

DU11

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

- - - - -

PRODUCT CODE: AC-8680D-MC

PRODUCT NAME: CZDUBDO DU11 OFFLINE RECEIVER TESTS

RELEASE DATE: JUN 1978

MAINTAINER : DIAGNOSTICS

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OF RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973, 1978 BY DIGITAL EQUIPMENT CORPORATION

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 SYNCHRONOUS/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

<

4.3.3.10 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

4.3.3.16 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 ? ANDDO 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 'F' IS STRUCK ON THE TELETYPE, THE PROGRAM
WILL FREEZE ON THE NEXT TEST UNTIL A KEY IS STRUCK ON
THE TELETYPE & 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 WILL NOT 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 ADDRESSES
 &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.
 - 7.2.2.2ORLOAD 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 000200
- 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 DUPR: LEVEL 5
OF SYNC CHARS SELECTED - 2 SYNCND: 377
IS SEC XMIT JUMPER # 6 IN ?- YES SEXMIT: 377
IS SEC REC JUMPER # 5 IN ?- YES SREC: 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

DZDUB-C MACY11 30A(1052) 21-NOV-78 15:36 PAGE 15
CZDU11.PAR 21-NOV-78 15:17

B 2

SEQ 0014

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
```



```

821                                     ;STANDARD INTERRUPT VECTORS
822
823
824                                     .-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                                     .-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                                     .-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 ITERATIONS 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 000
```

877 001164 000000
878 001166 000000
879 001170 000000
880 001172 000000
881 001174 000000

SAVR3: 0
SAVR4: 0
SAVR5: 0
SAVSP: 0
SAVPC: 0

:R3 STORAGE
:R4 STORAGE
:R5 STORAGE
:STACK POINTER STORAGE
: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 .EVEN
923
924
925 ;***** STATUS MAP *****
926
927 001300 001300 STATUS: .=1300
928 001300 000001 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				

```

994      104414
995 001416 016324
996      .CNTLU
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      ,SYNCRNO      ; 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      ; OPT'ONAL 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 @#PRTDU    ,@#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 BUFRXD     ,RXDBUF     ;
1076 002072 016767 177252 016454  MOV CSRRPAR    ,PARCSR     ; HRXDBUF
1077 002100 016767 177246 016450  MOV CSRHPAR    ,HPARCSR    ; PARCSR
1078 002106 016767 177242 016444  MOV CSRTX      ,TXCSR      ; CSRHPAR
1079 002114 016767 177236 016440  MOV CSRHTX     ,HTXCSR     ; CSRTX
1080 002122 016767 177232 016434  MOV BUFTXD     ,TXDBUF     ; CSRHTX
1081 002130 016767 177226 016430  MOV BUFTXD     ,TXDBUF     ; BUFTXD
1082 002136 000167 000466  JMP .BEGIN      ; HTXDBUF
1083      ;*****
1084 002142 032777 000001 176730 $67: BIT #SW00 ,@SWR      ; 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$: CLR @#FLAG      ; CLEAR FLAG SO STATUS MAP PRINTS OUT
1088 002162 012700 000300  MOV #300,R0      ; RESTORE VECTOR AREA TO TRAPCATCHER
1089 002166 012701 000302  MOV #302,R1      ; START AT LOCATION 300
1090 002172 012702 000004  MOV #4,R2
1091 002176 010110 2$: 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
1096 002212 002771  BLT 2$      ; END AT LOCATION 776
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

```



```

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 00
```

1218	002646	116767	176324	176424	MOV8 SYNCNO	,NOSYNC	; SYNC CHARS
1219	002654	116767	176317	176417	MOV8 SEXMIT	,MITSEX	; XMIT JUMPER
1220	002662	116767	176312	176412	MOV8 SEREC	,RESEC	; SEC REC JUMPER
1221	002670	116767	176305	176405	MOV8 OPTCLR	,CLROPT	; OPTIONAL JUMPER
1222	002676	116767	176300	176400	MOV8 MULTD	,DMULT	; MULTIPLE DEVICES
1223	002704	116767	176273	176373	MOV8 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	,ADDRROT	; ROTATING POINTER
1231	002764	013737	020110	001324	MOV @#DUPRT	,@#PRTDU	; DU11 PRIORITY
1232	002772	016767	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	

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						: (OVRUN, RXERR)		
1324						: MODE: ISYMOD		
1325						: LENGTH: SIX		
1326						: CHAR: 25		
1327						: :		
1328	003446	012767	000001	175452	TST1:	MOV #*, TSTNO		: SAVE THIS
1329	003454	012767	003710	175434		MOV #TST2, NEXT		: GO TO THIS TEST WHEN THRU


```

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,@TXCSR ;MASTER RESET
1389 003732 012777 000000 014614 MOV #ISYMOD,@PARCSR ;SET THE MODE
1390 003740 052777 000400 014612 BIS #MRESET,@TXCSR ;MASTER RESET
1391
1392 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1393 003746 012777 064001 014604 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1394
1395 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1396 003754 012777 002000 014572 MOV #ISYMOD!SIX!NOPAR!0,@PARCSR
1397 003762 052777 000020 014554 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
1398 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1399 003770 042777 020000 014562 BIC #CLK,@TXCSR ;POKE CLK DOWN
1400 003776 052777 020000 014554 BIS #CLK,@TXCSR ;POKE CLK UP
1401 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1402 004004 042777 020000 014546 BIC #CLK,@TXCSR ;POKE CLK DOWN
1403 004012 052777 020000 014540 BIS #CLK,@TXCSR ;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 @RXCSR ;RXDONE ?
1410 004054 100401 BMI 64$
1411 004056 104000 HLT ;RXDONE SHOULD BE SET
1412 004060
1413 004060 017701 014464 64$: MOV @RXDBUF,R1 ;ACTUAL
1414 004064 020001 CMP R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1415 004066 001401 BEQ 65$
1416 004070 104002 HLT 2 ;RECEIVED DATA DID NOT MATCH
1417 ;EXPECTED DATA - CHECK MAINT DATA
1418 ;OR RECEIVER LOGIC
1419 004072
1420 004072 012767 000010 175040 65$: MOV #8,,SHIFT ;# OF SHIFTS
1421 004100 012767 000324 175036 MOV #324,TEMP1 ;DATA CHAR
1422 004106 004767 014136 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1423 ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1424 004112 012767 000010 175020 MOV #8,,SHIFT ;# OF SHIFTS
1425 004120 012767 000324 175016 MOV #324,TEMP1 ;DATA CHAR
1426 004126 004767 014116 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1427 004132 012700 140052 MOV #140000!52,R0 ;EXPECTED DATA PLUS
1428 ;RXERR & OVRRUN
1429 004136 017701 014406 MOV @RXDBUF,R1 ;ACTUAL
1430 004142 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
1431 004144 001401 BEQ 66$
1432 004146 104002 HLT 2 ;SPECIFICALLY LOOK AT RXERR &
1433 ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1434 004150
1435 004150 104400 66$: 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:IS,MOD
1441 ;:LENGTH:SIX

```



```

1554      ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1555      ::(OVRUN,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
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$:
1594 005040 012767 000011 174072      MOV    #9,SHIFT      ;# OF SHIFTS
1595 005046 012767 000652 174070      MOV    #652,TEMP1    ;DATA CHAR
1596 005054 004767 013170      JSR    PC,RPOKE      ;SHIFT IN THIS CHAR
1597      ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1598 005060 012767 000011 174052      MOV    #9,SHIFT      ;# OF SHIFTS
1599 005066 012767 000652 174050      MOV    #652,TEMP1    ;DATA CHAR
1600 005074 004767 013150      JSR    PC,RPOKE      ;SHIFT IN THIS CHAR
1601 005100 012700 140125      MOV    #140000!125,R0 ;EXPECTED DATA PLUS
1602      ;RXERR & OVRUN
1603 005104 017701 013440      MOV    @RXDBUF,R1 ;ACTUAL
1604 005110 020001      CMP    R0,R1 ;COMPARE EXP VS. ACT
1605 005112 001401      BEQ    66$
1606 005114 104002      HLT    2 ;SPECIFICALLY LOOK AT RXERR &
1607      ;OVRUN BITS...THEY BOTH SHOULD BE SET
1608 005116      66$:
1609 005116 104400      SCOPE

```



```

1666 005360
1667 005360 104400
1668
1669
1670
1671
1672
1673
1674
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
1703 005532 017701 013012 64$: MOV @RXDBUF,R1 ;ACTUAL
1704 005536 020001 CMP R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1705 005540 001401 BEQ 65$
1706 005542 104002 HLT 2 ;RECEIVED DATA DID NOT MATCH
1707 ;EXPECTED DATA - CHECK MAINT DATA
1708 ;OR RECEIVER LOGIC
1709
1710 005544 012767 000011 173366 65$: MOV #9,,SHIFT ;# OF SHIFTS
1711 005552 012767 000776 173364 MOV #776,TEMP1 ;DATA CHAR
1712 005560 004767 012464 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1713 ;NOW SH!FT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1714 005564 012767 000011 173346 MOV #9,,SHIFT ;# OF SHIFTS
1715 005572 012767 000776 173344 MOV #776,TEMP1 ;DATA CHAR
1716 005600 004767 012444 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1717 005604 012700 140177 MOV #140000!177,R0 ;EXPECTED DATA PLUS
1718 ;RXERR & OVRRUN
1719 005610 017701 012734 MOV @RXDBUF,R1 ;ACTUAL
1720 005614 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
1721 005616 001401 BEQ 66$

```

```

1722 005620 104002          HLT      2          ;SPECIFICALLY LOOK AT RXERR &
1723                                     ;OVRUN 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                                     ;;(OVRUN,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 00566
```

```

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                                     ;OVRUN 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                                     ;;(OVRUN,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
1819 006236 017701 012306      64$:  MOV      @RXDBUF,R1 ;ACTUAL
1820 006242 020001      CMP      R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1821 006244 001401      BEQ      65$
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 & OVRUN
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
1839                                     ;SPECIFICALLY LOOK AT RXERR &
1840                                     ;OVRUN BITS...THEY BOTH SHOULD BE SET
1841 006326 104400      66$:
1842                                     SCOPE
1843                                     ::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1844                                     ::RECEIVER SECTION,IT USES THE ERROR FLAGS
1845                                     ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1846                                     ::(OVRUN,RXERR)
1847                                     ::MODE:ISYMOD
1848                                     ::LENGTH:EIGHT
1849                                     ::CHAR:252
1850 006330 012767 000012 172570  ST10: 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 SYNCHRONIZATION....
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 SYNCHRONIZATION
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
1877 006500 017701 012044      64$:
1878 006504 020001      MOV    @RXDBUF,R1      ;ACTUAL
1879 006506 001401      CMP    R0,R1      ;COMPARE EXPECTED VS. ACTUAL
1880 006510 104002      BEQ    65$
1881                                     ;RECEIVED DATA DID NOT MATCH
1882                                     ;EXPECTED DATA - CHECK MAINT DATA
1883                                     ;OR RECEIVER LOGIC
1884 006512 012767 000012 172420      65$:
1885 006520 012767 001524 172416      MOV    #10,SHIFT ;# OF SHIFTS
1886 006526 004767 011516      MOV    #1524,TEMP1 ;DATA CHAR
1887                                     JSR    PC,RPOKE ;SHIFT IN THIS CHAR
1888                                     ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1889 006532 012767 000012 172400      MOV    #10,SHIFT ;# OF SHIFTS
1890 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 & OVRUN
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							:OVRUN BITS...THEY BOTH SHOULD BE SET
1898	006570			66\$:			
1899	006570	104400					
1900						SCOPE	
1901						::THIS TEST VERIFYS WORD LENGTH SELECT OF THE	
1902						::RECEIVER SECTION,IT USES THE ERROR FLAGS	
1903						::TO DETERMINE THAT IT WAS SELECTED CORRECTLY	
1904						::(OVRUN,RXERR)	
1905						::MODE:ISYMOD	
1906						::LENGTH:EIGHT	
1907						::CHAR:377	
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 SYNCHRONIZATION
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 & OVRUN
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							:OVRUN BITS...THEY BOTH SHOULD BE SET
1956	007032				66\$:		
1957	007032	104400					
1958					SCOPE		
1959					::THIS TEST VERIFYS WORD LENGTH SELECT OF THE		
1960					::RECEIVER SECTION,IT USES THE ERROR FLAGS		
1961					::TO DETERMINE THAT IT WAS SELECTED CORRECTLY		
1962					::(OVRUN,RXERR)		
1963					::MODE:ISYMOD		
1964					::LENGTH:EIGHT		
1965					::CHAR:0		
1966	007034	012767	000014	172064	TST12: MOV	#12,TSTNO	:SAVE THIS
1967	007042	012767	007276	172046	MOV	#TST13,NEXT	:GC 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 SYNCHRONIZATION		
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							
2004	007236	012767	000012	171674			
2005	007244	012767	001000	171672			
2006	007252	004767	010772		JSR	PC,RPOKE	:SHIFT IN THIS CHAR
2007	007256	012700	140000		MOV	#140000!0,R0	:EXPECTED DATA PLUS
2008							
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							:OVRUN BITS...THEY BOTH SHOULD BE SET
2014	007274						
2015	007274	104400					
2016							
2017							
2018							
2019							
2020							
2021							
2022							
2023							
2024	007276	012767	000015	171622	TST13:	MOV	#13,TSTNO
2025	007304	012767	007524	171604		MOV	#TST14,NEXT
2026	007312	052777	000400	011240		BIS	#MRESET,@TXCSR
2027	007320	012777	020000	011226		MOV	#SYNEXT,@PARCSR
2028	007326	052777	000400	011224		BIS	#MRESET,@TXCSR
2029							
2030							
2031	007334	012777	064001	011216			
2032							
2033							
2034	007342	012777	020000	011204			
2035	007350	052777	000020	011166			
2036							
2037	007356	042777	020000	011174			
2038	007364	052777	020000	011166			
2039	007372	016703	011152				
2040	007376	012700	000025				
2041	007402	012767	000005	171530			
2042	007410	012767	000025	171526			
2043	007416	004767	010626				
2044	007422	105777	011116				
2045	007426	100401					
2046	007430	104000					
2047	007432						
2048	007432	017701	011112				
2049	007436	020001					
2050	007440	001401					
2051	007442	104002					
2052							
2053							
2054	007444						
2055	007444	012767	000005	171466			
2056	007452	012767	000025	171464			
2057	007460	004767	010564				

;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
 :# OF SHIFTS
 :DATA CHAR
 :SHIFT IN THIS CHAR
 :EXPECTED DATA PLUS
 :RXERR & OVRUN
 :ACTUAL
 :SPECIFICALLY LOOK AT RXERR &
 :OVRUN BITS...THEY BOTH SHOULD BE SET
 66\$:
 SCOPE
 ::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 ::RECEIVER SECTION,IT USES THE ERROR FLAGS
 ::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 ::(OVRUN,RXERR)
 ::MODE:SYNEXT
 ::LENGTH:FIVE
 ::CHAR:25
 :SAVE THIS
 :GO TO THIS TEST WHEN THRU
 :MASTER RESET
 :SET THE MODE
 :MASTER RESET
 :SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 :SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 :SET SEARCH SYNC
 :POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
 :POKE CLK DOWN
 :POKE CLK UP
 :SET UP FOR ERROR MESSAGE
 :EXPECTED
 :# OF SHIFTS
 :DATA CHAR
 :SHIFT IN THIS CHAR
 :RXDONE ?
 64\$:
 :RXDONE SHOULD BE SET
 :ACTUAL
 :COMPARE EXPECTED VS. ACTUAL
 65\$:
 :RECEIVED DATA DID NOT MATCH
 :EXPECTED DATA - CHECK MAINT DATA
 :OR RECEIVER LOGIC
 :# OF SHIFTS
 :DATA CHAR
 :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 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 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

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	

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					
2291						SCOPE	
2292						::THIS TEST VERIFYS WORD LENGTH SELECT OF THE	
2293						::RECEIVER SECTION,IT USES THE ERROR FLAGS	
2294						::TO DETERMINE THAT IT WAS SELECTED CORRECTLY	
2295						::(OVRRUN,RXERR)	
2296						::MODE:SYNEXT	
2297						::LENGTH:SIX	
2298						::CHAR:52	
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	000400	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 SYNCHRONIZATION
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 MAINI 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							

2450	011546	020001				CMP	R0,R1	:COMPARE EXP VS. ACT
2451	011550	001401				BEQ	66\$	
2452	011552	104002				HLT	2	:SPECIFICALLY LOOK AT RXERR & :OVRUN BITS...THEY BOTH SHOULD BE SET
2453								
2454	011554				66\$:			
2455	011554	104400						
2456								SCOPE
2457								::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2458								::RECEIVER SECTION,IT USES THE ERROR FLAGS
2459								::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2460								::(OVRUN,RXERR)
2461								::MODE:SYNEXT
2462								::LENGTH:SEVEN
2463								::CHAR:125
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	#PRESET,@TXCSR	:MASTER RESET
2467	011600	012777	020000	006746		MOV	#SYNEXT,@PARCSR	:SET THE MODE
2468	011606	052777	000400	006744		BIS	#PRESET,@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 SYNCHRONIZATION
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							


```

2618                                     ;OVRUN BITS...THEY BOTH SHOULD BE SET
2619 012456
2620 012456 104400
2621
2622
2623
2624
2625
2626
2627
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
2636 012516 012777 064001 006034 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2637 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2638
2639 012524 012777 024000 006022 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2640 012532 
```

```

2674 012704
2675 012704 104400
2676
2677
2678
2679
2680
2681
2682
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
2697 ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2698 012766 042777 020000 005564 BIC #CLK,@TXCSR ;POKE CLK DOWN
2699 012774 052777 020000 005556 BIS #CLK,@TXCSR ;POKE CLK UP
2700 013002 016703 005542 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2701 013006 012700 000125 MOV #125,R0 ;EXPECTED
2702 013012 012767 000010 166120 MOV #8,SHIFT ;# OF SHIFTS
2703 013020 012767 000125 166116 MOV #125,TEMP1 ;DATA CHAR
2704 013026 004767 005216 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2705 013032 105777 005506 TSTB @RXCSR ;RXDONE ?
2706 013036 100401 BMI 64$
2707 013040 104000 HLT ;RXDONE SHOULD BE SET
2708 013042 017701 005502 64$: MOV @RXDBUF,R1 ;ACTUAL
2709 013046 020001 CMP R0,R1 ;COMPARE EXPECTED VS. ACTUAL
2710 013050 001401 BEQ 65$
2711 013052 104002 HLT 2 ;RECEIVED DATA DID NOT MATCH
2712 ;EXPECTED DATA - CHECK MAINT DATA
2713 ;OR RECEIVER LOGIC
2714 013054 65$:
2715 013054 012767 000010 166056 MOV #8,SHIFT ;# OF SHIFTS
2716 013062 012767 000125 166054 MOV #125,TEMP1 ;DATA CHAR
2717 013070 004767 005154 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2718 ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2719 013074 012767 000010 166036 MOV #8,SHIFT ;# OF SHIFTS
2720 013102 012767 000125 166034 MOV #125,TEMP1 ;DATA CHAR
2721 013110 004767 005134 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2722 013114 012700 140125 MOV #140000!125,R0 ;EXPECTED DATA PLUS
2723 ;RXERR & OVRRUN
2724 013120 017701 005424 MOV @RXDBUF,R1 ;ACTUAL
2725 013124 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
2726 013126 001401 BEQ 66$
2727 013130 104002 HLT 2 ;SPECIFICALLY LOOK AT RXERR &
2728 ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2729 013132 66$:

```



```

2896
2897
2898
2899
2900
2901
2902
2903 014036 104402
2904 014040 017200
2905 014042 104410 014274
2906 014046 104402 016721
2907 014052 105767 165124
2908 014056 001511
2909 014060 005767 165132
2910 014064 001007
2911 014066 104402 016733
2912 014072 016700 165120
2913 014076 000000
2914
2915 014100 000167 165314
2916 014104 062767 000010 165072 RUNIT:
2917 014112 062767 000010 165072 ZERO:
2918 014120 000241
2919 014122 006167 165072
2920 014126 103410
2921
2922 014130 036767 165064 165060
2923 014136 001762
2924 014140 004767 000034
2925 014144 000167 000174
2926 014150 012767 000001 165042 2$:
2927
2928 014156 016767 165024 165020
2929 014164 016767 165024 165020
2930 014172 004767 000002
2931 014176 000441
2932 014200 016767 165000 004040 REPLAY:
2933 014206 004767 003702
2934 014212 016767 164774 004350
2935 014220 062767 000002 164764
2936 014226 016767 164760 004336
2937 014234 062767 000002 164750
2938 014242 016767 164744 004324
2939 014250 062767 000002 164734
2940 014256 016767 164730 004312
2941 014264 016767 004300 164720
2942 014272 000207
2943
2944 014274 000001
2945 014276 006 002
2946 014300 020544
2947
2948 014302
2949 014302 005067 164626
2950 014306 005067 164734
2951 014312 005267 164612

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

.EOP: TYPE
MEPASS
CONVRT ,OUTCRY
TYPE ,DEVICE
TSTB MULTD
BEQ CCC
TST ACTREG
BNE RUNIT
TYPE ,MCOW
MOV ACTREG,RO
HALT
;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
JMP .START
ADD #10,BASEADD
ADD #10,BASEIV
CLC
ROL ROTADD
BCS 2$
BIT ROTADD,ACTREG
BEQ RUNIT
JSR PC,REPLAY
JMP RESTRT
MOV #1,ROTADD
MOV KEEPADD,BASEADD
MOV KEEPIV,BASEIV
JSR PC,REPLAY
BR CCC
MOV BASEADD,DUBASE
JSR PC,DUADDR
MOV BASEIV,DURIV
ADD #2,BASEIV
MOV BASEIV,DURIS
ADD #2,BASEIV
MOV BASEIV,DUTIV
ADD #2,BASEIV
MOV BASEIV,DUTIS
MOV DURIV,BASEIV
RTS PC

OUTCRY: 1
.BYTE 6,2
RXCSR

CCC: CLR LSTERR
CLR ER?FLG
INC PASCNT

;TYPE NAME OF TEST
;ARE YOU RUNNING MULTIPLE DEVICES ?
;NO, JUMP AROUND
;ARE ANY DEVICES ACTIVE ?
;YES
;NO
;DISPLAY ACTREG
;SELECT SOMETHING TO RUN @ ACTREG:
;START OVER AGAIN.....YOU DESELECTED EVERYTHING
;NEXT BLOCK (ADDRESSES)
;NEXT BLOCK (VECTORS)
;UP DATE ROTATING POINTER
;IS IT THE LAST DEVICE
;TO BE TESTED IN THIS PASS ?
;TEST THIS DEVICE FOR ACTIVE STATUS
;IF NOT ACTIVE, TRY NEXT ADDRESS
;CALCULATE NEW PARAMETERS
;YES IT WAS ACTIVE, TEST THIS DEVICE
;OK!, NOW SET UP ROTATING
;POINTER FOR NEXT MULTIPLE PASS
;RESTORE BASE ADDRESS
;RESTORE BASE INTERRUPT VECTORS
;CALC NEW PARAMETERS
;JUMP AROUND REPLAY
;SET UP FOR NEW ADDRESSES
;CREATE NEW ADDRESSES
;CREATE DURIV
;CREATE DURIS
;CREATE DUTIV
;CREATE DUTIS
;RESTORE

```

```

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$
2970
2971 014370 013746 000004              MOV    @#4, -(SP)
2972 014374 012737 014414 000004      MOV    #1$, @#4
2973 014402 005737 177060              TST    @#177060
2974 014406 012637 000004      MOV    (SP)+, @#4
2975 014412 000404      BR     2$
2976 014414 022626      1$:  CMP    (SP)+, (SP)+
2977 014416 012637 000004      MOV    (SP)+, @#4
2978 014422 000403      BR     3$
2979 014424 016767 164466 164462  2$:  MOV    NEXT, RTRN
2980 014432 016716 164456      3$:  MOV    RTRN, (SP)
2981 014436 000002      RTI
2982 014440      4$:  .***** END OF CODE FOR THE X OR TESTER *****
2983 014440 104413      CKSWR          ;CHECK FOR ^G
2984 014442 032777 040000 164430      BIT    #SW14, @SWR
2985 014450 001407      TTST:  BEQ     1$
2986 014452 000432      BR     3$
2987 014454 105777 164424      TSTB   @TKCSR
2988 014460 100027      BPL     3$
2989 014462 017700 164420      MOV    @TKDBR, R0
2990 014466 000412      BR     2$
2991 014470 032777 004000 164402  1$:  BIT    #SW11, @SWR
2992 014476 001006      BNE     2$
2993 014500 005267 164420      INC    LPCNT
2994 014504 026767 164414 164410      CMP    LPCNT, ICOUNT
2
```


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		CPMB #15,(R4)
3079	015054	001420			BEQ PARERR
3080	015056	121427	000060	\$:	CPMB (R4),#60
3081	015062	002415			BLT PARERR
3082	015064	121427	000067		CPMB (R4),#67
3083	015070	003012			BGT PARERR
3084	015072	142714	000060		BICB #60,(R4)
3085	015076	152405			BISB (R4)+,R5
3086	015100	122714	000015		CPMB #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:	CPMB #15,(R4)
3093	015122	001003			BNE 120\$
3094	015124	005737	016246		TST WORDSW
3095	015130	001023			BNE PARTI
3096	015132	104404		120\$:	INSTR
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		_CLIM:	0
3119	015210	000000		HILIM:	0

:IS FIRST CHARACTER A <CR>
 :IS CKSWR ROUTINE BEING USED

```

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
```


3232	015632	000136		JMP	@(SP)+	;GO TO SUBROUTINE
3233						
3234						
3235						
3236	015634	104413		.HLT:	CKSWR	;CHECK FOR ^G
3237	015636	032777	020000	163234	BIT	#SW13,@SWR
3238	015644	001061			BNE	HALTS
3239	015646	021667	163262		CMP	(SP),LSTERR
3240	015652	001404				

```

3288 016074 000002      2$: RTI
3289 016076 000001      ERTAB: 1
3290 016100 006        .BYTE 6,2
3291 016102 001174      SAVPC
3292                                     ;ENTER HERE ON POWER FAILURE
3293
3294
3295 016104 010046      .PFAIL: MOV R0,-(SP) ;SAVE R0-R5 ON PROCESSOR STACK
3296 016106 010146      MOV R1,-(SP)
3297 016110 010246      MOV R2,-(SP)
3298 016112 010346      MOV R3,-(SP)
3299 016114 010446      MOV R4,-(SP)
3300 016116 010546      MOV R5,-(SP)
3301 016120 016746 161700 MOV 24,-(SP)
3302 016124 010667 163042 MOV SP,SAVSP ;SAVE STACK POINTER
3303 016130 012767 016142 161666 MOV #RESTART,24 ;SET UP FOR POWER UP TRAP
3304 016136 000000      HALT ;HALT ON POWER DOWN NORMAL
3305 016140 000777      1$: 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        .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 #CJREG,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	@WRDSW		
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	@WRDSW		
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	051040	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	046125	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: .ASCIIZ /=DEVICE /
3407	016726	042503	020040	000	
3408	016733	015	044012	053517	MCOW: .ASCIIZ <15><12>/HOW NOW BROWN COW? ...SELECT SOMETHING TO RUN @ACTREG/
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: .ASCIIZ <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: .ASCIIZ / ?/
3428	017110	005015	000		MCRLF: .ASCIIZ <15><12>
3429	017113	040	050040	053517	MPFAIL: .ASCIIZ / 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: .ASCIIZ <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: .ASCIIZ <15><12>/R/
3443	017231	015	052012	051505	MTSTPC: .ASCIIZ <15><12>/TEST PC-/
3444	017236	020124	041520	000055	
3445	017244	005015	047514	045503	MLOCK: .ASCIIZ <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: .ASCIIZ <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: .ASCIIZ <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
 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
 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
 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
 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
 3493 017654 052123 052101 051525
 3494 017662 020040 046440 050101
 3495 017670 020040 020040 005040
 3496 017676 000015
 3497
 3498
 3499
 3500
 3501 017700 000040
 3502 017740 000040
 3503 020000 000040
 3504
 3505
 3506
 3507
 3508
 3509 020040 006367 000044
 3510 020044 006367 000040
 3511 020050 006367 000034

MWIRE6: .ASCIIZ <15><12>/IS SEC XMIT JUMPER #6 IN? (Y OR N)-/

MWIRE5: .ASCIIZ <15><12>/IS SEC REC JUMPER #5 IN? (Y OR N)-/

MWIRE4: .ASCIIZ <15><12>/IS OPT CLR ENABLE JUMPER #4 IN? (Y OR N)-/

MEXTJ: .ASCIIZ <15><12>/IS THE TEST CONNECTOR INSTALLED ?(Y OR N)-/

MSTATUS: .ASCIIZ <12> <15>/ STATUS MAP / <12> <15>

.EVEN

;BUFFERS FOR INPUT-OUTPUT

INBUF: .BLKB 40

TEMP: .BLKB 40

MDATA: .BLKB 40

;*****^*****

;UTILITIES

;*****

;THIS UTILITY CALCULATES PRIORITY LEVEL

DULEV: ASL DUPRT ;SHIFT LEFT

ASL DUPRT ;

ASI 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$

```


