22,NEXT         ;GO TO THIS TEST WHEN THRU
2466  011572  052777  000400  006760         BIS     #MRESET,@TXCSR   ;MASTER RESET
2467  011600  012777  020000  006746         MOV     #SYNEXT,@PARCSR  ;SET THE MODE
2468  011606  052777  000400  006744         BIS     #MRESET,@TXCSR   ;MASTER RESET
2469
2470                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2471  011614  012777  064001  006736         MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2472
2473                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2474  011622  012777  024000  006724         MOV     #SYNEXT!SEVEN!NOPAR!0,@PARCSR
2475  011630  052777  000020  006706         BIS     #SYNSCH,@RXCSR   ;SET SEARCH SYNC
2476                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2477  011636  042777  020000  006714         BIC     #CLK,@TXCSR         ;POKE CLK DOWN
2478  011644  052777  020000  006706         BIS     #CLK,@TXCSR         ;POKE CLK UP
2479  011652  016703  006672                 MOV     RXDBUF,R3           ;SET UP FOR ERROR MESSAGE
2480  011656  012700  000125                 MOV     #125,R0  ;EXPECTED
2481  011662  012767  000007  167250         MOV     #7,SHIFT         ;# OF SHIFTS
2482  011670  012767  000125  167246         MOV     #125,TEMP1       ;DATA CHAR
2483  011676  004767  006346                 JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
2484  011702  105777  006636                 TSTB    @RXCSR   ;RXDONE ?
2485  011706  100401                         BMI     64$
2486  011710  104000                         HLT     ;RXDONE SHOULD BE SET
2487  011712                        64$:
2488  011712  017701  006632                 MOV     @RXDBUF,R1       ;ACTUAL
2489  011716  020001                         CMP     R0,R1    ;COMPARE EXPECTED VS. ACTUAL
2490  011720  001401                         BEQ     65$
2491  011722  104002                         HLT     2          ;RECEIVED DATA DID NOT MATCH
2492                                                           ;EXPECTED DATA - CHECK MAINT DATA
2493                                                           ;OR RECEIVER LOGIC
2494  011724                        65$:
2495  011724  012767  000007  167206         MOV     #7,SHIFT         ;# OF SHIFTS
2496  011732  012767  000125  167204         MOV     #125,TEMP1       ;DATA CHAR
2497  011740  004767  006304                 JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
2498                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2499  011744  012767  000007  167166         MOV     #7,SHIFT         ;# OF SHIFTS
2500  011752  012767  000125  167164         MOV     #125,TEMP1       ;DATA CHAR
2501  011760  004767  006264                 JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
2502  011764  012700  140125                 MOV     #140000!125,R0   ;EXPECTED DATA PLUS
2503                                                                  ;RXERR & OVRRUN
2504  011770  017701  006554                 MOV     @RXDBUF,R1       ;ACTUAL
2505  011774  020001                         CMP     R0,R1    ;COMPARE EXP VS. ACT
```

```
2506  011776  001401                      BEQ     66$
2507  012000  104002                      HLT     2            ;SPECIFICALLY LOOK AT RXERR &
2508                                                           ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2509  012002                      66$:
2510  012002  104400                      SCOPE
2511                                       ::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2512                                       ::RECEIVER SECTION,IT USES THE ERROR FLAGS
2513                                       ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2514                                       :: (OVRRUN,RXERR)
2515                                       ::MODE:SYNEXT
2516                                       ::LENGTH:SEVEN
2517                                       ::CHAR:52
2518                                       ::
2519  012004  012767  000026  167114  TST22: MOV   #22,TSTNO       ;SAVE THIS
2520  012012  012767  012232  167076        MOV   #TST23,NEXT           ;GO TO THIS TEST WHEN THRU
2521  012020  052777  000400  006532        BIS   #MRESET,@TXCSR  ;MASTER RESET
2522  012026  012777  020000  006520        MOV   #SYNEXT,@PARCSR ;SET THE MODE
2523  012034  052777  000400  006516        BIS   #MRESET,@TXCSR  ;MASTER RESET
2524
2525                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2526  012042  012777  064001  006510        MOV   #MTDATA!CLK!MINT!BREAK,@TXCSR
2527
2528                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2529  012050  012777  024000  006476        MOV   #SYNEXT!SEVEN!NOPAR!0,@PARCSR
2530  012056  052777  000020  006460        BIS   #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2531                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2532  012064  042777  020000  006466        BIC   #CLK,@TXCSR     ;POKE CLK DOWN
2533  012072  052777  020000  006460        BIS   #CLK,@TXCSR     ;POKE CLK UP
2534  012100  016703  006444                MOV   RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2535  012104  012700  000052                MOV   #52,R0  ;EXPECTED
2536  012110  012767  000007  167022        MOV   #7,SHIFT        ;# OF SHIFTS
2537  012116  012767  000052  167020        MOV   #52,TEMP1       ;DATA CHAR
2538  012124  004767  006120                JSR   PC,RPOKE        ;SHIFT IN THIS CHAR
2539  012130  105777  006410                TSTB  @RXCSR  ;RXDONE ?
2540  012134  100401                        BMI   64$
2541  012136  104000                        HLT   ;RXDONE SHOULD BE SET
2542  012140                      64$:
2543  012140  017701  006404                MOV   @RXDBUF,R1      ;ACTUAL
2544  012144  020001                        CMP   R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2545  012146  001401                        BEQ   65$
2546  012150  104002                        HLT   2            ;RECEIVED DATA DID NOT MATCH
2547                                                           ;EXPECTED DATA - CHECK MAINT DATA
2548                                                           ;OR RECEIVER LOGIC
2549  012152                      65$:
2550  012152  012767  000007  166760        MOV   #7,SHIFT        ;# OF SHIFTS
2551  012160  012767  000052  166756        MOV   #52,TEMP1       ;DATA CHAR
2552  012166  004767  006056                JSR   PC,RPOKE        ;SHIFT IN THIS CHAR
2553                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2554  012172  012767  000007  166740        MOV   #7,SHIFT        ;# OF SHIFTS
2555  012200  012767  000052  166736        MOV   #52,TEMP1       ;DATA CHAR
2556  012206  004767  006036                JSR   PC,RPOKE        ;SHIFT IN THIS CHAR
2557  012212  012700  140052                MOV   #140000!52,R0   ;EXPECTED DATA PLUS
2558                                                           ;RXERR & OVRRUN
2559  012216  017701  006326                MOV   @RXDBUF,R1      ;ACTUAL
2560  012222  020001                        CMP   R0,R1   ;COMPARE EXP VS. ACT
2561  012224  001401                        BEQ   66$
```

```
2562   012226  104002                         HLT    2             ;SPECIFICALLY LOOK AT RXERR &
2563                                                                ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2564   012230                          66$:
2565   012230  104400                         SCOPE
2566                                           ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2567                                           ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2568                                           ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2569                                           ;;(OVRRUN,RXERR)
2570                                           ;;MODE:SYNEXT
2571                                           ;;LENGTH:SEVEN
2572                                           ;;CHAR:177
2573                                           ;;
2574   012232  012767  000027  166666  TST23: MOV    #23,TSTNO     ;SAVE THIS
2575   012240  012767  012460  166650         MOV    #TST24,NEXT            ;GO TO THIS TEST WHEN THRU
2576   012246  052777  000400  006304         BIS    #MRESET,@TXCSR  ;MASTER RESET
2577   012254  012777  020000  006272         MOV    #SYNEXT,@PARCSR :SET THE MODE
2578   012262  052777  000400  006270         BIS    #MRESET,@TXCSR  ;MASTER RESET
2579
2580                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2581   012270  012777  064001  006262         MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2582
2583                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2584   012276  012777  024000  006250         MOV    #SYNEXT!SEVEN!NOPAR!0,@PARCSR
2585   012304  052777  000020  006232         BIS    #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2586                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2587   012312  042777  020000  006240         BIC    #CLK,@TXCSR     ;POKE CLK DOWN
2588   012320  052777  020000  006232         BIS    #CLK,@TXCSR     ;POKE CLK UP
2589   012326  016703  006216                 MOV    RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2590   012332  012700  000177                 MOV    #177,R0 ;EXPECTED
2591   012336  012767  000007  166574         MOV    #7,SHIFT        ;# OF SHIFTS
2592   012344  012767  000177  166572         MOV    #177,TEMP1      ;DATA CHAR
2593   012352  004767  005672                 JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2594   012356  105777  006162                 TSTB   @RXCSR  ;RXDONE ?
2595   012362  100401                         BMI    64$
2596   012364  104000                         HLT    ;RXDONE SHOULD BE SET
2597   012366                          64$:
2598   012366  017701  006156                 MOV    @RXDBUF,R1      ;ACTUAL
2599   012372  020001                         CMP    R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2600   012374  001401                         BEQ    65$
2601   012376  104002                         HLT    2             ;RECEIVED DATA DID NOT MATCH
2602                                                                ;EXPECTED DATA - CHECK MAINT DATA
2603                                                                ;OR RECEIVER LOGIC
2604   012400                          65$:
2605   012400  012767  000007  166532         MOV    #7,SHIFT        ;# OF SHIFTS
2606   012406  012767  000177  166530         MOV    #177,TEMP1      ;DATA CHAR
2607   012414  004767  005630                 JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2608                                   ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2609   012420  012767  000007  166512         MOV    #7,SHIFT        ;# OF SHIFTS
2610   012426  012767  000177  166510         MOV    #177,TEMP1      ;DATA CHAR
2611   012434  004767  005610                 JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2612   012440  012700  140177                 MOV    #140000!177,R0  ;EXPECTED DATA PLUS
2613                                                                ;RXERR & OVRRUN
2614   012444  017701  006100                 MOV    @RXDBUF,R1      ;ACTUAL
2615   012450  020001                         CMP    R0,R1   ;COMPARE EXP VS. ACT
2616   012452  001401                         BEQ    66$
2617   012454  104002                         HLT    2             ;SPECIFICALLY LOOK AT RXERR &
```

```
2618                                                ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2619  012456                          66$:
2620  012456  104400                          SCOPE
2621                                           ::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2622                                           ::RECEIVER SECTION,IT USES THE ERROR FLAGS
2623                                           ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2624                                           ::(OVRRUN,RXERR)
2625                                           ::MODE:SYNEXT
2626                                           ::LENGTH:SEVEN
2627                                           ::CHAR:0
2628                                           ::
2629  012460  012767  000030  166440  TST24:   MOV     #24,TSTNO          ;SAVE THIS
2630  012466  012767  012706  166422           MOV     #TST25,NEXT             ;GO TO THIS TEST WHEN THRU
2631  012474  052777  000400  006056           BIS     #MRESET,@TXCSR  ;MASTER RESET
2632  012502  012777  020000  006044           MOV     #SYNEXT,@PARCSR ;SET THE MODE
2633  012510  052777  000400  006042           BIS     #MRESET,@TXCSR  ;MASTER RESET
2634
2635                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2636  012516  012777  064001  006034           MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2637
2638                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2639  012524  012777  024000  006022           MOV     #SYNEXT!SEVEN!NOPAR!0,@PARCSR
2640  012532  052777  000020  006004           BIS     #SYNSCH,@RXCSR   ;SET SEARCH SYNC
2641                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2642  012540  042777  020000  006012           BIC     #CLK,@TXCSR      ;POKE CLK DOWN
2643  012546  052777  020000  006004           BIS     #CLK,@TXCSR      ;POKE CLK UP
2644  012554  016703  005770                   MOV     RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
2645  012560  012700  000000                   MOV     #0,R0    ;EXPECTED
2646  012564  012767  000007  166346           MOV     #7,SHIFT         ;# OF SHIFTS
2647  012572  012767  000000  166344           MOV     #0,TEMP1         ;DATA CHAR
2648  012600  004767  005444                   JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
2649  012604  105777  005734                   TSTB    @RXCSR   ;RXDONE ?
2650  012610  100401                           BMI     64$
2651  012612  104000                           HLT     ;RXDONE SHOULD BE SET
2652  012614                          64$:
2653  012614  017701  005730                   MOV     @RXDBUF,R1       ;ACTUAL
2654  012620  020001                           CMP     R0,R1    ;COMPARE EXPECTED VS. ACTUAL
2655  012622  001401                           BEQ     65$
2656  012624  104002                           HLT     2        ;RECEIVED DATA DID NOT MATCH
2657                                                            ;EXPECTED DATA - CHECK MAINT DATA
2658                                                            ;OR RECEIVER LOGIC
2659  012626                          65$:
2660  012626  012767  000007  166304           MOV     #7,SHIFT         ;# OF SHIFTS
2661  012634  012767  000000  166302           MOV     #0,TEMP1         ;DATA CHAR
2662  012642  004767  005402                   JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
2663                                   ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2664  012646  012767  000007  166264           MOV     #7,SHIFT         ;# OF SHIFTS
2665  012654  012767  000000  166262           MOV     #0,TEMP1         ;DATA CHAR
2666  012662  004767  005362                   JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
2667  012666  012700  140000                   MOV     #140000!0,R0     ;EXPECTED DATA PLUS
2668                                                            ;RXERR & OVRRUN
2669  012672  017701  005652                   MOV     @RXDBUF,R1       ;ACTUAL
2670  012676  020001                           CMP     R0,R1    ;COMPARE EXP VS. ACT
2671  012700  001401                           BEQ     66$
2672  012702  104002                           HLT     2        ;SPECIFICALLY LOOK AT RXERR &
2673                                                            ;OVRRUN BITS...THEY BOTH SHOULD BE SET
```

```
2674  012704                              66$:
2675  012704  104400                             SCOPE
2676                                              ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2677                                              ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2678                                              ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2679                                              ;;(OVRRUN,RXERR)
2680                                              ;;MODE:SYNEXT
2681                                              ;;LENGTH:EIGHT
2682                                              ;;CHAR:125
2683                                              ;;
2684  012706  012767  000031  166212     TST25:  MOV     #25,TSTNO        ;SAVE THIS
2685  012714  012767  013134  166174             MOV     #TST26,NEXT               ;GO TO THIS TEST WHEN THRU
2686  012722  052777  000400  005630             BIS     #MRESET,@TXCSR   ;MASTER RESET
2687  012730  012777  020000  005616             MOV     #SYNEXT,@PARCSR  ;SET THE MODE
2688  012736  052777  000400  005614             BIS     #MRESET,@TXCSR   ;MASTER RESET
2689
2690                                      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2691  012744  012777  064001  005606             MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2692
2693                                      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2694  012752  012777  026000  005574             MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
2695  012760  052777  000020  005556             BIS     #SYNSCH,@RXCSR   ;SET SEARCH SYNC
2696                                      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2697  012766  042777  020000  005564             BIC     #CLK,@TXCSR      ;POKE CLK DOWN
2698  012774  052777  020000  005556             BIS     #CLK,@TXCSR      ;POKE CLK UP
2699  013002  016703  005542                     MOV     RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
2700  013006  012700  000125                     MOV     #125,R0 ;EXPECTED
2701  013012  012767  000010  166120             MOV     #8.,SHIFT        ;# OF SHIFTS
2702  013020  012767  000125  166116             MOV     #125,TEMP1       ;DATA CHAR
2703  013026  004767  005216                     JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
2704  013032  105777  005506                     TSTB    @RXCSR  ;RXDONE ?
2705  013036  100401                             BMI     64$
2706  013040  104000                             HLT     ;RXDONE SHOULD BE SET
2707  013042                              64$:
2708  013042  017701  005502                     MOV     @RXDBUF,R1       ;ACTUAL
2709  013046  020001                             CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2710  013050  001401                             BEQ     65$
2711  013052  104002                             HLT     2               ;RECEIVED DATA DID NOT MATCH
2712                                                                      ;EXPECTED DATA - CHECK MAINT DATA
2713                                                                      ;OR RECEIVER LOGIC
2714  013054                              65$:
2715  013054  012767  000010  166056             MOV     #8.,SHIFT        ;# OF SHIFTS
2716  013062  012767  000125  166054             MOV     #125,TEMP1       ;DATA CHAR
2717  013070  004767  005154                     JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
2718                                      ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2719  013074  012767  000010  166036             MOV     #8.,SHIFT        ;# OF SHIFTS
2720  013102  012767  000125  166034             MOV     #125,TEMP1       ;DATA CHAR
2721  013110  004767  005134                     JSR     PC,RPOKE         ;SHIFT IN THIS CHAR
2722  013114  012700  140125                     MOV     #140000!125,R0   ;EXPECTED DATA PLUS
2723                                                                      ;RXERR & OVRRUN
2724  013120  017701  005424                     MOV     @RXDBUF,R1       ;ACTUAL
2725  013124  029001                             CMP     R0,R1   ;COMPARE EXP VS. ACT
2726  013126  001401                             BEQ     66$
2727  013130  104002                             HLT     2               ;SPECIFICALLY LOOK AT RXERR &
2728                                                                      ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2729  013132                              66$:
```

```
2730   013132   104400                        SCOPE
2731                                          ::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2732                                          ::RECEIVER SECTION,IT USES THE ERROR FLAGS
2733                                          ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2734                                          :: (OVRRUN,RXERR)
2735                                          ::MODE:SYNEXT
2736                                          ::LENGTH:EIGHT
2737                                          ::CHAR:252
2738                                          ::
2739   013134   012767   000032   165764  TST26:  MOV    #26,TSTNO        ;SAVE THIS
2740   013142   012767   013362   165746       MOV    #TST27,NEXT             ;GO TO THIS TEST WHEN THRU
2741   013150   052777   000400   005402       BIS    #MRESET,@TXCSR  ;MASTER RESET
2742   013156   012777   020000   005370       MOV    #SYNEXT,@PARCSR ;SET THE MODE
2743   013164   052777   000400   005366       BIS    #MRESET,@TXCSR  ;MASTER RESET
2744
2745                                       ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2746   013172   012777   064001   005360       MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2747
2748                                       ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2749   013200   012777   026000   005346       MOV    #SYNEXT!EIGHT!NOPAR!0,@PARCSR
2750   013206   052777   000020   005330       BIS    #SYNSCH,@RXCSR   ;SET SEARCH SYNC
2751                                       ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2752   013214   042777   020000   005336       BIC    #CLK,@TXCSR      ;POKE CLK DOWN
2753   013222   052777   020000   005330       BIS    #CLK,@TXCSR      ;POKE CLK UP
2754   013230   016703   005314                MOV    RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
2755   013234   012700   000252                MOV    #252,R0 ;EXPECTED
2756   013240   012767   000010   165672       MOV    #8.,SHIFT        ;# OF SHIFTS
2757   013246   012767   000252   165670       MOV    #252,TEMP1       ;DATA CHAR
2758   013254   004767   004770                JSR    PC,RPOKE         ;SHIFT IN THIS CHAR
2759   013260   105777   005260                TSTB   @RXCSR  ;RXDONE ?
2760   013264   100401                         BMI    64$
2761   013266   104000                         HLT    ;RXDONE SHOULD BE SET
2762   013270                           64$:
2763   013270   017701   005254                MOV    @RXDBUF,R1       ;ACTUAL
2764   013274   020001                         CMP    R0,R1    ;COMPARE EXPECTED VS. ACTUAL
2765   013276   001401                         BEQ    65$
2766   013300   104002                         HLT    2        ;RECEIVED DATA DID NOT MATCH
2767                                                           ;EXPECTED DATA - CHECK MAINT DATA
2768                                                           ;OR RECEIVER LOGIC
2769   013302                           65$:
2770   013302   012767   000010   165630       MOV    #8.,SHIFT        ;# OF SHIFTS
2771   013310   012767   000252   165626       MOV    #252,TEMP1       ;DATA CHAR
2772   013316   004767   004726                JSR    PC,RPOKE         ;SHIFT IN THIS CHAR
2773                                       ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2774   013322   012767   000010   165610       MOV    #8.,SHIFT        ;# OF SHIFTS
2775   013330   012767   000252   165606       MOV    #252,TEMP1       ;DATA CHAR
2776   013336   004767   004706                JSR    PC,RPOKE         ;SHIFT IN THIS CHAR
2777   013342   012700   140252                MOV    #140000!252,R0   ;EXPECTED DATA PLUS
2778                                                           ;RXERR & OVRRUN
2779   013346   017701   005176                MOV    @RXDBUF,R1       ;ACTUAL
2780   013352   020001                         CMP    R0,R1    ;COMPARE EXP VS. ACT
2781   013354   001401                         BEQ    66$
2782   013356   104002                         HLT    2        ;SPECIFICALLY LOOK AT RXERR &
2783                                                           ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2784   013360                           66$:
2785   013360   104400                         SCOPE
```

```
2786                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2787                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2788                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2789                                          ;;(OVRRUN,RXERR)
2790                                          ;;MODE:SYNEXT
2791                                          ;;LENGTH:EIGHT
2792                                          ;;CHAR:377
2793                                          ;;
2794   013362  012767  000033  165536  TST27:  MOV     #27,TSTNO          ;SAVE THIS
2795   013370  012767  013610  165520          MOV     #TST28,NEXT            ;GO TO THIS TEST WHEN THRU
2796   013376  052777  000400  005154          BIS     #MRESET,@TXCSR  ;MASTER RESET
2797   013404  012777  020000  005142          MOV     #SYNEXT,@PARCSR ;SET THE MODE
2798   013412  052777  000400  005140          BIS     #MRESET,@TXCSR  ;MASTER RESET
2799
2800                                          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2801   013420  012777  064001  005132          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2802
2803                                          ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2804   013426  012777  026000  005120          MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
2805   013434  052777  000020  005102          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2806                                          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2807   013442  042777  020000  005110          BIC     #CLK,@TXCSR       ;POKE CLK DOWN
2808   013450  052777  020000  005102          BIS     #CLK,@TXCSR       ;POKE CLK UP
2809   013456  016703  005066                  MOV     RXDBUF,R3         ;SET UP FOR ERROR MESSAGE
2810   013462  012700  000377                  MOV     #377,R0 ;EXPECTED
2811   013466  012767  000010  165444          MOV     #8.,SHIFT         ;# OF SHIFTS
2812   013474  012767  000377  165442          MOV     #377,TEMP1        ;DATA CHAR
2813   013502  004767  004542                  JSR     PC,RPOKE          ;SHIFT IN THIS CHAR
2814   013506  105767  005032                  TSTB    @RXCSR  ;RXDONE ?
2815   013512  100401                          BMI     64$
2816   013514  104000                          HLT             ;RXDONE SHOULD BE SET
2817   013516                          64$:
2818   013516  017701  005026                  MOV     @RXDBUF,R1        ;ACTUAL
2819   013522  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2820   013524  001401                          BEQ     65$
2821   013526  104002                          HLT     2         ;RECEIVED DATA DID NOT MATCH
2822                                                            ;EXPECTED DATA - CHECK MAINT DATA
2823                                                            ;OR RECEIVER LOGIC
2824   013530                          65$:
2825   013530  012767  000010  165402          MOV     #8.,SHIFT         ;# OF SHIFTS
2826   013536  012767  000377  165400          MOV     #377,TEMP1        ;DATA CHAR
2827   013544  004767  004500                  JSR     PC,RPOKE          ;SHIFT IN THIS CHAR
2828                                          ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2829   013550  012767  000010  165362          MOV     #8.,SHIFT         ;# OF SHIFTS
2830   013556  012767  000377  165360          MOV     #377,TEMP1        ;DATA CHAR
2831   013564  004767  004460                  JSR     PC,RPOKE          ;SHIFT IN THIS CHAR
2832   013570  012700  140377                  MOV     #140000!377,R0    ;EXPECTED DATA PLUS
2833                                                            ;RXERR & OVRRUN
2834   013574  017701  004750                  MOV     @RXDBUF,R1        ;ACTUAL
2835   013600  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
2836   013602  001401                          BEQ     66$
2837   013604  104002                          HLT     2         ;SPECIFICALLY LOOK AT RXERR &
2838                                                            ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2839   013606                          66$:
2840   013606  104400                          SCOPE
2841                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
```

```
2842                                        ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2843                                        ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2844                                        ;; (OVRRUN,RXERR)
2845                                        ;;MODE:SYNEXT
2846                                        ;;LENGTH:EIGHT
2847                                        ;;CHAR:0
2848                                        ;;
2849    013610  012767  000034  165310  TST28:  MOV    #28,TSTNO         ;SAVE THIS
2850    013616  012767  014036  165272          MOV    #.EOP,NEXT               ;GO TO THIS TEST WHEN THRU
2851    013624  052777  000400  004726          BIS    #MRESET,@TXCSR   ;MASTER RESET
2852    013632  012777  020000  004714          MOV    #SYNEXT,@PARCSR  ;SET THE MODE
2853    013640  052777  000400  004712          BIS    #MRESET,@TXCSR   ;MASTER RESET
2854
2855                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2856    013646  012777  064001  004704          MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2857
2858                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2859    013654  012777  026000  004672          MOV    #SYNEXT!EIGHT!NOPAR!0,@PARCSR
2860    013662  052777  000020  004654          BIS    #SYNSCH,@RXCSR   ;SET SEARCH SYNC
2861                                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2862    013670  042777  020000  004662          BIC    #CLK,@TXCSR       ;POKE CLK DOWN
2863    013676  052777  020000  004654          BIS    #CLK,@TXCSR       ;POKE CLK UP
2864    013704  016703  004640                  MOV    RXDBUF,R3         ;SET UP FOR ERROR MESSAGE
2865    013710  012700  000000                  MOV    #0,R0    ;EXPECTED
2866    013714  012767  000010  165216          MOV    #8.,SHIFT         ;# OF SHIFTS
2867    013722  012767  000000  165214          MOV    #0,TEMP1          ;DATA CHAR
2868    013730  004767  004314                  JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
2869    013734  105777  004604                  TSTB   @RXCSR  ;RXDONE ?
2870    013740  100401                          BMI    64$
2871    013742  104000                          HLT       ;RXDONE SHOULD BE SET
2872    013744                          64$:
2873    013744  017701  004600                  MOV    @RXDBUF,R1       ;ACTUAL
2874    013750  020001                          CMP    R0,R1    ;COMPARE EXPECTED VS. ACTUAL
2875    013752  001401                          BEQ    65$
2876    013754  104002                          HLT    2         ;RECEIVED DATA DID NOT MATCH
2877                                                             ;EXPECTED DATA - CHECK MAINT DATA
2878                                                             ;OR RECEIVER LOGIC
2879    013756                          65$:
2880    013756  012767  000010  165154          MOV    #8.,SHIFT         ;# OF SHIFTS
2881    013764  012767  000000  165152          MOV    #0,TEMP1          ;DATA CHAR
2882    013772  004767  004252                  JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
2883                                        ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2884    013776  012767  000010  165134          MOV    #8.,SHIFT         ;# OF SHIFTS
2885    014004  012767  000000  165132          MOV    #0,TEMP1          ;DATA CHAR
2886    014012  004767  004232                  JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
2887    014016  012700  140000                  MOV    #140000!0,R0     ;EXPECTED DATA PLUS
2888                                                             ;RXERR & OVRRUN
2889    014022  017701  004522                  MOV    @RXDBUF,R1       ;ACTUAL
2890    014026  020001                          CMP    R0,R1    ;COMPARE EXP VS. ACT
2891    014030  001401                          BEQ    66$
2892    014032  104002                          HLT    2         ;SPECIFICALLY LOOK AT RXERR &
2893                                                             ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2894    014034                          66$:
2895    014034  104400                          SCOPE
```

```
2896
2897                                           ;END OF PASS
2898                                           ;TYPE NAME OF TEST
2899                                           ;UPDATE PASS COUNT
2900                                           ;CHECK FOR EXIT TO ACT-11
2901                                           ;RESTART TEST
2902
2903   014036  104402              .EOP:   TYPE                         ;TYPE NAME OF TEST
2904   014040  017200                      MEPASS
2905   014042  104410  014274              CONVRT   .OUTCRY
2906   014046  104402  016721              TYPE     ,DEVICE
2907   014052  105767  165124              TSTB     MULTD-              ;ARE YOU RUNNING MULTIPLE DEVICES ?
2908   014056  001511                      BEQ      CCC                 ;NO,JUMP AROUND
2909   014060  005767  165132              TST      ACTREG              ;ARE ANY DEVICES ACTIVE ?
2910   014064  001007                      BNE      RUNIT               ;YES
2911   014066  104402  016733              TYPE     ,MCOW               ;NO
2912   014072  016700  165120              MOV      ACTREG,RO           ;DISPLAY ACTREG
2913   014076  000000                      HALT                         ;SELECT SOMETHING TO RUN @ ACTREG:
2914                                           ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2915   014100  000167  165314              JMP      .START              ;START OVER AGAIN......YOU DESELECTED EVERYTHING
2916   014104  062767  000010  165072 RUNIT: ADD    #10,BASEADD         ;NEXT BLOCK (ADDRESSES)
2917   014112  062767  000010  165072 ZERO:  ADD    #10,BASEIV          ;NEXT BLOCK (VECTORS)
2918   014120  000241                      CLC
2919   014122  006167  165072              ROL      ROTADD              ;UP DATE ROTATING POINTER
2920   014126  103410                      BCS      2$                  ;IS IT THE LAST DEVICE
2921                                                                    ;TO BE TESTED IN THIS PASS ?
2922   014130  036767  165064  165060       BIT     ROTADD,ACTREG       ;TEST THIS DEVICE FOR ACTIVE STATUS
2923   014136  001762                      BEQ      RUNIT               ;IF NOT ACTIVE, TRY NEXT ADDRESS
2924   014140  004767  000034              JSR      PC,REPLAY           ;CALCULATE NEW PARAMETERS
2925   014144  000167  000174              JMP      RESTRT              ;YES IT WAS ACTIVE,TEST THIS DEVICE
2926   014150  012767  000001  165042 2$:   MOV     #1,ROTADD           ;OK!,NOW SET UP  ROTATING
2927                                                                    ;POINTER FOR NEXT MULTIPLE PASS
2928   014156  016767  165024  165020       MOV     KEEPADD,BASEADD     ;RESTORE BASE ADDRESS
2929   014164  016767  165024  165020       MOV     KEEPIV,BASEIV       ;RESTORE BASE INTERRUPT VECTORS
2930   014172  004767  000002              JSR      PC,REPLAY           ;CALC NEW PARAMETERS
2931   014176  000441                      BR       CCC                 ;JUMP AROUND REPLAY
2932   014200  016767  165000  004040 REPLAY: MOV   BASEADD,DUBASE      ;SET UP FOR NEW ADDRESSES
2933   014206  004767  003702              JSR      PC,DUADDR           ;CREATE NEW ADDRESSES
2934   014212  016767  164774  004350       MOV     BASEIV,DURIV        ;CREATE DURIV
2935   014220  062767  000002  164764       ADD     #2,BASEIV
2936   014226  016767  164760  004336       MOV     BASEIV,DURIS        ;CREATE DURIS
2937   014234  062767  000002  164750       ADD     #2,BASEIV
2938   014242  016767  164744  004324       MOV     BASEIV,DUTIV        ;CREATE DUTIV
2939   014250  062767  000002  164734       ADD     #2,BASEIV
2940   014256  016767  164730  004312       MOV     BASEIV,DUTIS        ;CREATE DUTIS
2941   014264  016767  004300  164720       MOV     DURIV,BASEIV        ;RESTORE
2942   014272  000207                      RTS      PC
2943
2944   014274  000001              OUTCRY: 1
2945   014276     006     002              .BYTE    6,2
2946   014300  020544                      RXCSR
2947
2948   014302                      CCC:
2949   014302  005067  164626              CLR      LSTERR              ;CLEAR LAST ERROR PC
2950   014306  005067  164734              CLR      ERRFLG              ;CLEAR ERROR FLAG
2951   014312  005267  164612              INC      PASCNT              ;UPDATE PASS COUNT
```

```
2952  014316  016777  164606  164556         MOV    PASCNT,@LIGHTS           ;DISPLAY PASS COUNT
2953  014324  013701  000042                 MOV    @#42,R1                  ;CHECK FOR ACT-11 OR DDP
2954  014330  001405                          BEQ    RESTRT                   ;IF NOT, CONTINUE TESTING
2955  014332  000005                          RESET
2956  014334  004711                 LOGICAL:  JSR    PC,(R1)
2957  014336  000240                          NOP
2958  014340  000240                          NOP
2959  014342  000240                          NOP
2960  014344  012767  000340  163424  RESTRT: MOV    #340,PS                  ;PREVENT INTERRUPTS (PRIO: 7)
2961  014352  104413                          CKSWR                           ;CHECK FOR ^G
2962  014354  012767  003446  164532         MOV    #TST1,RTRN
2963  014362  000167  167060                 JMP    TST1
2964
2965                                           ;SCOPE LOOP AND INTERATION HANDLER
2966
2967  014366                         .SCOPE:
2968                                  ;**** START OF CODE FOR THE X OR TESTER ******
2969  014366  000424                          BR     4$                       ;IF RUNNING ON THE X OR TESTER CHANGE
2970                                                                           ;THIS INSTRUCTION TO A 'NOP''(NOP=240)
2971  014370  013746  000004                 MOV    @#4,-(SP)                ;SAVE CONTENTS OF ERROR VECTOR
2972  014374  012737  014414  000004         MOV    #1$,@#4                  ;SET FOR TIME OUT
2973  014402  005737  177060                 TST    @#177060                 ;TIME OUT ON  X OR ?
2974  014406  012637  000004                 MOV    (SP)+,@#4                ;RESTORE ERROR VECTOR
2975  014412  000404                          BR     2$                       ;GO TO NEXT TEST
2976  014414  022626                 1$:      CMP    (SP)+,(SP)+              ;CLEAR THE STACK AFTER A TIMEOUT
2977  014416  012637  000004                 MOV    (SP)+,@#4                ;RESTORE ERROR VECTOR
2978  014422  000403                          BR     3$                       ;LOOP ON PRESENT TEST
2979  014424  016767  164466  164462  2$:     MOV    NEXT,RTRN                ;SET UP NEXT TEST IN RTRN
2980  014432  016716  164456         3$:      MOV    RTRN,(SP)                ;SET UP STACK FOR RTI
2981  014436  000002                          RTI
2982  014440                         4$:      ;**** END OF CODE FOR THE  X OR TESTER ******
2983  014440  104413                          CKSWR                           ;CHECK FOR ^G
2984  014442  032777  040000  164430         BIT    #SW14,@SWR               ;LOOP ON CURRENT TEST ?
2985  014450  001407                 TTST:    BEQ    1$
2986  014452  000432                          BR     3$
2987  014454  105777  164424                 TSTB   @TKCSR                   ;TEST TTY FLAG
2988  014460  100027                          BPL    3$
2989  014462  017700  164420                 MOV    @TKDBR,R0                ;CLR DONE BIT
2990  014466  000412                          BR     2$                       ;IF A TTY KEY IS STRUCK GO TO NEXT TST
2991  014470  032777  004000  164402  1$:     BIT    #SW11,@SWR               ;INHIBIT ITERATIONS ?
2992  014476  001006                          BNE    2$
2993  014500  005267  164420                 INC    LPCNT
2994  014504  026767  164414  164410         CMP    LPCNT,ICOUNT             ;CHECK FOR ITERATION CNT FINISH
2995  014512  101412                          BLOS   3$
2996  014514  105067  164526         2$:      CLRB   ERRFLG
2997  014520  005067  164400                 CLR    LPCNT
2998  014524  012767  000005  163370         MOV    #5,ICOUNT                ;SET UP ITERATION COUNT
2999  014532  016767  164360  164354         MOV    NEXT,RTRN                ;SET UP NEXT TEST IN RTRN
3000  014540  016716  164350         3$:      MOV    RTRN,(SP)                ;SET UP STACK FOR RTI
3001  014544  000002                          RTI
3002  014546  001407                 BRW:     140''                          ;RESTORE 'BEQ 1$'' INSTRUCTION
3003  014550  000432                 BRX:     432                            ;RESTORE 'BR 3$'' INSTRUCTION
3004
3005                                           ;CHECK FOR FREEZE ON CURRENT DATA
3006
3007  014552  104413                 .SCOP1: CKSWR                           ;CHECK  FOR ^G
```

G 5

```
3008  014554  032777  001000  164316        BIT     #SW09,@SWR
3009  014562  001402                         BEQ     1$
3010  014564  016716  164330                 MOV     LOCK,(SP)
3011  014570  000002              1$:        RTI
3012
3013                              ;TELETYPE OUTPUT ROUTINE
3014
3015  014572  010546              .TYPE:     MOV     R5,-(SP)
3016  014574  017605  000002                 MOV     @2(SP),R5
3017  014600  062766  000002 000002          ADD     #2,2(SP)
3018  014606  105715              1$:        TSTB    (R5)              ;LOOK FOR '0'
3019  014610  001406                         BEQ     3$
3020  014612  105777  164272      2$:        TSTB    @TPCSR            ;TEST DONE BIT
3021  014616  100375                         BPL     2$
3022  014620  112577  164266                 MOVB    (R5)+,@TPDBR      ;TYPE CHAR
3023  014624  000770                         BR      1$                ;DO IT AGAIN UNTIL '0' IS SEEN
3024  014626  012605              3$:        MOV     (SP)+,R5
3025  014630  000002                         RTI
3026
3027                              ;ASCII STRING INPUT ROUTINE
3028
3029  014632  010346              .INSTR:    MOV     R3,-(SP)
3030  014634  010446                         MOV     R4,-(SP)
3031  014636  017667  000004 000010          MCV     @4(SP),.MSG
3032  014644  062766  000002 000004          ADD     #2,4(SP)
3033  014652  104402              .INST1:    TYPE
3034  014654  000000              .MSG:      0
3035  014656  012704  017700                 MOV     #INBUF,R4
3036  014662  012703  000007                 MOV     #7,R3
3037  014666  105777  164212      1$:        TSTB    @TKCSR
3038  014672  100375                         BPL     1$
3039  014674  117714  164206                 MOVB    @TKDBR,(R4)
3040  014700  142714  000200                 BICB    #200,(R4)
3041  014704  121427  000025                 CMPB    (R4),#25          ;IS IT <^U>
3042  014710  001003                         BNE     200$
3043  014712  104402  017110                 TYPE,MCRLF
3044  014716  000755                         BR      .INST1
3045  014720  122427  000015      200$:      CMPB    (R4)+,#15
3046  014724  001423                         BEQ     INSTR2
3047  014726  117777  164154 164156          MOVB    @TKDBR,@TPDBR
3048  014734  105777  164150      2$:        TSTB    @TPCSR
3049  014740  100375                         BPL     2$
3050  014742  005303                         DEC     R3
3051  014744  001350                         BNE     1$
3052  014746  000402                         BR      .INSTG
3053  014750  010346              .INSTE:    MOV     R3,-(SP)
3054  014752  010446                         MOV     R4,-(SP)
3055  014754  104402              .INSTG:    TYPE
3056  014756  017104                         MQM
3057  014760  005737  016246                 TST     @#RDSW
3058  014764  001402                         BEQ     400$
3059  014766  104402  017110                 TYPE,MCRLF
3060  014772  000727              400$:      BR      .INST1
3061  014774  012604              INSTR2:    MOV     (SP)+,R4
3062  014776  012603                         MOV     (SP)+,R3
3063  015000  000002                         RTI
```

```
3064
3065                                      ;CONVERT ASCII STRING TO OCTAL
3066
3067   015002  010546        .PARAM: MOV     R5,-(SP)
3068   015004  010446                MOV     R4,-(SP)
3069   015006  016605  000004        MOV     4(SP),R5
3070   015012  012567  000170        MOV     (R5)+,LOLIM
3071   015016  012567  000166        MOV     (R5)+,HILIM
3072   015022  012567  000164        MOV     (R5)+,DEVADR
3073   015026  112567  000162        MOVB    (R5)+,LOBITS
3074   015032  112567  000157        MOVB    (R5)+,ADRCNT
3075   015036  010566  000004        MOV     R5,4(SP)
3076   015042  005005        PARAM1: CLR     R5
3077   015044  012704  017700        MOV     #INBUF,R4
3078   015050  122714  000015        CMPB    #15,(R4)
3079   015054  001420                BEQ     PARERR
3080   015056  121427  000060    1$: CMPB    (R4),#60
3081   015062  002415                BLT     PARERR
3082   015064  121427  000067        CMPB    (R4),#67
3083   015070  003012                BGT     PARERR
3084   015072  142714  000060        BICB    #60,(R4)
3085   015076  152405                BISB    (R4)+,R5
3086   015100  122714  000015        CMPB    #15,(R4)
3087   015104  001414                BEQ     LIMITS
3088   015106  006305                ASL     R5
3089   015110  006305                ASL     R5
3090   015112  006305                ASL     R5
3091   015114  000760                BR      1$
3092   015116  122714  000015  PARERR: CMPB  #15,(R4)         ;IS FIRST CHARACTER A <CR>
3093   015122  001003                BNE     120$
3094   015124  005737  016246        TST     @#RDSW           ;IS CKSWR ROUTINE BEING USED
3095   015130  001023                BNE     PARTI
3096   015132  104404        120$:   INSTER
3097   015134  000742                BR      PARAM1
3098
3099                                  ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
3100
3101   015136  020567  000046  LIMITS: CMP   R5,HILIM
3102   015142  101365                BHI     PARERR
3103   015144  020567  000036        CMP     R5,LOLIM
3104   015150  103762                BLO     PARERR
3105   015152  136705  000036        BITB    LOBITS,R5
3106   015156  001357                BNE     PARERR
3107
3108                                  ;STORE NUMBER AT SPECIFIED ADDRESS
3109
3110   015160  016704  000026        MOV     DEVADR,R4
3111   015164  010524        1$:     MOV     R5,(R4)+
3112   015166  062705  000002        ADD     #2,R5
3113   015172  105367  000017        DECB    ADRCNT
3114   015176  001372                BNE     1$
3115   015200  012604        PARTI:  MOV     (SP)+,R4
3116   015202  012605                MOV     (SP)+,R5
3117   015204  000002                RTI
3118   015206  000000        LOLIM:  0
3119   015210  000000        HILIM:  0
```

```
3120   015212   000000                        DEVADR:  0
3121   015214   000000                        LOBITS:  0
3122            015215                         ADRCNT=LOBITS+1
3123
3124                                           ;SAVE PC OF TEST THAT FAILED AND R0-R5
3125
3126   015216   016667   000004   163750       .SAV05:  MOV     4(SP),SAVPC
3127
3128                                           ;SAVE R0-R5
3129
3130   015224   010567   163740               SV05:     MOV     R5,SAVR5
3131   015230   010467   163732                         MOV     R4,SAVR4
3132   015234   010367   163724                         MOV     R3,SAVR3
3133   015240   010267   163716                         MOV     R2,SAVR2
3134   015244   010167   163710                         MOV     R1,SAVR1
3135   015250   010067   163702                         MOV     R0,SAVR0
3136   015254   000002                                  RTI
3137
3138                                           ;RESTORE R0-R5
3139
3140   015256   016700   163674               .RES05:   MOV     SAVR0,R0
3141   015262   016701   163672                         MOV     SAVR1,R1
3142   015266   016702   163670                         MOV     SAVR2,R2
3143   015272   016703   163666                         MOV     SAVR3,R3
3144   015276   016704   163664                         MOV     SAVR4,R4
3145   015302   016705   163662                         MOV     SAVR5,R5
3146   015306   000002                                  RTI
3147
3148                                           ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3149
3150   015310   104402                         .CONVR:  TYPE
3151   015312   017110                                  MCRLF
3152   015314   010046                         .CNVRT:  MOV     R0,-(SP)
3153   015316   010146                                  MOV     R1,-(SP)
3154   015320   010346                                  MOV     R3,-(SP)
3155   015322   010446                                  MOV     R4,-(SP)
3156   015324   010546                                  MOV     R5,-(SP)
3157   015326   017601   000012                         MOV     @12(SP),R1
3158   015332   016767   002402   163610                MOV     TEMP,TEMP3
3159   015340   062766   000002   000012                ADD     #2,12(SP)
3160   015346   012167   000154                         MOV     (R1)+,WRDCNT
3161   015352   112167   000152               1$:       MOVB    (R1)+,CHRCNT
3162   015356   112167   000147                         MOVB    (R1)+,SPACNT
3163   015362   013167   000144                         MOV     @(R1)+,BINWRD
3164   015366   016704   000140               2$:       MOV     BINWRD,R4
3165   015372   116705   000132                         MOVB    CHRCNT,R5
3166   015376   012700   017740                         MOV     #TEMP,R0
3167   015402   010403                         3$:       MOV     R4,R3
3168   015404   042703   177770                         BIC     #177770,R3
3169   015410   062703   000060                         ADD     #060,R3
3170   015414   110320                                  MOVB    R3,(R0)+
3171   015416   006204                                  ASR     R4              ;SHIFT FOR NEXT #
3172   015420   042704   100000                         BIC     #100000,R4      ;CLUGE TO STOP BIT 15 PROPAGATING.
3173   015424   006204                                  ASR     R4              ;DITTO
3174   015426   006204                                  ASR     R4              ;DITTO
3175   015430   005305                                  DEC     R5
```

```
3176  015432  001363                   BNE     3$
3177  015434  012703  020000           MOV     #MDATA,R3
3178  015440  114023            4$:    MOVB    -(R0),(R3)+
3179  015442  105367  000062           DECB    CHRCNT
3180  015446  001374                   BNE     4$
3181  015450  105767  000055           TSTB    SPACNT
3182  015454  001405                   BEQ     6$
3183  015456  112723  000040    5$:    MOVB    #040,(R3)+
3184  015462  105367  000043           DECB    SPACNT
3185  015466  001373                   BNE     5$
3186  015470  105013            6$:    CLRB    (R3)
3187  015472  104402                   TYPE
3188  015474  020000                   MDATA
3189  015476  005367  000024           DEC     WRDCNT
3190  015502  001323                   BNE     1$
3191  015504  016767  163440  002226   MOV     TEMP3,TEMP
3192  015512  012605                   MOV     (SP)+,R5
3193  015514  012604                   MOV     (SP)+,R4
3194  015516  012603                   MOV     (SP)+,R3
3195  015520  012601                   MOV     (SP)+,R1
3196  015522  012600                   MOV     (SP)+,R0
3197  015524  000002                   RTI
3198  015526  000000           WRDCNT: 0
3199  015530  000000           CHRCNT: 0
3200          015531           SPACNT=CHRCNT+1
3201  015532  000000           BINWRD: 0
3202
3203                                   ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
3204                                   ;BUFFER TO THE CHARACTERS 'N' AND 'Y'.
3205                                   ; IF THE CHARACTER IS 'N' CLEAR THE FLAG
3206                                   ; IF THE CHARACTER IS 'Y' SET THE FLAG
3207
3208  015534  017605  000000   .SETFLG:MOV     @(SP),R5
3209  015540  122767  000116  002132   CMPB    #'N,INBUF        ;IS IT 'N' ?
3210  015546  001002                   BNE     1$
3211  015550  105015                   CLRB    (R5)     ;000
3212  015552  000406                   BR      2$
3213  015554  122767  000131  002116  1$:  CMPB    #'Y,INBUF             ;IS IT 'Y' ?
3214  015562  001005                   BNE     3$
3215  015564  112715  177777           MOVB    #-1,(R5)              ;377
3216  015570  062716  000002   2$:     ADD     #2,(SP)
3217  015574  000002                   RTI
3218  015576  104404           3$:     INSTER                    ;RETRY
3219  015600  000755                   BR      .SETFLG
3220                                   ; TRAP DISPATCH SERVICE
3221                                   ;ARGUMENT OF TRAP IS EXTRACTED
3222                                   ;AND USED AS OFFSET TO OBTAIN POINTER
3223                                   ;TO SELECTED SUBROUTINE
3224
3225  015602  011646           .TRPSR: MOV     (SP),-(SP)            ;GET PC OF RETURN
3226  015604  162716  000002           SUB     #2,(SP)               ;=PC OF TRAP
3227  015610  017616  000000           MOV     @(SP),(SP)            ;GET TRP
3228  015614  006316           TRPOK:  ASL     (SP)                  ;MULTIPLY TRAP ARG BY 2
3229  015616  042716  177001           BIC     #177001,(SP)          ;CLEAR UNWANTED BITS
3230  015622  062716  001366           ADD     #.TRPTAB,(SP)         ;POINTER TO SUBROUTINE ADDRESS
3231  015626  017616  000000           MOV     @(SP),(SP)            ;SUBROUTINE ADDRESS
```

```
3232   015632  000136                          JMP      @(SP)+            ;GO TO SUBROUTINE
3233
3234                                            ;ERROR HANDLER
3235
3236   015634  104413                  .HLT:   CKSWR                      ;CHECK FOR ^G
3237   015636  032777  020000  163234           BIT      #SW13,@SWR        ;INHIBIT ERROR TYPE OUT ?
3238   015644  001061                          BNE      HALTS
3239   015646  021667  163262                  CMP      (SP),LSTERR
3240   015652  001404                          BEQ      1$
3241   015654  011667  163254                  MOV      (SP),LSTERR
3242   015660  105067  163362                  CLRB     ERRFLG
3243   015664  104406                  1$:     SAV05
3244   015666  011605                          MOV      (SP),R5
3245   015670  162705  000002                  SUB      #2,R5
3246   015674  011504                          MOV      (R5),R4
3247   015676  006304                          ASL      R4
3248   015700  061504                          ADD      (R5),R4
3249   015702  006304                          ASL      R4
3250   015704  042704  177001                  BIC      #177001,R4
3251   015710  062704  020514                  ADD      #.ERRTAB,R4
3252   015714  012467  000040                  MOV      (R4)+,ERRMSG
3253   015720  012467  000046                  MOV      (R4)+,DATAHD
3254   015724  011467  000054                  MOV      (R4),DATABP
3255   015730  105767  163312                  TSTB     ERRFLG
3256   015734  001403                          BEQ      TYPMSG
3257   015736  005767  000042                  TST      DATABP
3258   015742  001014                          BNE      TYPDAT
3259   015744  104410                  TYPMSG: CONVRT
3260   015746  016076                          ERTAB0
3261   015750  112767  177777  163270           MOVB     #-1,ERRFLG
3262   015756  104402                          TYPE
3263   015760  000000                  ERRMSG: 0
3264   015762  005767  000004                  TST      DATAHD
3265   015766  001402                          BEQ      TYPDAT
3266   015770  104402                          TYPE
3267   015772  000000                  DATAHD: 0
3268   015774  005767  000004                  TYPDAT: TST      DATABP
3269   016000  001402                          BEQ      RESREG
3270   016002  104410                          CONVRT
3271   016004  000000                  DATABP: 0
3272   016006  104407                  RESREG: RES05
3273   016010  005777  163064                  HALTS:  TST      @SWR
3274   016014  100005                          BPL      EXITER
3275   016016  010046                          PUSHR0
3276   016020  016600  000002                  MOV      2(SP),R0
3277   016024  000000                          HALT
3278   016026  012600                          POPR0
3279   016030  104413                  EXITER: CKSWR                      ;CHECK FOR ^G
3280   016032  005267  163074                  INC      ERRCNT
3281   016036  032777  000400  163034           BIT      #SW08,@SWR        ;LOOP ON ERROR ?
3282   016044  001007                          BNE      1$
3283   016046  032777  002000  163024           BIT      #SW10,@SWR        ;ESCAPE TO NEXT ON ERROR ?
3284   016054  001407                          BEQ      2$
3285   016056  016767  163034  163030           MOV      NEXT,RTRN         ;SET UP FOR NEXT TEST
3286   016064  012706  001100          1$:     MOV      #STACK,SP         ;REINITIALIZE SP
3287   016070  000177  163020                  JMP      @RTRN
```

```
3288  016074  000002              2$:     RTI
3289  016076  000001              ERTABC: 1
3290  016100    006      002              .BYTE   6,2
3291  016102  001174                      SAVPC
3292                                ;ENTER HERE ON POWER FAILURE
3293
3294
3295  016104  010046              .PFAIL: MOV     R0,-(SP)            ;SAVE R0-R5 ON PROCESSOR STACK
3296  016106  010146                      MOV     R1,-(SP)
3297  016110  010246                      MOV     R2,-(SP)
3298  016112  010346                      MOV     R3,-(SP)
3299  016114  010446                      MOV     R4,-(SP)
3300  016116  010546                      MOV     R5,-(SP)
3301  016120  016746   161700             MOV     24,-(SP)
3302  016124  010667   163042             MOV     SP,SAVSP            ;SAVE STACK POINTER
3303  016130  012767   016142 161666      MOV     #RESTART,24         ;SET UP FOR POWER UP TRAP
3304  016136  000000                      HALT                        ;HALT ON POWER DOWN NORMAL
3305  016140  000777              1$:     BR      1$
3306
3307                                ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3308
3309  016142  016706   163024     RESTAR: MOV     SAVSP,SP            ;RESTORE STACK POINTER
3310  016146  012605                      MOV     (SP)+,R5            ;RESTORE R0-R5
3311  016150  012604                      MOV     (SP)+,R4
3312  016152  012603                      MOV     (SP)+,R3
3313  016154  012602                      MOV     (SP)+,R2
3314  016156  012601                      MOV     (SP)+,R1
3315  016160  012600                      MOV     (SP)+,R0
3316  016162  012767   016104 161634      MOV     #.PFAIL,24          ;SET UP FOR POWER FAILURE
3317  016170  012767   000340 161600      MOV     #340,PS
3318  016176  012706   001100             MOV     #STACK,SP
3319  016202  005067   001532             CLR     TEMP
3320  016206  005267   001526     1$:     INC     TEMP
3321  016212  001375                      BNE     1$
3322  016214  104410                      CONVRT
3323  016216  016240                      PFTAB
3324  016220  104402                      TYPE
3325  016222  017113                      MPFAIL
3326  016224  005067   163016             CLR     ERRFLG
3327  016230  005067   162700             CLR     LSTERR
3328  016234  000177   162654             JMP     @RTRN
3329  016240  000001              PFTAB:  1
3330  016242    006      002              .BYTE   6,2
3331  016244  001114                      RTRN
3332
3333
3334                                ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
3335                                ;OF LOC.176.
3336                                ;LOCATIONS USED:
3337  016246  000000              RDSW:   .WORD   0
3338
3339
3340  016250  005737   000042     .CKSWR: TST     @#42
3341  016254  001042                      BNE     OUT
3342  016256  022767   000176 162614      CMP     #SWREG,SWR          ;SOFTWARE SWITCH REGISTER PRESENT
3343  016264  001036                      BNE     OUT                 ;NO, GET OUT
```

```
3344  016266  105777  162612              TSTB    @TKCSR          ;YES,WAIT FOR
3345  016272  100033                      BPL     OUT             ;READY, GET CHARACTER
3346  016274  017767  162606  176352      MOV     @TKDBR,.MSG     ;AND STRIP OFF
3347  016302  042767  177600  176344      BIC     #177600,.MSG    ;THE GARBAGE
3348  016310  122767  000007  176336      CMPB    #7,.MSG         ;IS IT A <^G>
3349  016316  001021                      BNE     OUT
3350  016320  104402  016376              TYPE,$CNTG
3351  016324  005137  016246      .CNTLU: COM     @#RDSW
3352  016330  104402  016403              TYPE,$MSWR
3353  016334  104411  016370              CNVRT,SWREGC
3354  016340  104403  016413              INSTR,$MNEW
3355  016344  104405                      PARAM
3356  016346  000000                      0
3357  016350  177777                      177777
3358  016352  000176                      SWREG
3359  016354     000     001      .BYTE           0,1
3360  016356  104402  017110              TYPE,MCRLF
3361  016362  005037  016246      OUT:    CLR     @#RDSW
3362  016366  000002                      RTI
3363  016370  000001      SWREGC: 1
3364  016372     006     002      .BYTE           6,2
3365  016374  000176                      SWREG
3366  016376  005015  043536     000 $CNTG:  .ASCIZ  <15><12>/^G/
3367  016403     015  051412  051127 $MSWR:  .ASCIZ  <15><12>/SWR= /
3368  016410  020075     000
3369  016413     040  047040  053505 $MNEW:  .ASCIZ  /  NEW= /
3370  016420  020075     000
3371          016424              .EVEN
3372  016424  005015  042012  030525 MTITLE: .ASCIZ  <15><12><12>/DU11 CZDUB-C TAPE B /<15><12>
3373  016432  020061  055103  052504
3374  016440  026502  020103  040524
3375  016446  042520  041040  006440
3376  016454  000012
3377  016456  005015  042526  052103 MVECTO: .ASCIZ  <15><12>/VECTOR ADDRESS-/
3378  016464  051117  040440  042104
3379  016472  042522  051523  000055
3380  016500  005015  051461  020124 MREGAD: .ASCIZ  <15><12>/1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-/
3381  016506  042504  044526  042503
3382  016514  020072  042522  042503
3383  016522  053111  051105  041440
3384  016530  047117  051124  046117
3385  016536  051104  043505  051511
3386  016544  042524  020122  042101
3387  016552  051104  051505  026523
3388  016560     000
3389  016561     015  040412  042522 MMULT:  .ASCIZ  <15><12>/ARE YOU RUNNING MULTIPLE DEVICES ? (Y OR N)-/
3390  016566  054440  052517  051040
3391  016574  047125  044516  043516
3392  016602  046440  046516  044524
3393  016610  046120  020105  042504
3394  016616  044526  042503  020123
3395  016624  020077  054450  047440
3396  016632  020122  024516  000055
3397  016640  005015  040514  052123 MLASTD: .ASCIZ  <15><12>/LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-/
3398  016646  042040  053105  041511
3399  016654  035105  042522  042503
```

```
3400   016662   053111   051105   041440
3401   016670   047117   051124   046117
3402   016676   051040   043505   051511
3403   016704   042524   020122   042101
3404   016712   051104   051505   026523
3405   016720      000
3406   016721      075   042504   044526   DEVICE: .ASCIZ  /=DEVICE /
3407   016726   042503   020040      000
3408   016733      015   044012   053517   MCOW:   .ASCIZ  <15><12>/HOW NOW BROWN COW? ...SELECT SOMETHING TO RUN aACTREG/
3409   016740   047040   053517   041040
3410   016746   047522   047127   041440
3411   016754   053517   020077   027056
3412   016762   051456   046105   041505
3413   016770   020124   047523   042515
3414   016776   044124   047111   020107
3415   017004   047524   051040   047125
3416   017012   040040   041501   051124
3417   017020   043505      000
3418   017023      015   047412   052125   MRANGE: .ASCIZ  <15><12>/OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-/
3419   017030   047440   020106   040522
3420   017036   043516   035105   042522
3421   017044   054524   042520   046040
3422   017052   051501   020124   042504
3423   017060   044526   042503   051040
3424   017066   041530   051123   040440
3425   017074   042104   042522   051523
3426   017102   000055
3427   017104   020040   000077            MQM:    .ASCIZ  /  ?/
3428   017110   005015      000            MCRLF:  .ASCIZ  <15><12>
3429   017113      040   050040   053517   MPFAIL: .ASCIZ  /  POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
3430   017120   051105   043040   044501
3431   017126   052514   042522   020054
3432   017134   051120   043517   040522
3433   017142   020115   042522   052123
3434   017150   051101   020124   052101
3435   017156   052040   051505   020124
3436   017164   047111   050040   047522
3437   017172   051107   051505   000123
3438   017200   005015   047105   020104   MEPASS: .ASCIZ  <15><12>/END OF PASS TAPE B/
3439   017206   043117   050040   051501
3440   017214   020123   040524   042520
3441   017222   041040      000
3442   017225      015   051012      000   MR:     .ASCIZ  <15><12>/R/
3443   017231      015   052012   051505   MTSTPC: .ASCIZ  <15><12>/TEST PC-/
3444   017236   020124   041520   000055
3445   017244   005015   047514   045503   MLOCK:  .ASCIZ  <15><12>/LOCK ON SELECTED TEST? (Y OR N)-/
3446   017252   047440   020116   042523
3447   017260   042514   052103   042105
3448   017266   052040   051505   037524
3449   017274   024040   020131   051117
3450   017302   047040   026451      000
3451   017307      015   042012   020125   MLEVEL: .ASCIZ  <15><12>/DU PRIORITY LEVEL-/
3452   017314   051120   047511   044522
3453   017322   054524   046040   053105
3454   017330   046105   000055
3455   017334   005015   020043   043117   MSYNC:  .ASCIZ  <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/
```

```
3456  017342  051440  047131  020103
3457  017350  044103  051101  020123
3458  017356  042523  042514  052103
3459  017364  042105  024040  030440
3460  017372  047440  020122  024462
3461  017400  000055
3462  017402  005015  051511  051440  MWIRE6: .ASCIZ  <15><12>/IS SEC XMIT JUMPER #6 IN? (Y OR N)-/
3463  017410  041505  054040  044515
3464  017416  020124  052512  050115
3465  017424  051105  021440  020066
3466  017432  047111  020077  054450
3467  017440  047440  020122  024516
3468  017446  000055
3469  017450  005015  051511  051440  MWIRE5: .ASCIZ  <15><12>/IS SEC REC JUMPER #5 IN? (Y OR N)-/
3470  017456  041505  051040  041505
3471  017464  045040  046525  042520
3472  017472  020122  032443  044440
3473  017500  037516  024040  020131
3474  017506  051117  047040  026451
3475  017514     000
3476  017515     015  044412  020123  MWIRE4: .ASCIZ  <15><12>/IS OPT CLR ENABLE JUMPER #4 IN? (Y OR N)-/
3477  017522  050117  020124  046103
3478  017530  020122  047105  041101
3479  017536  042514  045040  046525
3480  017544  042520  020122  032043
3481  017552  044440  037516  024040
3482  017560  020131  051117  047040
3483  017566  026451     000
3484  017571     015  044412  020123  MEXTJ:  .ASCIZ  <15><12>/IS THE TEST CONNECTOR INSTALLED ?(Y OR N)-/
3485  017576  044124  020105  042524
3486  017604  052123  041440  047117
3487  017612  042516  052103  051117
3488  017620  044440  051516  040524
3489  017626  046114  042105  037440
3490  017634  054450  047440  020122
3491  017642  024516  000055
3492  017646  006412  020040  020040  MSTATUS: .ASCIZ  <12> <15>/    STATUS    MAP    / <12> <15>
3493  017654  052123  052101  051525
3494  017662  020040  046440  050101
3495  017670  020040  020040  005040
3496  017676  000015
3497                                  .EVEN
3498
3499                                     ;BUFFERS FOR INPUT-OUTPUT
3500
3501  017700  000040                  INBUF:  .BLKB   40
3502  017740  000040                  TEMP:   .BLKB   40
3503  020000  000040                  MDATA:  .BLKB   40
3504                                  ;***********************************************************
3505                                     ;UTILITIES
3506                                  ;***********************************************************
3507
3508                                  ;THIS UTILITY CALCULATES PRIORITY LEVEL
3509  020040  006367  000044          DULEV:  ASL     DUPRT   ;SHIFT LEFT
3510  020044  006367  000040                  ASL     DUPRT   ;
3511  020050  006367  000034                  ASL     DUPRT   ;
```

```
3512  020054  006367  000030             ASL     DUPRT   :
3513  020060  006367  000024             ASL     DUPRT   :
3514  020064  016767  000020  000020     MOV     DUPRT,LESS1      ;MOVE THIS TO LESS1
3515  020072  162767  000001  000012     SUB     #1,LESS1         ;CREATE LESS1
3516  020100  042767  000037  000004     BIC     #37,LESS1        ;CLEAR TNZVC
3517  020106  000207                     RTS     PC
3518  020110  000240           DUPRT:    LEVEL5
3519  020112  000200           LESS1:    LEVEL4  ;LEVEL TO ALLOW INTERRUPTS
3520
3521                                      ;NEW DU ADDRESSES
3522  020114  016767  000126  000422  DUADDR: MOV   DUBASE,RXCSR   ;XXX0
3523  020122  005267  000120             INC     DUBASE
3524  020126  016767  000114  000412     MOV     DUBASE,HRXCSR    ;XXX1
3525  020134  005267  000106             INC     DUBASE
3526  020140  016767  000102  000402     MOV     DUBASE,RXDBUF    ;XXX2
3527  020146  016767  000074  000400     MOV     DUBASE,PARCSR    ;XXX2
3528  020154  005267  000066             INC     DUBASE
3529  020160  016767  000062  000364     MOV     DUBASE,HRXDBUF   ;XXX3
3530  020166  016767  000054  000362     MOV     DUBASE,HPARCSR   ;XXX3
3531  020174  005267  000046             INC     DUBASE
3532  020200  016767  000042  000352     MOV     DUBASE,TXCSR     ;XXX4
3533  020206  005267  000034             INC     DUBASE
3534  020212  016767  000030  000342     MOV     DUBASE,HTXCSR    ;XXX5
3535  020220  005267  000022             INC     DUBASE
3536  020224  016767  000016  000332     MOV     DUBASE,TXDBUF    ;XXX6
3537  020232  005267  000010             INC     DUBASE
3538  020236  016767  000004  000322     MOV     DUBASE,HTXDBUF   ;XXX7
3539  020244  000207                     RTS     PC
3540  020246  000000           DUBASE:   0
3541
3542                                      ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3543                                      ;INFORMATION CONTAINED IN TEMP1 AND IT IS
3544                                      ;SHIFTED IN BY THE CONTENTS OF SHIFT
3545  020250  042777  040000  000302  RPOKE:  BIC   #MTDATA,@TXCSR
3546  020256  005067  160664             CLR     TEMP2
3547  020262  006067  160656             ROR     TEMP1    ;FORCE CARRY
3548  020266  006067  160654             ROR     TEMP2    ;PICK UP CARRY IN BIT 15
3549  020272  006267  160650             ASR     TEMP2    ;SHIFT INTO BIT 14
3550  020276  042767  100000  160642     BIC     #BIT15,TEMP2     ;CLR BIT 15
3551  020304  056777  160636  000246     BIS     TEMP2,@TXCSR     ;POKE MAINT DATA
3552  020312  042777  020000  000240     BIC     #CLK,@TXCSR      ;POKE CLK
3553  020320  052777  020000  000232     BIS     #CLK,@TXCSR      :
3554  020326  005367  160606             DEC     SHIFT
3555  020332  001346                     BNE     RPOKE
3556  020334  000207                     RTS     PC
3557                                      ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3558  020336  016767  160602  160602  ODD8:   MOV   TEMP1,TEMP2    ;SAVE TEMP1
3559  020344  005067  160600             CLR     TEMP3
3560  020350  012727  000010             MOV     #8.,(PC)+
3561  020354  000000           4$:       0
3562  020356  006067  160564  1$:        ROR     TEMP2
3563  020362  005567  160562             ADC     TEMP3
3564  020366  005367  177762             DEC     4$
3565  020372  001371                     BNE     1$
3566  020374  006067  160550             ROR     TEMP3
3567  020400  103404                     BCS     2$
```

```
3568  020402  052767  000400  160534          BIS     #BIT8,TEMP1      ;SET ODD PARITY
3569  020410  000403                           BR      3$
3570  020412  042767  000400  160524  2$:      BIC     #BIT8,TEMP1      ;CLR EVEN PARITY
3571                                            ;TEMP1 NOW HAS ODD PARITY CHARACTER
3572  020420  000207                  3$:      RTS     PC
3573
3574                                            ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3575  020422  016767  160516  160516  EVEN8:   MOV     TEMP1,TEMP2      ;SAVE TEMP1
3576  020430  005067  160514                   CLR     TEMP3
3577  020434  012727  000010                   MOV     #8.,(PC)+
3578  020440  000000                  4$:      0
3579  020442  006067  160500          1$:      ROR     TEMP2
3580  020446  005567  160476                   ADC     TEMP3
3581  020452  005367  177762                   DEC     4$
3582  020456  001371                           BNE     1$
3583  020460  006067  160464                   ROR     TEMP3
3584  020464  103004                           BCC     2$
3585  020466  052767  000400  160450           BIS     #BIT8,TEMP1      ;SET EVEN PARITY
3586  020474  000403                           BR      3$
3587  020476  042767  000400  160440  2$:      BIC     #BIT8,TEMP1      ;CLR ODD PARITY
3588                                            ;TEMP1 NOW HAS EVEN PARITY CHARACTER
3589  020504  000207                  3$:      RTS     PC
3590
3591  020506  062716  000002          TRPREG:  ADD     #2,(SP)  ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3592                                            ;IN MAIN PART OF THE PROGRAM
3593  020512  000002                           RTI
3594                                            ;ERROR HLT TABLE
3595  020514  020600                  .ERRTAB: EM0      ;HLT 0  BIT ERROR (GENERAL)
3596  020516  000000                           0
3597  020520  000000                           0
3598  020522  020614                           EM1      ;HLT 1  REGISTER ERROR
3599  020524  020765                           DH1
3600  020526  021006                           DT1
3601  020530  020656                           EM2      ;HLT 2  RECEIVER ERROR
3602  020532  020765                           DH1
3603  020534  021006                           DT1
3604  020536  020720                           EM3      ;HLT 3  TRANSMITTER ERROR
3605  020540  020765                           DH1
3606  020542  021006                           DT1
3607                                            ;DEFAULT DU ADDRESSES
3608  020544  160040                  RXCSR:   160040
3609  020546  160041                  HRXCSR:  160041
3610  020550  160042                  RXDBUF:  160042
3611  020552  160043                  HRXDBUF:         160043
3612  020554  160042                  PARCSR:  160042
3613  020556  160043                  HPARCSR:         160043
3614  020560  160044                  TXCSR:   160044
3615  020562  160045                  HTXCSR:  160045
3616  020564  160046                  TXDBUF:  160046
3617  020566  160047                  HTXDBUF:         160047
3618                                            ;DEFAULT DU VECTORS
3619  020570  000770                  DURIV:   770      ;REC INTR VECTOR
3620  020572  000772                  DURIS:   772      ;REC INTR STATUS
3621  020574  000774                  DUTIV:   774      ;XMIT INTR VECTOR
3622  020576  000776                  DUTIS:   776      ;XMIT INTR STATUS
3623                                            ;ERROR MESSAGES
```

```
3624  020600  036440  042440  051122  EM0:   .ASCIZ  / = ERROR PC/
3625  020606  051117  050040  000103
3626  020614  036440  051040  043505  EM1:   .ASCIZ  / = REGISTER ERROR PC/<15><12><1>/REGISTER /
3627  020622  051511  042524  020122
3628  020630  051105  047522  020122
3629  020636  041520  005015  051001
3630  020644  043505  051511  042524
3631  020652  020122  000040
3632  020656  036440  051040  041505  EM2:   .ASCIZ  / = RECEIVER ERROR PC/<15><12><1>/REGISTER /
3633  020664  044505  042526  020122
3634  020672  051105  047522  020122
3635  020700  041520  005015  051001
3636  020706  043505  051511  042524
3637  020714  020122  000040
3638  020720  036440  052040  040522  EM3:   .ASCIZ  / = TRANSMITTER ERROR PC/<15><12><1>/REGISTER /
3639  020726  051516  044515  052124
3640  020734  051105  042440  051122
3641  020742  051117  050040  006503
3642  020750  000412  042522  044507
3643  020756  052123  051105  020040
3644  020764     000
3645                                          ;DATA HEADERS FOR ERROR MESSAGES
3646  020765     105  050130  041505  DH1:   .ASCIZ  /EXPECTED  ACTUAL/
3647  020772  042524  020104  040440
3648  021000  052103  040525  000114
3649                                   .EVEN
3650                                          ;DATA TABLES FOR ERROR MESSAGES
3651  021006  000003                 DT1:   3
3652  021010     006     004          .BYTE   6,4
3653  021012  001164                  SAVR3   ;REGISTER
3654  021014     006     004          .BYTE   6,4
3655  021016  001156                  SAVR0   ;EXPECTED DATA
3656  021020     006     002          .BYTE   6,2
3657  021022  001160                  SAVR1   ;ACTUAL DATA
3658          000001                 .END
```

F 6

DZDUB-C MACY11 30A(1052) 21-NOV-78 15:36 PAGE 74
CZDUBD.P11   31-OCT-78 13:59   CROSS REFERENCE TABLE -- USER SYMBOLS   SEQ 0070

```
AAA      002520  .      1177#
ACTREG   001216         897#   1061*  1125*  1139*  1140*  1147*  1229   2909   2912   2922
ADDBAS   001306         937#   1057   1224*
ADDKEE   001310         938#   1058   1225*
ADDLAS   001312         939#   1059   1226*
ADDROT   001322         943#   1062   1230*
ADRCNT=  015215         3074*  3113*  3122#
A10010=  ****** GX      676
BASEAD   001204         892#   1057*  1064*  1107*  1144*  1145   1151*  1153*  1224   2916*  2928*  2932
BASEDU   001364         960#   1071   1237*
BASEIV   001212         895#   1063*  1117*  1228   2917*  2929*  2934   2935*  2936   2937*  2938   2939*  2940
                        2941*
BBB      002336         1124   1128#
BINWRD   015532         3163*  3164   3201#
BITW  =  002000         807#
BIT0  =  000001         755#   783    814
BIT1  =  000002         754#   782
BIT10 =  002000         745#   773    807
BIT11 =  004000         744#   772
BIT12 =  010000         743#   771    788
BIT13 =  020000         742#   770    787    806
BIT14 =  040000         741#   769    786    805
BIT15 =  100000         740#   768    785    804    3550
BIT2  =  000004         753#   781    1292
BIT3  =  000010         752#   780    813
BIT4  =  000020         751#   779    812
BIT5  =  000040         750#   778    811
BIT6  =  000100         749#   777    810
BIT7  =  000200         748#   776    809
BIT8  =  000400         747#   775    791    808    3568   3570   3585   3587
BIT9  =  001000         746#   774    790
BREAK =  000001         814#   1335   1393   1451   1509   1567   1625   1683   1741   1799   1857   1915   1973
                        2031   2086   2141   2196   2251   2306   2361   2416   2471   2526   2581   2636   2691
                        2746   2801   2856
BRW      014546         1289   1303   3002#
BRX      014550         1290   1304   3003#
BUFF1    000002R        676#   1266   1272
BUFHRX   001346         953#   1075   1241*
BUFHTX   001362         959#   1081   1247*
BUFRXD   001344         952#   1074   1240*
BUFTXD   001360         958#   1080   1246*
BYJMR    001305         933#   1056   1223*
CARDET=  010000         771#
CCC      014302         2908   2931   2948#
CHRCNT   015530         3161*  3165   3179*  3199#  3200
CKSWR =  104413         992#   2961   2983   3007   3236          3279
CLK   =  020000         806#   1335   1341   1342   1344   1345   1393   1399   1400   1402   1403   1451   1457
                        1458   1460   1461   1509   1515   1516   1518   1519   1567   1573   1574   1576   1577
                        1625   1631   1632   1634   1635   1683   1689   1690   1692   1693   1741   1747   1748
                        1750   1751   1799   1805   1806   1808   1809   1857   1863   1864   1866   1867   1915
                        1921   1922   1924   1925   1973   1979   1980   1982   1983   2031   2037   2038   2086
                        2092   2093   2141   2147   2148   2196   2202   2203   2251   2257   2258   2306   2312
                        2313   2361   2367   2368   2416   2422   2423   2471   2477   2478   2526   2532   2533
                        2581   2587   2588   2636   2642   2643   2691   2697   2698   2746   2752   2753   2801
                        2807   2808   2856   2862   2863   3552   3553
CLROPT   001303         931#   1054   1221*
```

G 6

DZDUB-C MACY11 30A(1052) 21-NOV-78 15:36 PAGE 75
CZDUBD.P11   31-OCT-78 13:59   CROSS REFERENCE TABLE -- USER SYMBOLS   SEQ 0071

```
CNTLU = 104414         994#    1039
CNVRT = 104411         988#    3353
CONVRT= 104410         986#    1275    2905    3259    3270    3322
COUNT   001142         868#
COUNT1  001230         908#    1265*   1277*
CSRHPA  001352         955#    1077    1243*
CSRHRX  001342         951#    1073    1239*
CSRHTX  001356         957#    1079    1245*
CSRPAR  001350         954#    1076    1242*
CSRRX   001340         950#    1072    1238*
CSRTX   001354         956#    1078    1244*
CTS   = 020000         770#
D     = ****** GX      675
DATABP  016004        3254*    3257    3268    3271#
DATAHD  015772        3253*    3264    3267#
DEVADR  015212        3072*    3110    3120#
DEVICE  016721        2906    3406#
DH1     020765        3599    3602    3605    3646#
DISPRE  000174         835#    1032
DLIGHT= 177570         725#    1023
DMULT   001304         932#    1055    1222*
DNA   = 100000         804#
DNAINT= 000040         811#
DSC   = 100000         768#
DSINTE= 000040         778#
DSR   = 001000         774#
DSWR  = 177570         724#    1022
DTR   = 000002         782#
DT1     021006        3600    3603    3606    3651#
DUADDR  020114        1106    2933    3522#
DUBASE  020246        1071*   1102    1105    1237    2932*   3522    3523*   3524    3525*   3526    3527    3528*   3529
                      3530    3531*   3532    3533*   3534    3535*   3536    3537*   3538    3540#
DULEV   020040        1172    3509#
DUPRT   020110        1070*   1169    1231    3509*   3510*   3511*   3512*   3513*   3514    3518#
DURIS   020572        1066*   1233    2936*   3620#
DURIV   020570        1065*   1113    1116    1117    1232    2934*   2941    3619#
DUTIS   020576        1068*   1235    2940*   3622#
DUTIV   020574        1067*   1234    2938*   3621#
EIGHT = 006000         799#    1802    1860    1918    1976    2694    2749    2804    2859
EM0     020600        3595    3624#
EM1     020614        3598    3626#
EM2     020656        3601    3632#
EM3     020720        3604    3638#
ERRCNT  001132         861#    1013*   3280*
ERRFLG  001246         920#    1012*   2950*   2996*   3242*   3255    3261*   3326*
ERRMSG  015760        3252*   3263#
ERTAB0  016076        3260    3289#
EVEN8   020422        3575#
EVEPAR= 001400         802#
EVPAR = 000400         791#
EXITER  016030        3274    3279#
FILBUF  003204        1268#   1271
FIVE  = 000000         796#    2034    2089    2144    2199
FLAG    001222         905#    1087*   1258    1262*
FRMERR= 020000         787#
HALTS   016010        3238    3273#
```

H 6

DZDUB-C MACY11 30A(1052) 21-NOV-78 15:36 PAGE 76
CZDUBD.P11   31-OCT-78 13:59   CROSS REFERENCE TABLE -- USER SYMBOLS   SEQ 0072

```
HDXEN = 000010          813#
HEREU   003132         1217    1258#
HILIM   015210         3071*   3101    3119#
HOLD    001136          866#
HOLD0   001224          906#   1263*   1279
HOLD1   001226          907#   1264*   1280
HPARCS  020556         1077*   1243    3530*   3613#
HRXCSR  020546         1073*   1239    3524*   3609#
HRXDBU  020552         1075*   1241    3529*   3611#
HTXCSR  020562         1079*   1245    3534*   3615#
HTXDBU  020566         1081*   1247    3538*   3617#
ICOUNT  001122          857#   2994    2998#
INBUF   017700         1180    1184    3035    3077    3209    3213    3501#
INIFLG  001244          918#   1018    1021*
INSTER= 104404          978#   1188    3096    3218
INSTR = 104403          976#   1097    1108    1118    1129    1154    1164    1178    1191    1195    1199    1203    1294
                       1307    3354
INSTR2  014774         3046    3061#
ISYMOD= 000000          795#   1331    1338    1389    1396    1447    1454    1505    1512    1563    1570    1621    1628
                       1679    1686    1737    1744    1795    1802    1853    1860    1911    1918    1969    1976
IVBASE  001314          940#   1063    1228*
IVKEEP  001316          941#   1060    1227*
JMRBY   001203          888#   1056*   1206    1223
KEEPAD  001206          893#   1058*   1064    1105*   1107    1151    1153    1225    2928
KEEPIV  001214          896#   1060*   1116*   1227    2929
LASTAD  001210          894#   1059*   1134    1145    1159    1226
LESS1   020112         1069*   1236    3514*   3515*   3516*   3519#
LEVEL0= 000000          765#
LEVEL1= 000040          764#
LEVEL2= 000100          763#
LEVEL3= 000140          762#
LEVEL4= 000200          761#   3519
LEVEL5= 000240          760#   3518
LEVEL6= 000300          759#
LEVEL7= 000340          758#
LIGHTS  001102          846#   1023*   1032*   2952*
LIMITS  015136         3087    3101#
LOBITS  015214         3073*   3105    3121#   3122
LOCK    001120          856#   3010
LOGICA  014334         2956#
LOKFLG  001247          921#   1297    1298
LOLIM   015206         3070*   3103    3118#
LPCNT   001124          858#   1009*   2993*   2994    2997*
LSTERR  001134          862#   1014*   2949*   3239    3241*   3327*
L1ESS   001336          949#   1069    1236*
MAP     001634         1036    1038    1046#
MCOW    016733         2911    3408#
MCRLF   017110         1276    3043    3059    3151    3360    3428#
MDATA   020000         3177    3188    3503#
MEPASS  017200         2904    3438#
MEXT  = 010000          818#
MEXTJ   017571         1204    3484#
MINT  = 004000          817#   1335    1393    1451    1509    1567    1625    1683    1741    1799    1857    1915    1973
                       2031    2086    2141    2196    2251    2306    2361    2416    2471    2526    2581    2636    2691
                       2746    2801    2856
MITSEX  001301          929#   1052    1219*
```

I 6

DZDUB-C MACY11 30A(1052) 21-NOV-78 15:36 PAGE 77
CZDUBD.P11   31-OCT-78 13:59   CROSS REFERENCE TABLE -- USER SYMBOLS   SEQ 0073

```
MLASTD  016640          1130    3397#
MLEVEL  017307          1165    3451#
MLOCK   017244          1295    3445#
MMULT   016561          1119    3389#
MPFAIL  017113          3325    3429#
MQM     017104          3056    3427#
MR      017225          1317    3442#
MRANGE  017023          1155    3418#
MREGAD  016500          1098    3380#
MRESET= 000400          808#    1330    1332    1388    1390    1446    1448    1504    1506    1562    1564    1620    1622
                        1678    1680    1736    1738    1794    1796    1852    1854    1910    1912    1968    1970    2026
                        2028    2081    2083    2136    2138    2191    2193    2246    2248    2301    2303    2356    2358
                        2411    2413    2466    2468    2521    2523    2576    2578    2631    2633    2686    2688    2741
                        2743    2796    2798    2851    2853
MSTATU  017646          1261    3492#
MSYNC   017334          1179    3455#
MTDATA= 040000          805#    1335    1393    1451    1509    1567    1625    1683    1741    1799    1857    1915    1973
                        2031    2086    2141    2196    2251    2306    2361    2416    2471    2526    2581    2636    2691
                        2746    2801    2856    3545
MTITLE  016424          1020    3372#
MTSTPC  017231          1308    3443#
MULTD   001202          887#    1055*   1121    1122    1222    1286    2907
MVECTO  016456          1109    3377#
MWIRE4  017515          1200    3476#
MWIRE5  017450          1196    3469#
MWIRE6  017402          1192    3462#
NEXT    001116          855#    1329*   1387*   1445*   1503*   1561*   1619*   1677*   1735*   1793*   1851*   1909*   1967*
                        2025*   2080*   2135*   2190*   2245*   2300*   2355*   2410*   2465*   2520*   2575*   2630*   2685*
                        2740*   2795*   2850*   2979    2999    3285
NOPAR = 000000          800#    1338    1396    1454    1512    1570    1628    1686    1744    1802    1860    1918    1976
                        2034    2089    2144    2199    2254    2309    2364    2419    2474    2529    2584    2639    2694
                        2749    2804    2859
NOSYNC  001300          928#    1051    1218*
ODDPAR= 001000          801#
ODD8    020336          3558#
ONCE    001522          1019    1022#
OPTCLR  001201          886#    1054*   1202    1221
OUT     016362          3341    3343    3345    3349    3361#
OUTCRY  014274          2905    2944#
OUTMUL  002476          1127    1152    1163#
OVRRUN= 040000          786#
PARAM = 104405          980#    1099    1110    1131    1156    1166    1309    3355
PARAM1  015042          3076#   3097
PARCSR  020554          1076*   1242    1331*   1338*   1389*   1396*   1447*   1454*   1505*   1512*   1563*   1570*   1621*
                        1628*   1679*   1686*   1737*   1744*   1795*   1802*   1853*   1860*   1911*   1918*   1969*   1976*
                        2027*   2034*   2082*   2089*   2137*   2144*   2192*   2199*   2247*   2254*   2302*   2309*   2357*
                        2364*   2412*   2419*   2467*   2474*   2522*   2529*   2577*   2584*   2632*   2639*   2687*   2694*
                        2742*   2749*   2797*   2804*   2852*   2859*   3527*   3612#
PAREN = 001000          790#
PARER = 010000          788#
PARERR  015116          3079    3081    3083    3092#   3102    3104    3106
PARTI   015200          3095    3115#
PASCNT  001130          860#    1011*   2951*   2952
PFTAB   016240          3323    3329#
POPRO = 012600          734#    3278
POP1SP= 005726          732#
```

J 6

DZDUB-C MACY11 30A(1052) 21-NOV-78 15:36 PAGE 78
CZDUBD.P11   31-OCT-78 13:59   CROSS REFERENCE TABLE -- USER SYMBOLS   SEQ 0074

```
POP2SP= 022626       736#
PRTDU  001324        944#   1070   1231*
PS   = 177776        726#   1006*  1211*   2960*   3317*
PUSHR0= 010046       733#   3275
PUSH1S= 005746       731#
PUSH2S= 024646       735#
RDSW   016246        3057   3094   3337#   3351*   3361*
RECACT= 004000       772#
REGACT 001320        942#   1061   1229*
REPLAY 014200        2924   2930   2932#
RESEC  001302        930#   1053   1220*
RESREG 016006        3269   3272#
RESTAR 016142        3303   3309#
RESTRT 014344        2925   2954   2960#
RES05 = 104407       984#   3272
RING  = 040000       769#
RINTEN= 000100       777#
RISDU  001334        948#   1066   1233*
RIVDU  001326        945#   1065   1232*
ROTADD 001220        900#   1062*  1126*   1138*   1140   1142*   1147   1150*   1230   2919*   2922   2926*
RPOKE  020250        1350   1364   1368    1408    1422   1426    1466   1480    1484   1524    1538   1542   1582
                     1596   1600   1640    1654    1658   1698    1712   1716    1756   1770    1774   1814   1828
                     1832   1872   1886    1890    1930   1944    1948   1988    2002   2006    2043   2057   2061
                     2098   2112   2116    2153    2167   2171    2208   2222    2226   2263    2277   2281   2318
                     2332   2336   2373    2387    2391   2428    2442   2446    2483   2497    2501   2538   2552
                     2556   2593   2607    2611    2648   2662    2666   2703    2717   2721    2758   2772   2776
                     2813   2827   2831    2868    2882   2886    3545#  3555
RTRN   001114        854#   1016*  1312    1316*   1318   2962*   2979*  2980    2999*  3000    3285*  3287   3328
                     3331
RTS   = 000004       781#
RUNA  = ****** U        1     53     69     147     663
RUNB  = 000000          1     53     69     147     663
RUNC  = ****** U       21     57     69     150     668
RUND  = ****** U       21     57     69     150     668
RUNE  = ****** U       21     57     72     150     668
RUNF  = ****** U       21     57     72     150     668
RUNIT  014104        2910   2916#  2923
RXCSR  020544        1072*  1238   1339*   1351    1397*  1409    1455*  1467    1513*  1525    1571*  1583   1629*
                     1641   1687*  1699    1745*   1757   1803*   1815   1861*   1873   1919*   1931   1977*  1989
                     2035*  2044   2090*   2099    2145*  2154    2200*  2209    2255*  2264    2310*  2319   2365*
                     2374   2420*  2429    2475*   2484   2530*   2539   2585*   2594   2640*   2649   2695*  2704
                     2750*  2759   2805*   2814    2860*  2869    2946   3522*   3608#
RXDBUF 020550        1074*  1240   1346    1355    1371   1404    1413   1429    1462   1471    1487   1520   1529
                     1545   1578   1587    1603    1636   1645    1661   1694    1703   1719    1752   1761   1777
                     1810   1819   1835    1868    1877   1893    1926   1935    1951   1984    1993   2009   2039
                     2048   2064   2094    2103    2119   2149    2158   2174    2204   2213    2229   2259   2268
                     2284   2314   2323    2339    2369   2378    2394   2424    2433   2449    2479   2488   2504
                     2534   2543   2559    2589    2598   2614    2644   2653    2669   2699    2708   2724   2754
                     2763   2779   2809    2818    2834   2864    2873   2889    3526*  3610#
RXDONE= 000200       776#
RXERR = 100000       785#
SAVPC  001174        881#   3126*  3291
SAVR0  001156        874#   3135*  3140    3655
SAVR1  001160        875#   3134*  3141    3657
SAVR2  001162        876#   3133*  3142
SAVR3  001164        877#   3132*  3143    3653
```

K 6

DZDUB-C MACY11 30A(1052) 21-NOV-78 15:36 PAGE 79
CZDUBD.P11   31-OCT-78 13:59   CROSS REFERENCE TABLE -- USER SYMBOLS   SEQ 0075

```
SAVR4   001166      878#   3131*   3144
SAVR5   001170      879#   3130*   3145
SAVSP   001172      880#   3302*   3309
SAV05 = 104406      982#   3243
SCOPE = 104400      970#   1377    1435    1493    1551    1609    1667    1725    1783    1841    1899    1957    2015
                   2070    2125    2180    2235    2290    2345    2400    2455    2510    2565    2620    2675    2730
                   2785    2840    2895
SCOP1 = 104401      972#
SEND  = 000020      812#
SEREC   001200      885#   1053*   1198    1220
SETFG   003144     1259    1261#
SETFLG= 104412      990#   1120    1193    1197    1201    1205    1296
SEVEN = 004000      798#   1570    1628    1686    1744    2474    2529    2584    2639
SEXMIT  001177      884#   1052*   1194    1219
SHIFT   001140      867#   1348#   1362*   1366*   1406*   1420*   1424*   1464*   1478*   1482*   1522*   1536*   1540*
                   1580*   1594*   1598*   1638*   1652*   1656*   1696*   1710*   1714*   1754*   1768*   1772*   1812*
                   1826*   1830*   1870*   1884*   1888*   1928*   1942*   1946*   1986*   2000*   2004*   2041*   2055*
                   2059*   2096*   2110*   2114*   2151*   2165*   2169*   2206*   2220*   2224*   2261*   2275*   2279*
                   2316*   2330*   2334*   2371*   2385*   2389*   2426*   2440*   2444*   2481*   2495*   2499*   2536*
                   2550*   2554*   2591*   2605*   2609*   2646*   2660*   2664*   2701*   2715*   2719*   2756*   2770*
                   2774*   2811*   2825*   2829*   2866*   2880*   2884*   3554*
SIX   = 002000      797#   1338    1396    1454    1512    2254    2309    2364    2419
SPACNT= 015531     3162*   3181    3184*   3200#
SRD   = 002000      773#
STACK = 001100      727#   1007    1283    3286    3318
STATUS  001300      928#   1267    1270
STD   = 000010      780#
STFLG   001245      919#   1010*
STPSYN= 000400      775#
SV05    015224     3130#
SWR     001100      845#   1022*   1027    1031*   1037    1046    1084    1216    1292    1305    2984    2991    3008
                   3237    3273    3281    3283    3342
SWREG   000176      836#   1031    1037    3342    3358    3365
SWREGC  016370     3353    3363#
SW00  = 000001      707#   1084
SW01  = 000002      706#   1305
SW02  = 000004      705#
SW03  = 000010      704#
SW04  = 000020      703#
SW05  = 000040      702#
SW06  = 000100      701#
SW07  = 000200      700#   1046    1216
SW08  = 000400      699#   3281
SW09  = 001000      698#   3008
SW10  = 002000      697#   3283
SW11  = 004000      696#   2991
SW12  = 010000      695#
SW13  = 020000      694#   3237
SW14  = 040000      693#   2984
SW15  = 100000      692#
SYNCNO  001176      883#   1051*   1182*   1186*   1218
SYNEXT= 020000      794#   2027    2034    2082    2089    2137    2144    2192    2199    2247    2254    2302    2309
                   2357    2364    2412    2419    2467    2474    2522    2529    2577    2584    2632    2639    2687
                   2694    2742    2749    2797    2804    2852    2859
SYNINT= 030000      793#
SYNSCH= 000020      779#   1339    1397    1455    1513    1571    1629    1687    1745    1803    1861    1919    1977
```

L 6

DZDUB-C MACY11 30A(1052) 21-NOV-78 15:36 PAGE 80
CZDUBD.P11   31-OCT-78 13:59   CROSS REFERENCE TABLE -- USER SYMBOLS   SEQ 0076

		2035	2090	2145	2200	2255	2310	2365	2420	2475	2530	2585	2640	2695
		2750	2805	2860										
SYSTST=	014000	819#												
TABLE	001232	909#	1273*	1274*	1275									
TEMP	017740	3158	3166	3191*	3319*	3320*	3502#							
TEMP1	001144	869#	1349*	1363*	1367*	1407*	1421*	1425*	1465*	1479*	1483*	1523*	1537*	1541*
		1581*	1595*	1599*	1639*	1653*	1657*	1697*	1711*	1715*	1755*	1769*	1773*	1813*
		1827*	1831*	1871*	1885*	1889*	1929*	1943*	1947*	1987*	2001*	2005*	2042*	2056*
		2060*	2097*	2111*	2115*	2152*	2166*	2170*	2207*	2221*	2225*	2262*	2276*	2280*
		2317*	2331*	2335*	2372*	2386*	2390*	2427*	2441*	2445*	2482*	2496*	2500*	2537*
		2551*	2555*	2592*	2606*	2610*	2647*	2661*	2665*	2702*	2716*	2720*	2757*	2771*
		2775*	2812*	2826*	2830*	2867*	2881*	2885*	3547*	3558	3568*	3570*	3575	3585*
		3587*												
TEMP2	001146	870#	3546*	3548*	3549*	3550*	3551	3558*	3562*	3575*	3579*			
TEMP3	001150	871#	3158*	3191	3559*	3563*	3566*	3576*	3580*	3583*				
TEMP4	001152	872#												
TEMP5	001154	873#												
THRU	003262	1260	1283#											
TISDU	001332	947#	1068	1235*										
TIVDU	001330	946#	1067	1234*										
TKCSR	001104	847#	2987	3037	3344									
TKDBR	001106	848#	2989	3039	3047	3346								
TLAST =	013610	1311	2897#											
TPCSR	001110	849#	3020	3048										
TPDBR	001112	850#	3022*	3047*										
TRPOK	015614	3228#												
TRPREG	020506	3591#												
TSTNO	001126	859#	1015*	1328*	1386*	1444*	1502*	1560*	1618*	1676*	1734*	1792*	1850*	1908*
		1966*	2024*	2079*	2134*	2189*	2244*	2299*	2354*	2409*	2464*	2519*	2574*	2629*
		2684*	2739*	2794*	2849*									
TST1	003446	1310	1316	1328#	2962	2963								
TST10	006330	1793	1850#											
TST11	006572	1851	1908#											
TST12	007034	1909	1966#											
TST13	007276	1967	2024#											
TST14	007524	2025	2079#											
TST15	007752	2080	2134#											
TST16	010200	2135	2189#											
TST17	010426	2190	2244#											
TST18	010654	2245	2299#											
TST19	011102	2300	2354#											
TST2	003710	1329	1386#											
TST20	011330	2355	2409#											
TST21	011556	2410	2464#											
TST22	012004	2465	2519#											
TST23	012232	2520	2574#											
TST24	012460	2575	2629#											
TST25	012706	2630	2684#											
TST26	013134	2685	2739#											
TST27	013362	2740	2794#											
TST28	013610	2795	2849#	2897										
TST29 =	****** U	2850												
TST3	004152	1387	1444#											
TST4	004414	1445	1502#											
TST5	004656	1503	1560#											
TST6	005120	1561	1618#											

M 6

DZDUB-C MACY11 30A(1052) 21-NOV-78 15:36 PAGE 81
CZDUBD.P11   31-OCT-78 13:59   CROSS REFERENCE TABLE -- USER SYMBOLS   SEQ 0077

```
TST7     005362        1619    1676#
TST8     005624        1677    1734#
TST9     006066        1735    1792#
TTST     014450        1289*   1290*   1300*   1301*   1303*   1304*   2985#
TXCSR    020560        1078*   1244    1330*   1332*   1335*   1341*   1342*   1344*   1345*   1388*   1390*   1393*   1399*
                       1400*   1402*   1403*   1446*   1448*   1451*   1457*   1458*   1460*   1461*   1504*   1506*   1509*
                       1515*   1516*   1518*   1519*   1562*   1564*   1567*   1573*   1574*   1576*   1577*   1620*   1622*
                       1625*   1631*   1632*   1634*   1635*   1678*   1680*   1683*   1689*   1690*   1692*   1693*   1736*
                       1738*   1741*   1747*   1748*   1750*   1751*   1794*   1796*   1799*   1805*   1806*   1808*   1809*
                       1852*   1854*   1857*   1863*   1864*   1866*   1867*   1910*   1912*   1915*   1921*   1922*   1924*
                       1925*   1968*   1970*   1973*   1979*   1980*   1982*   1983*   2026*   2028*   2031*   2037*   2038*
                       2081*   2083*   2086*   2092*   2093*   2136*   2138*   2141*   2147*   2148*   2191*   2193*   2196*
                       2202*   2203*   2246*   2248*   2251*   2257*   2258*   2301*   2303*   2306*   2312*   2313*   2356*
                       2358*   2361*   2367*   2368*   2411*   2413*   2416*   2422*   2423*   2466*   2468*   2471*   2477*
                       2478*   2521*   2523*   2526*   2532*   2533*   2576*   2578*   2581*   2587*   2588*   2631*   2633*
                       2636*   2642*   2643*   2686*   2688*   2691*   2697*   2698*   2741*   2743*   2746*   2752*   2753*
                       2796*   2798*   2801*   2807*   2808*   2851*   2853*   2856*   2862*   2863*   3532*   3545*   3551*
                       3552*   3553*   3614#
TXDBUF   020564        1080*   1246    3536*   3616#
TXDONE=  000200         809#
TXINTE=  000100         810#
TYPDAT   015774        3258    3265    3268#
TYPE  =  104402         974#   1020    1261    1276    1317    2903    2906    2911    3033    3043    3055    3059    3150
                       3187    3262    3266    3324    3350    3352    3360
TYPMSG   015744        3256    3259#
UP       003224        1273#   1278
USER  =  000000         816#
VOID  =  000001         783#
WRDCNT   015526        3160*   3189*   3198#
ZERO     014112        2917#
$CNTG    016376        3350    3366#
$E    =  000036         676#   1329    1330#   1387    1388#   1445    1446#   1503    1504#   1561    1562#   1619    1620#
                       1677    1678#   1735    1736#   1793    1794#   1851    1852#   1909    1910#   1967    1968#   2025
                       2026#   2080    2081#   2135    2136#   2190    2191#   2245    2246#   2300    2301#   2355    2356#
                       2410    2411#   2465    2466#   2520    2521#   2575    2576#   2630    2631#   2685    2686#   2740
                       2741#   2795    2796#   2850    2851#
$MNEW    016413        3354    3369#
$MSWR    016403        3352    3367#
$N    =  000034         676#   1328    1330#   1386    1388#   1444    1446#   1502    1504#   1560    1562#   1618    1620#
                       1676    1678#   1734    1736#   1792    1794#   1850    1852#   1908    1910#   1966    1968#   2024
                       2026#   2079    2081#   2134    2136#   2189    2191#   2244    2246#   2299    2301#   2354    2356#
                       2409    2411#   2464    2466#   2519    2521#   2574    2576#   2629    2631#   2684    2686#   2739
                       2741#   2794    2796#   2849    2851#   2897#
$Y    =  000015         962#    970     972#    974#    976#    978#    980#    982#    984#    986#    988#    990#    992#
                        994#    996#
$67      002142        1047    1084#
.     =  021024         821#    824#    834#    841#    927#    928#    929#    930#    931#    932#    933#    937#    938#
                        939#    940#    941#    942#    943#    944#    945#    946#    947#    948#    949#    950#    951#
                        952#    953#    954#    955#    956#    957#    958#    959#    960#   3371#   3501#   3502#   3503#
.BEGIN   002630        1082    1086    1211#
.CKSWR   016250         993    3340#
.CNTLU   016324         995    3351#
.CNVRT   015314         989    3152#
.CONVR   015310         987    3150#
.EOP     014036        2850    2903#
.ERRTA   020514        3251    3595#
```

DZDUB-C MACY11 30A(1052) 21-NOV-78 15:36 PAGE 82
CZDUBD.P11 31-OCT-78 13:59 CROSS REFERENCE TABLE -- USER SYMBOLS

```
.HLT    015634          827     3236#
.INSTE  014750          979     3053#
.INSTG  014754          3052    3055#
.INSTR  014632          977     3029#
.INST1  014652          3033#   3044    3060
.MSG    014654          3031*   3034#   3346*   3347*   3348
.PARAM  015002          981     3067#
.PFAIL  016104          825     1008    3295#   3316
.RES05  015256          985     3140#
.SAV05  015216          983     3126#
.SCOPE  014366          971     2967#
.SCOP1  014552          973     3007#
.SETFL  015534          991     3208#   3219
.START  001420          837     1006#   1016    2915
.TRPSR  015602          829     3225#
.TRPTA  001366          967#    3230
.TYPE   014572          975     3015#
```

Name														
HLT	737#	1353	1358	1374	1411	1416	1432	1469	1474	1490	1527	1532	1548	1585	1590
	1606	1643	1648	1664	1701	1706	1722	1759	1764	1780	1817	1822	1838	1875	1880
	1896	1933	1938	1954	1991	1996	2012	2046	2051	2067	2101	2106	2122	2156	2161
	2177	2211	2216	2232	2266	2271	2287	2321	2326	2342	2376	2381	2397	2431	2436
	2452	2486	2491	2507	2541	2546	2562	2596	2601	2617	2651	2656	2672	2706	2711
	2727	2761	2766	2782	2816	2821	2837	2871	2876	2892					
PRGEND	676#	2896													
PRGFRT	676#	677													
PUSSYF	676#														
RSETUP	676#	1330	1388	1446	1504	1562	1620	1678	1736	1794	1852	1910	1968	2026	2081
	2136	2191	2246	2301	2356	2411	2466	2521	2576	2631	2686	2741	2796	2851	
TSETUP	676#														
$BEGIN	676#	1207													
$BINAR	676#														
$BUFFE	676#	3498													
$CABLE	676#														
$CATCH	676#	820													
$CLRVE	676#	1022													
$CONVR	676#	3147													
$DNA	676#														
$EOP	676#	2896													
$GETFL	676#	1118	1191	1195	1199	1203	1294								
$GETPA	676#	1097	1108	1128	1154	1163	1307								
$GETSY	676#	1173													
$HEADE	676#	677													
$HLT	676#	3233													
$INSTR	676#	3026													
$ISOB	676#														
$MATCH	676#														
$MRR	676#														
$MRRW	676#														
$MRW	676#														
$MSG	676#	3372													
$PARAM	676#	3064													
$PFAIL	676#	3292													
$POKE	676#														
$POKER	676#	1343	1401	1459	1517	1575	1633	1691	1749	1807	1865	1923	1981	2036	2091
	2146	2201	2256	2311	2366	2421	2476	2531	2586	2641	2696	2751	2806	2861	
$PCNET	676#														
$RECAC	676#														
$REG	676#	3123													
$RESET	676#	1330	1332	1388	1390	1446	1448	1504	1506	1562	1564	1620	1622	1678	1680
	1736	1738	1794	1796	1852	1854	1910	1912	1968	1970	2026	2028	2081	2083	2136
	2138	2191	2193	2246	2248	2301	2303	2356	2358	2411	2413	2466	2468	2521	2523
	2576	2578	2631	2633	2686	2688	2741	2743	2796	2798	2851	2853			
$RXACT	676#														
$SCOPE	676#	2964													
$SCOP1	676#	3004													
$SETFL	676#	3202													
$SETVE	676#	821													
$START	676#	998													
$STRIP	676#														
$SYMBO	676#	689													
$SYNCR	676#	1339	1397	1455	1513	1571	1629	1687	1745	1803	1861	1919	1977		
$TRAPS	676#	962													
$TRPAR	676#														

C 7

DZDUB-C MACY11 30A(1052) 21-NOV-78 15:36 PAGE 85
CZDUBD.P11   31-OCT-78 13:59   CROSS REFERENCE TABLE -- MACRO NAMES   SEQ 0080

$TRPDE	676#	970	972	974	976	978	980	982	984	986	988	990	992	994	
$TRPSR	676#	3220													
$TSTNO	676#	1328	1386	1444	1502	1560	1618	1676	1734	1792	1850	1908	1966	2024	2079
	2134	2189	2244	2299	2354	2409	2464	2519	2574	2629	2684	2739	2794	2849	
$TYPE	676#	3012													
$UNIBU	676#														
$VARIA	676#	840													
$WORDF	676#														
$WORDO	676#	1320	1378	1436	1494	1552	1610	1668	1726	1784	1842	1900	1958	2016	2071
	2126	2181	2236	2291	2346	2401	2456	2511	2566	2621	2676	2731	2786	2841	
$WORDP	676#														

. ABS. 021024   000

ERRORS DETECTED: 0

CZDUBD.BIN,CZDUBD.SEQ/CRF/SOL/NL:TOC=CZDU11.HLO/EQ:RUNB,CZDU11.PAR,CZDU11.KET,CZDUBD.P11
RUN-TIME: 8 12 1 SECONDS
RUN-TIME RATIO: 92/22=4.0
CORE USED: 19K (37 PAGES)