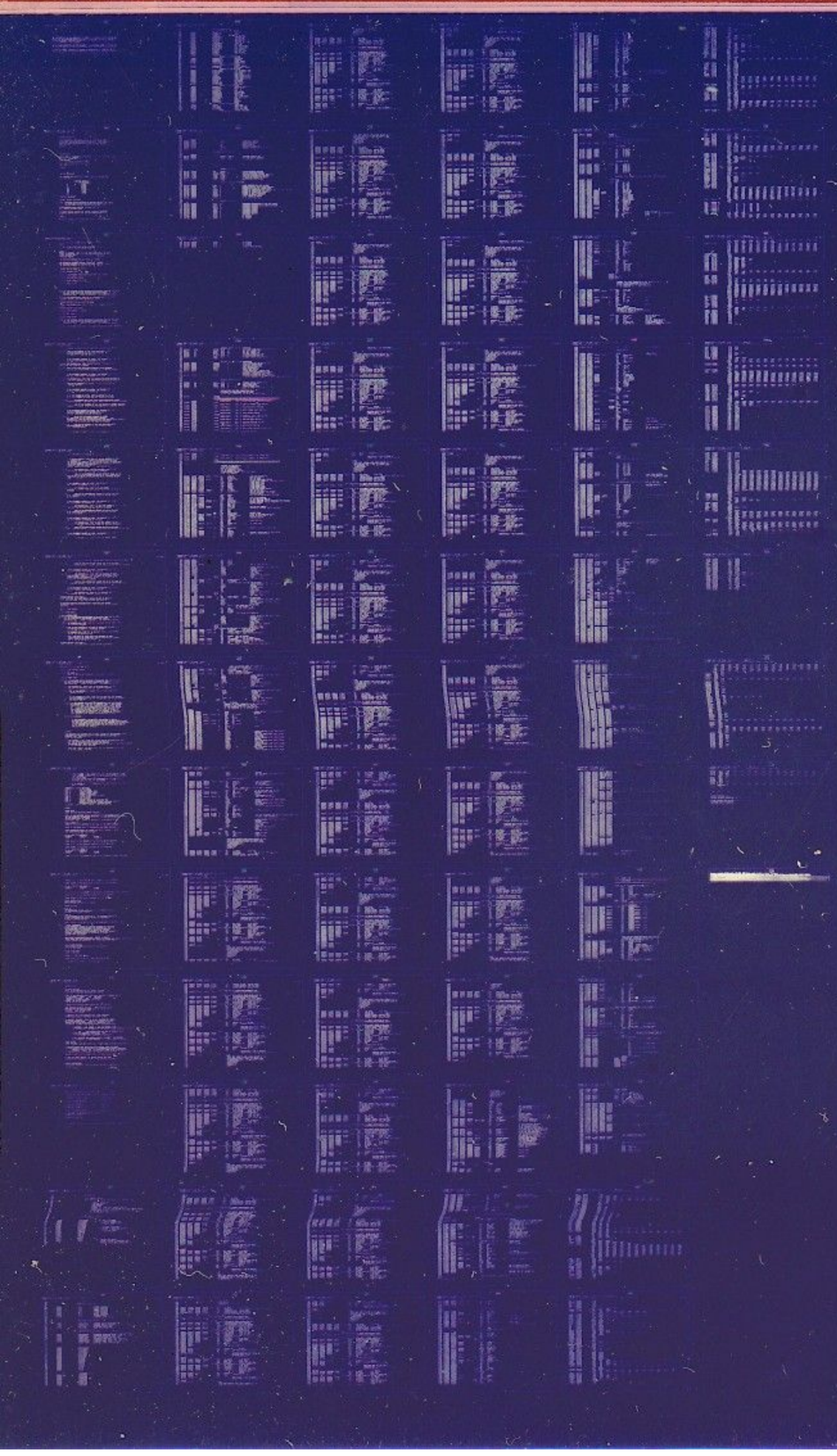
DU11

OFF-LINE RECEIVER TESTS
MD-11-DZDUB-C

EP-DZDUB-C-DL-A

FICHE 1 OF 1

NOV 1976
digital
MADE IN USA



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.

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

3. LOADING PROCEDURE

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

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

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

NOTE: 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

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 SWD1=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 SWDD=1 AND ANSWER "NO" TO THE MULTIPLE DEVICE QUESTION SEE 4.3.3

4.3.4.1 LOAD 000200

4.3.4.2 SET SWD1=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 SWD2 =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

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
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 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 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

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

11. LISTINGS

```
                                .ENABLE ABS

                                ;DU11 DZDUB-C TAPE B
                                ;COPYRIGHT 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                        ;STARTING PROCEDURE
                                        ;LOAD PROGRAM
                                        ;PRESS START
                                        ;PROGRAM WILL TYPE "DU11 DZDUB-C TAPE B "
                                        ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                                        ;AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE B"
                                        ;AND THEN RESUME TESTING


                                ;SWITCH REGISTER OPTIONS

  100000                        SW15=100000             ;=1,HALT ON ERROR.
  040000                        SW14=40000              ;=1,LOOP ON CURRENT TEST
  020000                        SW13=20000              ;=1,INHIBIT ERROR TYPEOUT
  010000                        SW12=10000
  004000                        SW11=4000               ;=1,INHIBIT ITERATIONS
  002000                        SW10=2000               ;=1,ESCAPE TO NEXT TEST ON ERROR
  001000                        SW09=1000               ;=1,LOOP WITH CURRENT DATA
  000400                        SW08=400                ;=1,LOOP ON ERROR
  000100                        SW06=100
  000040                        SW05=40
  000020                        SW04=20
  000010                        SW03=10
  000004                        SW02=4                  ;LOCK ON TEST SELECT
  000002                        SW01=2                  ;RESTART PROGRAM AT SELECTED TEST
  000001                        SW00=1                  ;RESELECT VECTOR AND CONTROL REGISTER
                                                        ;ADDRESS AFTER PROGRAM RESTART
```

```
                                ;REGISTER DEFINITIONS
                                
000000                          R0=%0           ;GENERAL REGISTER
000001                          R1=%1           ;GENERAL REGISTER
000002                          R2=%2           ;GENERAL REGISTER
000003                          R3=%3           ;GENERAL REGISTER
000004                          R4=%4           ;GENERAL REGISTER
000005                          R5=%5           ;GENERAL REGISTER
000006                          SP=%6           ;PROCESSOR STACK POINTER
000007                          PC=%7           ;PROGRAM COUNTER
                                
                                ;LOCATION EQUIVALENCIES
                                
177570                          DSWR=177570     ;HARDWARE SWITCH REGISTER LOC.
177570                          DLIGHTS=177570  ;HARDWARE DISPLAY REGISTER LOC.
177776                          PS=177776       ;PROCESSOR STATUS WORD
001100                          STACK=1100      ;START OF PROCESSOR STACK
                                
                                ;INSTRUCTION DEFINITIONS
                                
005746                          PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
005726                          POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
010046                          PUSHR0=10046    ;SAVE R0 ON STACK =MOV R0,-(SP)
012600                          POPR0=12600     ;RESTORE R0 FROM STACK =MOV (SP)+,R0
024646                          PUSH2SP=24646   ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
022626                          POP2SP=22626    ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
                                .EQUIV  EMT,HLT ;BASIC DEFINITION OF ERROR CALL
                                
                                
100000                          BIT15=100000
040000                          BIT14=40000
020000                          BIT13=20000
010000                          BIT12=10000
004000                          BIT11=4000
002000                          BIT10=2000
001000                          BIT9=1000
000400                          BIT8=400
000200                          BIT7=200
000100                          BIT6=100
000040                          BIT5=40
000020                          BIT4=20
000010                          BIT3=10
000004                          BIT2=4
000002                          BIT1=2
000001                          BIT0=1
                                
                                ;PROCESSER LEVELS
000340                          LEVEL7=340
000300                          LEVEL6=300
000240                          LEVEL5=240
000200                          LEVEL4=200
000140                          LEVEL3=140
000100                          LEVEL2=100
000040                          LEVEL1=040
000000                          LEVEL0=000
```

```
                        ;REGISTER DEFINITIONS
                                ;RXCSR BIT DEFINITIONS
100000                  DSC=BIT15               ;DATA SET CHANGE
040000                  RING=BIT14              ;RING
020000                  CTS=BIT13               ;CLR TO SEND
010000                  CARDET=BIT12            ;CARRIER DETECT
004000                  RECACT=BIT11            ;REC ACTIVE
002000                  SRD=BIT10               ;SEC REC DATA
001000                  DSR=BIT9                ;DATA SET RDY
000400                  STPSYN=BIT8             ;STRIP SYNC
000200                  RXDONE=BIT7             ;REC DONE
000100                  RINTEN=BIT6             ;REC INTR ENABLE
000040                  DSINTE=BIT5             ;DSC INTR ENABLE
000020                  SYNSCH=BIT4             ;SYNC SEARCH
000010                  STD=BIT3                ;SEC XMIT DATA
000004                  RTS=BIT2                ;REQ TO SEND
000002                  DTR=BIT1                ;DATA TERM RDY
000001                  VOID=BIT0
                                ;RXDBUF BIT DEFINITIONS
100000                  RXERR=BIT15             ;REC ERROR
040000                  OVRRUN=BIT14            ;OVERRUN
020000                  FRMERR=BIT13            ;FRAME ERROR
010000                  PARER=BIT12             ;PARITY ERROR
                                ;PARCSR BIT DEFINITIONS
001000                  PAREN=BIT9              ;PARITY ENABLE
000400                  EVPAR=BIT8              ;EVEN PARITY SENSE
                                ;PARCSR WRD DEFINITIONS
030000                  SYNINT=30000            ;SYNC EXTERNAL MODE
020000                  SYNEXT=20000            ;SYNC INTERNAL MODE
000000                  ISYMOD=0                ;ISOC MODE
000000                  FIVE=0                  ;WORD LENGTH 5 BITS
002000                  SIX=2000                ;WORD LENGTH 6 BITS
004000                  SEVEN=4000              ;WORD LENGTH 7 BITS
006000                  EIGHT=6000              ;WORD LENGTH 8 BITS
000000                  NOPAR=0                 ;NO PARITY
001000                  ODDPAR=1000             ;ODD PARITY
001400                  EVEPAR=1400             ;EVEN PARITY
                                ;TXCSR BIT DEFINITIONS
100000                  DNA=BIT15               ;DATA NOT AVAILABLE
040000                  MTDATA=BIT14            ;MAINT DATA
020000                  CLK=BIT13               ;CLK
002000                  BITW=BIT10              ;BIT WINDOW
000400                  MRESET=BIT8             ;MASTER RESET
000200                  TXDONE=BIT7             ;XMIT DONE
000100                  TXINTE=BIT6             ;XMIT INTR ENABLE
000040                  DNAINTE=BIT5            ;DNA INTR ENAB
000020                  SEND=BIT4               ;SEND
000010                  HDXEN=BIT3              ;HDX/FDX
000001                  BREAK=BIT0              ;BREAK
                                ;TXCSR WRD DEFINITIONS
000000                  USER=0                  ;USER MODE
004000                  MINT=4000               ;MAINT INT MODE
010000                  MEXT=10000              ;MAINT EXT MODE
014000                  SYSTST=14000            ;SYSTEM TEST MODE
                        ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
```

```
                                        ;STANDARD INTERRUPT VECTORS
                                
                        000024          .=24
000024  015006                         .PFAIL          ;POWER FAIL HANDLER
000026  000340                         340             ;SERVICE AT LEVEL 7
000030  014536                         .HLT            ;ERROR HANDLER
000032  000340                         340             ;SERVICE AT LEVEL 7
000034  014504                         .TRPSRV         ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                         340             ;SERVICE AT LEVEL 7

                                ;SOFTWARE SWITCH REGISTER

                        000174          .=174
000174  000000                 DISPREG:.WORD 0         ;SOFTWARE DISPLAY REG.
000176  000000                 SWREG:  .WORD 0         ;SOFTWARE SWITCH REGISTER
000200  000167  001054                 JMP     .START  ;GO TO START OF PROGRAM


                        001100          .=1100

                                        ;INDIRECT POINTERS

001100  177570                 SWR:    177570          ;SWITCH REGISTER POINTER
001102  177570                 LIGHTS:177570           ;DISPLAY REGISTER POINTER
001104  177560                 TKCSR:  177560          ;TELETYPE KEYBOARD CONTROL REGISTER
001106  177562                 TKDBR:  177562          ;TELETYPE KEYBOARD DATA BUFFER
001110  177564                 TPCSR:  177564          ;TELEPRINTER CONTROL REGISTER
001112  177566                 TPDBR:  177566          ;TELEPRINTER DATA BUFFER

                                        ;PROGRAM CONTROL PARAMETERS

001114  000000                 RTRN:   0               ;SCOPE ADDRESS FOR LOOP ON TEST
001116  000000                 NEXT:   0               ;ADDRESS OF NEXT TEST TO BE EXECUTED
001120  000000                 LOCK:   0               ;ADDRESS FOR LOCK ON CURRENT DATA
001122  000000                 ICOUNT: 0               ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
001124  000000                 LPCNT:  0               ;NUMBER OF ITEREATIONS COMPLETED
001126  000000                 TSTNO:  0               ;NUMBER OF TEST IN PROGRESS
001130  000000                 PASCNT: 0               ;NUMBER OF PASSES COMPLETED
001132  000000                 ERRCNT: 0               ;TOTAL NUMBER OF ERRORS
001134  000000                 LSTERR: 0               ;PC OF LAST ERROR CALL

                                        ;PROGRAM VARIABLES

001136  000020                 HOLD:   20              ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
001140  000000                 SHIFT:  0               ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
001142  000000                 COUNT:  0               ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
001144  000000                 TEMP1:  0               ;TEMPORARY STORAGE
001146  000000                 TEMP2:  0               ;TEMPORARY STORAGE
001150  000000                 TEMP3:  0               ;TEMPORARY STORAGE
001152  000000                 TEMP4:  0               ;TEMPORARY STORAGE
001154  000000                 TEMP5:  0               ;TEMPORARY STORAGE
001156  000000                 SAVR0:  0               ;R0 STORAGE
001160  000000                 SAVR1:  0               ;R1 STORAGE
001162  000000                 SAVR2:  0               ;R2 STORAGE
```

```
  752  001164  000000                    SAVR3:  0                       ;R3 STORAGE
  753  001166  000000                    SAVR4:  0                       ;R4 STORAGE
  754  001170  000000                    SAVR5:  0                       ;R5 STORAGE
  755  001172  000000                    SAVSP:  0                       ;STACK POINTER STORAGE
  756  001174  000000                    SAVPC:  0                       ;PROGRAM COUNTER STORAGE
```

```
                                        ;PROGRAM CONVERSATIONAL PARAMETERS
001176    377           SYNCNO: .BYTE   377             ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
001177    377           SEXMIT: .BYTE   377             ;SEC XMIT JUMPER "IN"
001200    377           SEREC:  .BYTE   377             ;SEC REC JUMPER "IN"
001201    377           OPTCLR: .BYTE   377             ;OPTIONAL JUMPER CLR "IN"
001202    000           MULTD:  .BYTE   0               ;NO MULTIPLE DEVICE FLAG
001203    377           JMRBY:  .BYTE   377             ;EXTERNAL MODEM BYPASS JUMPER "IN"
                        .EVEN

                                        ;PROGRAM MULTIPLE DEVICE PARAMETERS
001204  000000          BASEADD:        0               ;PROG CONTROLLED 1ST DEVICE ADDR
001206  000000          KEEPADD:        0               ;SAVED 1ST DEVICE ADDR
001210  000000          LASTADD:        0               ;LAST DEVICE RXCSR ADDR
001212  000000          BASEIV:         0               ;PROG CONTROLLED IV
001214  000000          KEEPIV:         0               ;SAVED INTR VECTOR
001216  000000          ACTREG:         0               ;ACTIVE REGISTER ,,,MODIFY THIS
                                                        ;LOCATION TO DISQUALIFY OR QUALIFY
                                                        ;DEVICES (1= RUN,,,0= DON'T RUN)
001220  000000          ROTADD:         0               ;ROTATING POINTER FOR ACTREG..POINTS
                                                        ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DE

                                        ;PROGRAM CONTROL FLAGS

001222    000           INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
001223    000           STFLG:  .BYTE   0               ;TEST START FLAG
001224    000           ERRFLG: .BYTE   0               ;ERROR OCCURED FLAG
001225    000           LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG

                                        ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
                                        ;POINTERS TO SUBROUTINES CAN BE FOUND
                                        ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS

001226                  .TRPTAB:
                        ;*****************************************************************************
                        ;*****************************************************************************
        104400          SCOPE=TRAP+0    ;CALL TO SCOPE LOOP AND ITERATION HANDLER
001226  013270  .SCOPE
        104401          SCOP1=TRAP+1    ;CALL TO LOOP ON CURRENT DATA HANDLER
001230  013454  .SCOP1
        104402          TYPE=TRAP+2     ;CALL TO TELETYPE OUTPUT ROUTINE
001232  013474  .TYPE
        104403          INSTR=TRAP+3    ;CALL TO ASCII STRING INPUT ROUTINE
001234  013534  .INSTR
        104404          INSTER=TRAP+4   ;CALL TO INPUT ERROR HANDLER
001236  013652  .INSTER
        104405          PARAM=TRAP+5    ;CALL TO NUMERICAL DATA INPUT ROUTINE
001240  013704  .PARAM
        104406          SAVOS=TRAP+6    ;CALL TO REGISTER SAVE ROUTINE
001242  014120  .SAVOS
        104407          RESOS=TRAP+7    ;CALL TO REGISTER RESTORE ROUTINE
001244  014160  .RESOS
        104410          CONVRT=TRAP+10  ;CALL TO DATA OUTPUT ROUTINE
001246  014212  .CONVRT
        104411          CNVRT=TRAP+11   ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF
001250  014216  .CNVRT
        104412          SETFLG=TRAP+12  ;CALL TO FLAG SET ROUTINE
```

```
                                .SETFLG
                                        CKSWR=TRAP+13   ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
                                .CKSWR
                                        CNTLU=TRAP+14   ;CALL TO ALLOW LOADING  OF SWREG FROM TTY
                                .CNTLU
                                ;*****************************************************************************
                                ;*****************************************************************************

                                        ;PROGRAM INITIALIZATION
                                        ;LOCK OUT INTERRUPTS
                                        ;SET UP PROCESSOR STACK
                                        ;SET UP POWER FAIL VECTOR
                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
                                        ;TYPE TITLE MESSAGE

001260  012767  000340  176510  .START: MOV     #340,PS                 ;LOCK OUT INTERRUPTS
001266  012706  001100                  MOV     #STACK,SP               ;SET UP STACK
001272  012737  015006  000024          MOV     #.PFAIL,@#24            ;SET UP POWER FAIL VECTOR
001300  005067  177620                  CLR     LPCNT                   ;CLEAR # OF ITERATION COMPLETED LOCATION
001304  105067  177713                  CLRB    STFLG                   ;CLEAR START FLAG
001310  005067  177614                  CLR     PASCNT                  ;CLEAR PASS COUNT
001314  105067  177704                  CLRB    ERRFLG                  ;CLEAR ERROR FLAG
001320  005067  177606                  CLR     ERRCNT                  ;CLEAR ERROR COUNT
001324  005067  177604                  CLR     LSTERR                  ;CLEAR LAST ERROR POINTER
001330  012767  000001  177570          MOV     #1,TSTNO                ;SET UP FOR TEST 1
001336  012767  001260  177550          MOV     #.START,RTRN            ;SET UP FOR POWER FAIL BEFORE
                                                                        ;TESTING STARTS
001344  105767  177652                  TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
001350  001004                          BNE     ONCE
001352  104402  015326                  TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
001356  105167  177640                  COMB    INIFLG                  ;IF NOT SET FLAG AND DO
001362  012767  177570  177510  ONCE:   MOV     #DSWR,SWR               ;RELOAD HARDWARE SWITCH REGISTER INTO POINTER
001370  012767  177570  177504          MOV     #DLIGHTS,LIGHTS         ;RELOAD HARDWARE DISPLAY REGISTER INTO POINTER
001376  013746  000006                  MOV     @#6,-(SP)               ;SAVE VECTORS
001402  013746  000004                  MOV     @#4,-(SP)
001406  012737  001426  000004          MOV     #64$,@#4                ;SET UP FOR TIMEOUT
001414  022777  177777  177456          CMP     #-1,@SWR                ;REFERENCE HARDWARE SWITCH REGISTER
001422  001402                          BEQ     65$
001424  000407                          BR      66$
001426  022626                  64$:    CMP     (SP)+,(SP)+             ;ADJUST STACK
001430  012767  000176  177442  65$:    MOV     #SWREG,SWR              ;POINT TO SOFTWARE SWITCH REG
001436  012767  000174  177436          MOV     #DISPREG,LIGHTS         ;POINT TO SOFT DISPLAY REG
001444  012637  000004          66$:    MOV     (SP)+,@#4               ;RESTORE VECTORS
001450  012637  000006                  MOV     (SP)+,@#6
001454  005737  000042                  TST     @#42                    ;UNDER MONITOR
001460  001005                          BNE     67$
001462  022767  000176  177410          CMP     #SWREG,SWR              ;IS SWREG USED
001470  001001                          BNE     67$
001472  104414                          CNTLU
001474  032777  000001  177376  67$:    BIT     #SW00,@SWR              ;RESELECT VECTOR & CONTROL REG?
001502  001002                          BNE     1$
001504  000167  000446                  JMP     .BEGIN
001510  012700  000300          1$:     MOV     #300,R0                 ;RESTORE VECTOR AREA TO TRAPCATCHER
001514  012701  000302                  MOV     #302,R1                 ;START AT LOCATION 300
001520  012702  000004                  MOV     #4,R2
001524  010110                  2$:     MOV     R1,(R0)
```

```
 869  001526  005011                          CLR     (R1)
 870  001530  060200                          ADD     R2,R0
 871  001532  060201                          ADD     R2,R1
 872  001534  022701  001000                  CMP     #1000,R1                ;END AT LOCATION 776
 873  001540  002771                          BLT     2$
 874  001542  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
 875  001544  015402                          MREGAD                          ;MESSAGE
 876  001546  104405                          PARAM                           ;CONVERT STRING
 877  001550  160000                          160000                          ;LOW LIMIT
 878  001552  167776                          167776                          ;HIGH LIMIT
 879  001554  017236                          DUBASE                          ;STORE AT THIS LOCATION
 880  001556     001                  .BYTE   1                               ;MASK
 881  001557     001                  .BYTE   1                               ;HOW MANY TIMES + 2
 882  001560  016767  015452  177420          MOV     DUBASE,KEEPADD  ;SAVE
 883  001566  004767  015312                  JSR     PC,DUADDR
 884  001572  016767  177410  177404          MOV     KEEPADD,BASEADD ;RESTORE FOR ROTATION
 885  001600  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
 886  001602  015360                          MVECTO                          ;MESSAGE
 887  001604  104405                          PARAM                           ;CONVERT STRING
 888  001606  000300                          300                             ;LOW LIMIT
 889  001610  000776                          776                             ;HIGH LIMIT
 890  001612  017560                          DURIV                           ;STORE AT THIS LOCATION
 891  001614     001                  .BYTE   1                               ;MASK
 892  001615     004                  .BYTE   4                               ;HOW MANY TIMES + 2
 893  001616  016767  015736  177370          MOV     DURIV,KEEPIV    ;SAVE
 894  001624  016767  015730  177360          MOV     DURIV,BASEIV    ;SET UP FOR ROTATION
 895  001632  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
 896  001634  015463                          MMULT                           ;MESSAGE
 897  001636  104412                          SETFLG                          ;SET FLAG BASED UPON INPUT STRING
 898  001640  001202                          MULTD                           ;THIS FLAG
 899  001642  105767  177334                  TSTB    MULTD   ;ARE THERE MULTIPLE DEVICES
 900                                                          ;ON THE SYSTEM ?
 901  001646  100406                          BMI     BBB     ;YES,ASK NEXT QUESTION
 902  001650  005067  177342                  CLR     ACTREG
 903  001654  005067  177340                  CLR     ROTADD
 904  001660  000167  000140                  JMP     OUTMUL  ;JUMP AROUND NEXT QUESTION
 905  001664                          BBB:
 906  001664  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
 907  001666  015542                          MLASTD                          ;MESSAGE
 908  001670  104405                          PARAM                           ;CONVERT STRING
 909  001672  160000                          160000                          ;LOW LIMIT
 910  001674  167776                          167776                          ;HIGH LIMIT
 911  001676  001210                          LASTADD                         ;STORE AT THIS LOCATION
 912  001700     001                  .BYTE   1                               ;MASK
 913  001701     001                  .BYTE   1                               ;HOW MANY TIMES + 2
 914                                  ;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
 915  001702  012767  000001  177310  1$:     MOV     #1,ROTADD       ;SET UP POINTER
 916  001710  005067  177302                  CLR     ACTREG  ;CLR ACTIVE REGISTER
 917  001714  056767  177300  177274  2$:     BIS     ROTADD,ACTREG   ;MAKE THIS DEVICE ACTIVE
 918  001722  000241                          CLC
 919  001724  006167  177270                  ROL     ROTADD  ;SET UP POINTER
 920  001730  103421                          BCS     3$      ;ARE YOU OUT OF RANGE ?
 921  001732  062767  000010  177244          ADD     #10,BASEADD     ;SET UP BASE ADDRESS
 922  001740  026767  177244  177236          CMP     LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
 923  001746  101362                          BHI     2$      ;NO DO IT AGAIN
 924  001750  056767  177244  177240          BIS     ROTADD,ACTREG   ;THIS ASSUMES THAT THERE ARE AT
```

```
                                        ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
                                        ;MULTIPLE DEVICE QUESTION
001756  012767  000001  177234  4$:   MOV     #1,ROTADD       ;SET UP FOR LATER USE IN END OF PASS ROUTINE
001764  016767  177216  177212        MOV     KEEPADD,BASEADD ;DITTO
001772  000414                        BR      OUTMUL  ;CONTINUE QUESTIONS
001774  016767  177206  177202  3$:   MOV     KEEPADD,BASEADD ;RESTORE
002002  104403                        INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
002004  015725                        MRANGE                          ;MESSAGE
002006  104405                        PARAM                           ;CONVERT STRING
002010  160000                        160000                          ;LOW LIMIT
002012  167776                        167776                          ;HIGH LIMIT
002014  001210                        LASTADD                         ;STORE AT THIS LOCATION
002016     001                .BYTE   1                               ;MASK
002017     001                .BYTE   1                               ;HOW MANY TIMES + 2
002020  000167  177656                JMP     1$      ;DO IT AGAIN
002024                        OUTMUL:
002024  104403                        INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
002026  016211                        MLEVEL                          ;MESSAGE
002030  104405                        PARAM                           ;CONVERT STRING
002032  000004                        4                               ;LOW LIMIT
002034  000007                        7                               ;HIGH LIMIT
002036  017100                        DUPRT                           ;STORE AT THIS LOCATION
002040     000                .BYTE   0                               ;MASK
002041     001                .BYTE   1                               ;HOW MANY TIMES + 2
002042  004767  014762                JSR     PC,DULEV
                                        ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
                                        ;BUFFER TO THE CHARACTERS "1" AND "2"
                                        ;IF THE CHARACTER IS "1" CLEAR THE FLAG
                                        ;IF THE CHARACTER IS "2" SET THE FLAG
002046                          AAA:
002046  104403                        INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
002050  016236                        MSYNC                           ;MESSAGE
002052  122767  000061  014610  3$:   CMPB    #'1,INBUF               ;IS IT "1" ?
002060  001003                        BNE     1$
002062  105067  177110                CLRB    SYNCNO                  ;000
002066  000412                        BR      4$
002070  122767  000062  014572  1$:   CMPB    #'2,INBUF               ;IS IT "2" ?
002076  001004                        BNE     2$
002100  112767  177777  177070        MOVB    #-1,SYNCNO                      ;377
002106  000402                        BR      4$
002110  104404                2$:     INSTER                          ;RETRY
002112  000757                        BR      3$
002114  000240                4$:     NOP
002116  104403                        INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
002120  016304                        MWIRE6                          ;MESSAGE
002122  104412                        SETFLG                          ;SET FLAG BASED UPON INPUT STRING
002124  001177                        SEXMIT                          ;THIS FLAG
002126  104403                        INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
002130  016352                        MWIRE5                          ;MESSAGE
002132  104412                        SETFLG                          ;SET FLAG BASED UPON INPUT STRING
002134  001200                        SEREC                           ;THIS FLAG
002136  104403                        INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
002140  016417                        MWIRE4                          ;MESSAGE
002142  104412                        SETFLG                          ;SET FLAG BASED UPON INPUT STRING
002144  001201                        OPTCLR                          ;THIS FLAG
002146  104403                        INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
```

```
  981  002150  016473                          MEXTJ                           ;MESSAGE
  982  002152  104412                          SETFLG                          ;SET FLAG BASED UPON INPUT STRING
  983  002154  001203                          JMRBY                           ;THIS FLAG
  984
  985                                          ;TEST START AND RESTART
  986
  987  002156  012767  000340  175612  .BEGIN: MOV     #340,PS                 ;LOCK OUT INTERRUPTS
  988  002164  012706  001100                  MOV     #STACK,SP               ;SET UP STACK
  989  002170  005737  000042                  TST     @#42                    ;IS PROGRAM UNDER MONITOR CONTROL
  990  002174  001056                          BNE     3$
  991  002176  105767  177000                  TSTB    MULTD   ;DON'T ALLOW LOCK ON TEST IF RUNNING
  992                                                  ;MULTIPLE DEVICES
  993  002202  001407                          BEQ     5$                      ;IF NO,TEST FOR LOCK ON TEST
  994  002204  016767  011240  011140          MOV     BRW,TTST                ;RESTORE NORMAL SCOPE LOOP
  995  002212  016767  011234  011134          MOV     BRX,TTST+2              ;DITTO
  996  002220  000444                          BR      3$                      ;JUMP AROUND IF YES
  997  002222  032777  000004  176650  5$:     BIT     #BIT2,@SWR              ;CHECK FOR LOCK ON TEST
  998  002230  001416                          BEQ     1$
  999  002232  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
 1000  002234  016146                          MLOCK                           ;MESSAGE
 1001  002236  104412                          SETFLG                          ;SET FLAG BASED UPON INPUT STRING
 1002  002240  001225                          LOKFLG                          ;THIS FLAG
 1003  002242  105767  176757                  TSTB    LOKFLG                  ;IS LOCK ON TEST OPTION SELECTED
 1004  002246  001407                          BEQ     1$
 1005  002250  012767  000240  011074          MOV     #NOP,TTST
 1006  002256  012767  000240  011070          MOV     #NOP,TTST+2             ;SET UP TO LOCK
 1007  002264  000406                          BR      2$
 1008  002266  016767  011156  011056  1$:     MOV     BRW,TTST
 1009  002274  016767  011152  011052          MOV     BRX,TTST+2              ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
 1010  002302  032777  000002  176570  2$:     BIT     #SW01,@SWR              ;IF SW01=1, GET STARTING PC
 1011  002310  001410                          BEQ     3$
 1012  002312  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
 1013  002314  016133                          MTSTPC                          ;MESSAGE
 1014  002316  104405                          PARAM                           ;CONVERT STRING
 1015  002320  002350                          TST1                            ;LOW LIMIT
 1016  002322  012512                          TLAST                           ;HIGH LIMIT
 1017  002324  001114                          RTRN                            ;STORE AT THIS LOCATION
 1018  002326     001                  .BYTE   1                               ;MASK
 1019  002327     001                  .BYTE   1                               ;HOW MANY TIMES + 2
 1020  002330  000403                          BR      4$
 1021  002332  012767  002350  176554  3$:     MOV     #TST1,RTRN              ;START AT TEST 1
 1022  002340  104402  016127          4$:     TYPE    ,MR                     ;TYPE R
 1023  002344  000177  176544                  JMP     @RTRN                   ;START TESTING
 1024
 1025                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 1026                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
 1027                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 1028                                          ;;(OVRRUN,RXERR)
 1029                                          ;;MODE:ISYMOD
 1030                                          ;;LENGTH:SIX
 1031                                          ;;CHAR:25
 1032                                          ;;
 1033  002350  012767  000001  176550  TST1:   MOV     #1,TSTNO        ;SAVE THIS
 1034  002356  012767  002612  176532          MOV     #TST2,NEXT              ;GO TO THIS TEST WHEN THRU
 1035  002364  052777  000400  015156          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1036  002372  012777  000000  015144          MOV     #ISYMOD,@PARCSR ;SET THE MODE
```

```
1037  002400  052777  000400  015142          BIS     #MRESET,@TXCSR  ;MASTER RESET
1038
1039                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1040  002406  012777  064001  015134          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1041
1042                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1043  002414  012777  002000  015122          MOV     #ISYMOD!SIX!NOPAR!0,@PARCSR
1044  002422  052777  000020  015104          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1045                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1046  002430  042777  020000  015112          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1047  002436  052777  020000  015104          BIS     #CLK,@TXCSR     ;POKE CLK UP
1048                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1049  002444  042777  020000  015076          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1050  002452  052777  020000  015070          BIS     #CLK,@TXCSR     ;POKE CLK UP
1051  002460  016703  015054                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1052  002464  012700  000025                  MOV     #25,R0  ;EXPECTED
1053  002470  012767  000010  176442          MOV     #8.,SHIFT       ;# OF SHIFTS
1054  002476  012767  000252  176440          MOV     #252,TEMP1      ;DATA CHAR
1055  002504  004767  014530                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1056  002510  105777  015020                  TSTB    @RXCSR  ;RXDONE ?
1057  002514  100401                          BMI     64$
1058  002516  104000                          HLT     ;RXDONE SHOULD BE SET
1059  002520                          64$:
1060  002520  017701  015014                  MOV     @RXDBUF,R1      ;ACTUAL
1061  002524  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1062  002526  001401                          BEQ     65$
1063  002530  104002                          HLT     2               ;RECEIVED DATA DID NOT MATCH
1064                                                                  ;EXPECTED DATA - CHECK MAINT DATA
1065                                                                  ;OR RECEIVER LOGIC
1066  002532                          65$:
1067  002532  012767  000010  176400          MOV     #8.,SHIFT       ;# OF SHIFTS
1068  002540  012767  000252  176376          MOV     #252,TEMP1      ;DATA CHAR
1069  002546  004767  014466                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1070                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1071  002552  012767  000010  176360          MOV     #8.,SHIFT       ;# OF SHIFTS
1072  002560  012767  000252  176356          MOV     #252,TEMP1      ;DATA CHAR
1073  002566  004767  014446                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1074  002572  012700  140025                  MOV     #140000!25,R0   ;EXPECTED DATA PLUS
1075                                                                  ;RXERR & OVRRUN
1076  002576  017701  014736                  MOV     @RXDBUF,R1      ;ACTUAL
1077  002602  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
1078  002604  001401                          BEQ     66$
1079  002606  104002                          HLT     2               ;SPECIFICALLY LOOK AT RXERR &
1080                                                                  ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1081  002610                          66$:
1082  002610  104400                          SCOPE
1083                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1084                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1085                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1086                                          ;;(OVRRUN,RXERR)
1087                                          ;;MODE:ISYMOD
1088                                          ;;LENGTH:SIX
1089                                          ;;CHAR:52
1090                                          ;;
1091  002612  012767  000002  176306  TST2:   MOV     #2,TSTNO        ;SAVE THIS
1092  002620  012767  003054  176270          MOV     #TST3,NEXT              ;GO TO THIS TEST WHEN THRU
```

```
 1093  002626  052777  000400  014714          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1094  002634  012777  000000  014702          MOV     #ISYMOD,@PARCSR ;SET THE MODE
 1095  002642  052777  000400  014700          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1096
 1097                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1098  002650  012777  064001  014672          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1099
 1100                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1101  002656  012777  002000  014660          MOV     #ISYMOD!SIX!NOPAR!0,@PARCSR
 1102  002664  052777  000020  014642          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 1103                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 1104  002672  042777  020000  014650          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1105  002700  052777  020000  014642          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1106                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1107  002706  042777  020000  014634          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1108  002714  052777  020000  014626          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1109  002722  016703  014612                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1110  002726  012700  000052                  MOV     #52,R0  ;EXPECTED
 1111  002732  012767  000010  176200          MOV     #8.,SHIFT       ;# OF SHIFTS
 1112  002740  012767  000324  176176          MOV     #324,TEMP1      ;DATA CHAR
 1113  002746  004767  014266                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1114  002752  105777  014556                  TSTB    @RXCSR  ;RXDONE ?
 1115  002756  100401                          BMI     64$
 1116  002760  104000                          HLT     ;RXDONE SHOULD BE SET
 1117  002762                          64$:
 1118  002762  017701  014552                  MOV     @RXDBUF,R1      ;ACTUAL
 1119  002766  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
 1120  002770  001401                          BEQ     65$
 1121  002772  104002                          HLT     2               ;RECEIVED DATA DID NOT MATCH
 1122                                                                  ;EXPECTED DATA - CHECK MAINT DATA
 1123                                                                  ;OR RECEIVER LOGIC
 1124  002774                          65$:
 1125  002774  012767  000010  176136          MOV     #8.,SHIFT       ;# OF SHIFTS
 1126  003002  012767  000324  176134          MOV     #324,TEMP1      ;DATA CHAR
 1127  003010  004767  014224                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1128                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
 1129  003014  012767  000010  176116          MOV     #8.,SHIFT       ;# OF SHIFTS
 1130  003022  012767  000324  176114          MOV     #324,TEMP1      ;DATA CHAR
 1131  003030  004767  014204                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1132  003034  012700  140052                  MOV     #140000!52,R0   ;EXPECTED DATA PLUS
 1133                                                                  ;RXERR & OVRRUN
 1134  003040  017701  014474                  MOV     @RXDBUF,R1      ;ACTUAL
 1135  003044  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
 1136  003046  001401                          BEQ     66$
 1137  003050  104002                          HLT     2               ;SPECIFICALLY LOOK AT RXERR &
 1138                                                                  ;OVRRUN BITS...THEY BOTH SHOULD BE SET
 1139  003052                          66$:
 1140  003052  104400                          SCOPE
 1141                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 1142                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
 1143                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 1144                                          ;;(OVRRUN,RXERR)
 1145                                          ;;MODE:ISYMOD
 1146                                          ;;LENGTH:SIX
 1147                                          ;;CHAR:77
 1148                                          ;;
```

```
1149  003054  012767  000003  176044  TST3:   MOV     #3,TSTNO        ;SAVE THIS
1150  003062  012767  003316  176026          MOV     #TST4,NEXT              ;GO TO THIS TEST WHEN THRU
1151  003070  052777  000400  014452          BIS     #MRESET,@TXCSR  ;MASTER RESET
1152  003076  012777  000000  014440          MOV     #ISYMOD,@PARCSR ;SET THE MODE
1153  003104  052777  000400  014436          BIS     #MRESET,@TXCSR  ;MASTER RESET
1154
1155                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1156  003112  012777  064001  014430          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1157
1158                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1159  003120  012777  002000  014416          MOV     #ISYMOD!SIX!NOPAR!0,@PARCSR
1160  003126  052777  000020  014400          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1161                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1162  003134  042777  020000  014406          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1163  003142  052777  020000  014400          BIS     #CLK,@TXCSR     ;POKE CLK UP
1164                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1165  003150  042777  020000  014372          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1166  003156  052777  020000  014364          BIS     #CLK,@TXCSR     ;POKE CLK UP
1167  003164  016703  014350                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1168  003170  012700  000077                  MOV     #77,R0  ;EXPECTED
1169  003174  012767  000010  175736          MOV     #8.,SHIFT       ;# OF SHIFTS
1170  003202  012767  000376  175734          MOV     #376,TEMP1      ;DATA CHAR
1171  003210  004767  014024                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1172  003214  105777  014314                  TSTB    @RXCSR  ;RXDONE ?
1173  003220  100401                          BMI     64$
1174  003222  104000                          HLT     ;RXDONE SHOULD BE SET
1175  003224                          64$:
1176  003224  017701  014310                  MOV     @RXDBUF,R1      ;ACTUAL
1177  003230  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1178  003232  001401                          BEQ     65$
1179  003234  104002                          HLT     2               ;RECEIVED DATA DID NOT MATCH
1180                                                                  ;EXPECTED DATA - CHECK MAINT DATA
1181                                                                  ;OR RECEIVER LOGIC
1182  003236                          65$:
1183  003236  012767  000010  175674          MOV     #8.,SHIFT       ;# OF SHIFTS
1184  003244  012767  000376  175672          MOV     #376,TEMP1      ;DATA CHAR
1185  003252  004767  013762                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1186                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1187  003256  012767  000010  175654          MOV     #8.,SHIFT       ;# OF SHIFTS
1188  003264  012767  000376  175652          MOV     #376,TEMP1      ;DATA CHAR
1189  003272  004767  013742                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1190  003276  012700  140077                  MOV     #140000!77,R0   ;EXPECTED DATA PLUS
1191                                                                  ;RXERR & OVRRUN
1192  003302  017701  014232                  MOV     @RXDBUF,R1      ;ACTUAL
1193  003306  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
1194  003310  001401                          BEQ     66$
1195  003312  104002                          HLT     2               ;SPECIFICALLY LOOK AT RXERR &
1196                                                                  ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1197  003314                          66$:
1198  003314  104400                          SCOPE
1199                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1200                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1201                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1202                                          ;;(OVRRUN,RXERR)
1203                                          ;;MODE:ISYMOD
1204                                          ;;LENGTH:SIX
```

```
1205                                            ;;CHAR:0
1206                                            ;;
1207  003316  012767  000004  175602  TST4:     MOV     #4,TSTNO        ;SAVE THIS
1208  003324  012767  003560  175564            MOV     #TST5,NEXT              ;GO TO THIS TEST WHEN THRU
1209  003332  052777  000400  014210            BIS     #MRESET,@TXCSR  ;MASTER RESET
1210  003340  012777  000000  014176            MOV     #ISYMOD,@PARCSR ;SET THE MODE
1211  003346  052777  000400  014174            BIS     #MRESET,@TXCSR  ;MASTER RESET
1212
1213                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1214  003354  012777  064001  014166            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1215
1216                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1217  003362  012777  002000  014154            MOV     #ISYMOD!SIX!NOPAR!0,@PARCSR
1218  003370  052777  000020  014136            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1219                                            ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1220  003376  042777  020000  014144            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1221  003404  052777  020000  014136            BIS     #CLK,@TXCSR     ;POKE CLK UP
1222                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1223  003412  042777  020000  014130            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1224  003420  052777  020000  014122            BIS     #CLK,@TXCSR     ;POKE CLK UP
1225  003426  016703  014106                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1226  003432  012700  000000                    MOV     #0,R0   ;EXPECTED
1227  003436  012767  000010  175474            MOV     #8.,SHIFT       ;# OF SHIFTS
1228  003444  012767  000200  175472            MOV     #200,TEMP1      ;DATA CHAR
1229  003452  004767  013562                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1230  003456  105777  014052                    TSTB    @RXCSR  ;RXDONE ?
1231  003462  100401                            BMI     64$
1232  003464  104000                            HLT     ;RXDONE SHOULD BE SET
1233  003466                            64$:
1234  003466  017701  014046                    MOV     @RXDBUF,R1      ;ACTUAL
1235  003472  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1236  003474  001401                            BEQ     65$
1237  003476  104002                            HLT     2               ;RECEIVED DATA DID NOT MATCH
1238                                                                    ;EXPECTED DATA - CHECK MAINT DATA
1239                                                                    ;OR RECEIVER LOGIC
1240  003500                            65$:
1241  003500  012767  000010  175432            MOV     #8.,SHIFT       ;# OF SHIFTS
1242  003506  012767  000200  175430            MOV     #200,TEMP1      ;DATA CHAR
1243  003514  004767  013520                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1244                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1245  003520  012767  000010  175412            MOV     #8.,SHIFT       ;# OF SHIFTS
1246  003526  012767  000200  175410            MOV     #200,TEMP1      ;DATA CHAR
1247  003534  004767  013500                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1248  003540  012700  140000                    MOV     #140000!0,R0    ;EXPECTED DATA PLUS
1249                                                                    ;RXERR & OVRRUN
1250  003544  017701  013770                    MOV     @RXDBUF,R1      ;ACTUAL
1251  003550  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
1252  003552  001401                            BEQ     66$
1253  003554  104002                            HLT     2               ;SPECIFICALLY LOOK AT RXERR &
1254                                                                    ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1255  003556                            66$:
1256  003556  104400                            SCOPE
1257                                            ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1258                                            ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1259                                            ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1260                                            ;;(OVRRUN,RXERR)
```

```
                                        ;;MODE:ISYMOD
                                        ;;LENGTH:SEVEN
                                        ;;CHAR:125
                                        ;;
003560  012767  000005  175340  TST5:   MOV     #5,TSTNO        ;SAVE THIS
003566  012767  004022  175322          MOV     #TST6,NEXT              ;GO TO THIS TEST WHEN THRU
003574  052777  000400  013746          BIS     #MRESET,@TXCSR  ;MASTER RESET
003602  012777  000000  013734          MOV     #ISYMOD,@PARCSR ;SET THE MODE
003610  052777  000400  013732          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
003616  012777  064001  013724          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
003624  012777  004000  013712          MOV     #ISYMOD!SEVEN!NOPAR!0,@PARCSR
003632  052777  000020  013674          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                        ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
003640  042777  020000  013702          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003646  052777  020000  013674          BIS     #CLK,@TXCSR     ;POKE CLK UP
                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
003654  042777  020000  013666          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003662  052777  020000  013660          BIS     #CLK,@TXCSR     ;POKE CLK UP
003670  016703  013644                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
003674  012700  000125                  MOV     #125,R0 ;EXPECTED
003700  012767  000011  175232          MOV     #9.,SHIFT       ;# OF SHIFTS
003706  012767  000652  175230          MOV     #652,TEMP1      ;DATA CHAR
003714  004767  013320                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
003720  105777  013610                  TSTB    @RXCSR  ;RXDONE ?
003724  100401                          BMI     64$
003726  104000                          HLT     ;RXDONE SHOULD BE SET
003730                          64$:
003730  017701  013604                  MOV     @RXDBUF,R1      ;ACTUAL
003734  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
003736  001401                          BEQ     65$
003740  104002                          HLT     2               ;RECEIVED DATA DID NOT MATCH
                                                                ;EXPECTED DATA - CHECK MAINT DATA
                                                                ;OR RECEIVER LOGIC
003742                          65$:
003742  012767  000011  175170          MOV     #9.,SHIFT       ;# OF SHIFTS
003750  012767  000652  175166          MOV     #652,TEMP1      ;DATA CHAR
003756  004767  013256                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
003762  012767  000011  175150          MOV     #9.,SHIFT       ;# OF SHIFTS
003770  012767  000652  175146          MOV     #652,TEMP1      ;DATA CHAR
003776  004767  013236                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
004002  012700  140125                  MOV     #140000!125,R0  ;EXPECTED DATA PLUS
                                                                ;RXERR & OVRRUN
004006  017701  013526                  MOV     @RXDBUF,R1      ;ACTUAL
004012  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
004014  001401                          BEQ     66$
004016  104002                          HLT     2               ;SPECIFICALLY LOOK AT RXERR &
                                                                ;OVRRUN BITS...THEY BOTH SHOULD BE SET
004020                          66$:
004020  104400                          SCOPE
                                        ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
                                        ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
```

```
1317                                            ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1318                                            ;;(OVRRUN,RXERR)
1319                                            ;;MODE:ISYMOD
1320                                            ;;LENGTH:SEVEN
1321                                            ;;CHAR:52
1322                                            ;;
1323  004022  012767  000006  175076  TST6:     MOV     #6,TSTNO        ;SAVE THIS
1324  004030  012767  004264  175060            MOV     #TST7,NEXT              ;GO TO THIS TEST WHEN THRU
1325  004036  052777  000400  013504            BIS     #MRESET,@TXCSR  ;MASTER RESET
1326  004044  012777  000000  013472            MOV     #ISYMOD,@PARCSR ;SET THE MODE
1327  004052  052777  000400  013470            BIS     #MRESET,@TXCSR  ;MASTER RESET
1328
1329                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1330  004060  012777  064001  013462            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1331
1332                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1333  004066  012777  004000  013450            MOV     #ISYMOD!SEVEN!NOPAR!0,@PARCSR
1334  004074  052777  000020  013432            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1335                                            ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1336  004102  042777  020000  013440            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1337  004110  052777  020000  013432            BIS     #CLK,@TXCSR     ;POKE CLK UP
1338                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1339  004116  042777  020000  013424            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1340  004124  052777  020000  013416            BIS     #CLK,@TXCSR     ;POKE CLK UP
1341  004132  016703  013402                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1342  004136  012700  000052                    MOV     #52,R0  ;EXPECTED
1343  004142  012767  000011  174770            MOV     #9.,SHIFT       ;# OF SHIFTS
1344  004150  012767  000524  174766            MOV     #524,TEMP1      ;DATA CHAR
1345  004156  004767  013056                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1346  004162  105777  013346                    TSTB    @RXCSR  ;RXDONE ?
1347  004166  100401                            BMI     64$
1348  004170  104000                            HLT     ;RXDONE SHOULD BE SET
1349  004172                            64$:
1350  004172  017701  013342                    MOV     @RXDBUF,R1      ;ACTUAL
1351  004176  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1352  004200  001401                            BEQ     65$
1353  004202  104002                            HLT     2               ;RECEIVED DATA DID NOT MATCH
1354                                                                    ;EXPECTED DATA - CHECK MAINT DATA
1355                                                                    ;OR RECEIVER LOGIC
1356  004204                            65$:
1357  004204  012767  000011  174726            MOV     #9.,SHIFT       ;# OF SHIFTS
1358  004212  012767  000524  174724            MOV     #524,TEMP1      ;DATA CHAR
1359  004220  004767  013014                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1360                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1361  004224  012767  000011  174706            MOV     #9.,SHIFT       ;# OF SHIFTS
1362  004232  012767  000524  174704            MOV     #524,TEMP1      ;DATA CHAR
1363  004240  004767  012774                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1364  004244  012700  140052                    MOV     #140000!52,R0   ;EXPECTED DATA PLUS
1365                                                                    ;RXERR & OVRRUN
1366  004250  017701  013264                    MOV     @RXDBUF,R1      ;ACTUAL
1367  004254  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
1368  004256  001401                            BEQ     66$
1369  004260  104002                            HLT     2               ;SPECIFICALLY LOOK AT RXERR &
1370                                                                    ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1371  004262                            66$:
1372  004262  104400                            SCOPE
```

```
 1373                                           ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 1374                                           ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
 1375                                           ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 1376                                           ;;(OVRRUN,RXERR)
 1377                                           ;;MODE:ISYMOD
 1378                                           ;;LENGTH:SEVEN
 1379                                           ;;CHAR:177
 1380                                           ;;
 1381  004264  012767  000007  174634  TST7:    MOV     #7,TSTNO        ;SAVE THIS
 1382  004272  012767  004526  174616           MOV     #TST8,NEXT              ;GO TO THIS TEST WHEN THRU
 1383  004300  052777  000400  013242           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1384  004306  012777  000000  013230           MOV     #ISYMOD,@PARCSR ;SET THE MODE
 1385  004314  052777  000400  013226           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1386
 1387                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1388  004322  012777  064001  013220           MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1389
 1390                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1391  004330  012777  004000  013206           MOV     #ISYMOD!SEVEN!NOPAR!0,@PARCSR
 1392  004336  052777  000020  013170           BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 1393                                           ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 1394  004344  042777  020000  013176           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1395  004352  052777  020000  013170           BIS     #CLK,@TXCSR     ;POKE CLK UP
 1396                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1397  004360  042777  020000  013162           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1398  004366  052777  020000  013154           BIS     #CLK,@TXCSR     ;POKE CLK UP
 1399  004374  016703  013140                   MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1400  004400  012700  000177                   MOV     #177,R0 ;EXPECTED
 1401  004404  012767  000011  174526           MOV     #9.,SHIFT       ;# OF SHIFTS
 1402  004412  012767  000776  174524           MOV     #776,TEMP1      ;DATA CHAR
 1403  004420  004767  012614                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1404  004424  105777  013104                   TSTB    @RXCSR  ;RXDONE ?
 1405  004430  100401                           BMI     64$
 1406  004432  104000                           HLT     ;RXDONE SHOULD BE SET
 1407  004434                          64$:
 1408  004434  017701  013100                   MOV     @RXDBUF,R1      ;ACTUAL
 1409  004440  020001                           CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
 1410  004442  001401                           BEQ     65$
 1411  004444  104002                           HLT     2               ;RECEIVED DATA DID NOT MATCH
 1412                                                                   ;EXPECTED DATA - CHECK MAINT DATA
 1413                                                                   ;OR RECEIVER LOGIC
 1414  004446                          65$:
 1415  004446  012767  000011  174464           MOV     #9.,SHIFT       ;# OF SHIFTS
 1416  004454  012767  000776  174462           MOV     #776,TEMP1      ;DATA CHAR
 1417  004462  004767  012552                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1418                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
 1419  004466  012767  000011  174444           MOV     #9.,SHIFT       ;# OF SHIFTS
 1420  004474  012767  000776  174442           MOV     #776,TEMP1      ;DATA CHAR
 1421  004502  004767  012532                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1422  004506  012700  140177                   MOV     #140000!177,R0  ;EXPECTED DATA PLUS
 1423                                                                   ;RXERR & OVRRUN
 1424  004512  017701  013022                   MOV     @RXDBUF,R1      ;ACTUAL
 1425  004516  020001                           CMP     R0,R1   ;COMPARE EXP VS. ACT
 1426  004520  001401                           BEQ     66$
 1427  004522  104002                           HLT     2               ;SPECIFICALLY LOOK AT RXERR &
 1428                                                                   ;OVRRUN BITS...THEY BOTH SHOULD BE SET
```

```
 1429  004524                          66$:
 1430  004524  104400                          SCOPE
 1431                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 1432                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
 1433                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 1434                                          ;;(OVRRUN,RXERR)
 1435                                          ;;MODE:ISYMOD
 1436                                          ;;LENGTH:SEVEN
 1437                                          ;;CHAR:0
 1438                                          ;;
 1439  004526  012767  000010  174372  TST8:    MOV     #8,TSTNO        ;SAVE THIS
 1440  004534  012767  004770  174354          MOV     #TST9,NEXT              ;GO TO THIS TEST WHEN THRU
 1441  004542  052777  000400  013000          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1442  004550  012777  000000  012766          MOV     #ISYMOD,@PARCSR ;SET THE MODE
 1443  004556  052777  000400  012764          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1444
 1445                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1446  004564  012777  064001  012756          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1447
 1448                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1449  004572  012777  004000  012744          MOV     #ISYMOD!SEVEN!NOPAR!0,@PARCSR
 1450  004600  052777  000020  012726          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 1451                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 1452  004606  042777  020000  012734          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1453  004614  052777  020000  012726          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1454                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1455  004622  042777  020000  012720          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1456  004630  052777  020000  012712          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1457  004636  016703  012676                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1458  004642  012700  000000                  MOV     #0,R0   ;EXPECTED
 1459  004646  012767  000011  174264          MOV     #9.,SHIFT       ;# OF SHIFTS
 1460  004654  012767  000400  174262          MOV     #400,TEMP1      ;DATA CHAR
 1461  004662  004767  012352                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1462  004666  105777  012642                  TSTB    @RXCSR  ;RXDONE ?
 1463  004672  100401                          BMI     64$
 1464  004674  104000                          HLT     ;RXDONE SHOULD BE SET
 1465  004676                          64$:
 1466  004676  017701  012636                  MOV     @RXDBUF,R1      ;ACTUAL
 1467  004702  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
 1468  004704  001401                          BEQ     65$
 1469  004706  104002                          HLT     2               ;RECEIVED DATA DID NOT MATCH
 1470                                                                  ;EXPECTED DATA - CHECK MAINT DATA
 1471                                                                  ;OR RECEIVER LOGIC
 1472  004710                          65$:
 1473  004710  012767  000011  174222          MOV     #9.,SHIFT       ;# OF SHIFTS
 1474  004716  012767  000400  174220          MOV     #400,TEMP1      ;DATA CHAR
 1475  004724  004767  012310                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1476                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
 1477  004730  012767  000011  174202          MOV     #9.,SHIFT       ;# OF SHIFTS
 1478  004736  012767  000400  174200          MOV     #400,TEMP1      ;DATA CHAR
 1479  004744  004767  012270                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1480  004750  012700  140000                  MOV     #140000!0,R0    ;EXPECTED DATA PLUS
 1481                                                                  ;RXERR & OVRRUN
 1482  004754  017701  012560                  MOV     @RXDBUF,R1      ;ACTUAL
 1483  004760  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
 1484  004762  001401                          BEQ     66$
```

```
1485  004764  104002                          HLT     2       ;SPECIFICALLY LOOK AT RXERR &
1486                                                          ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1487  004766                          66$:
1488  004766  104400                          SCOPE
1489                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1490                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1491                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1492                                          ;;(OVRRUN,RXERR)
1493                                          ;;MODE:ISYMOD
1494                                          ;;LENGTH:EIGHT
1495                                          ;;CHAR:125
1496                                          ;;
1497  004770  012767  000011  174130  TST9:   MOV     #9,TSTNO        ;SAVE THIS
1498  004776  012767  005232  174112          MOV     #TST10,NEXT             ;GO TO THIS TEST WHEN THRU
1499  005004  052777  000400  012536          BIS     #MRESET,@TXCSR  ;MASTER RESET
1500  005012  012777  000000  012524          MOV     #ISYMOD,@PARCSR ;SET THE MODE
1501  005020  052777  000400  012522          BIS     #MRESET,@TXCSR  ;MASTER RESET
1502
1503                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1504  005026  012777  064001  012514          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1505
1506                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1507  005034  012777  006000  012502          MOV     #ISYMOD!EIGHT!NOPAR!0,@PARCSR
1508  005042  052777  000020  012464          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1509                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1510  005050  042777  020000  012472          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1511  005056  052777  020000  012464          BIS     #CLK,@TXCSR     ;POKE CLK UP
1512                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1513  005064  042777  020000  012456          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1514  005072  052777  020000  012450          BIS     #CLK,@TXCSR     ;POKE CLK UP
1515  005100  016703  012434                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1516  005104  012700  000125                  MOV     #125,R0 ;EXPECTED
1517  005110  012767  000012  174022          MOV     #10.,SHIFT      ;# OF SHIFTS
1518  005116  012767  001252  174020          MOV     #1252,TEMP1     ;DATA CHAR
1519  005124  004767  012110                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1520  005130  105777  012400                  TSTB    @RXCSR  ;RXDONE ?
1521  005134  100401                          BMI     64$
1522  005136  104000                          HLT     ;RXDONE SHOULD BE SET
1523  005140                          64$:
1524  005140  017701  012374                  MOV     @RXDBUF,R1      ;ACTUAL
1525  005144  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1526  005146  001401                          BEQ     65$
1527  005150  104002                          HLT     2       ;RECEIVED DATA DID NOT MATCH
1528                                                          ;EXPECTED DATA - CHECK MAINT DATA
1529                                                          ;OR RECEIVER LOGIC
1530  005152                          65$:
1531  005152  012767  000012  173760          MOV     #10.,SHIFT      ;# OF SHIFTS
1532  005160  012767  001252  173756          MOV     #1252,TEMP1     ;DATA CHAR
1533  005166  004767  012046                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1534                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1535  005172  012767  000012  173740          MOV     #10.,SHIFT      ;# OF SHIFTS
1536  005200  012767  001252  173736          MOV     #1252,TEMP1     ;DATA CHAR
1537  005206  004767  012026                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1538  005212  012700  140125                  MOV     #140000!125,R0  ;EXPECTED DATA PLUS
1539                                                          ;RXERR & OVRRUN
1540  005216  017701  012316                  MOV     @RXDBUF,R1      ;ACTUAL
```

```
 1541  005222  020001                        CMP     R0,R1   ;COMPARE EXP VS. ACT
 1542  005224  001401                        BEQ     66$
 1543  005226  104002                        HLT     2       ;SPECIFICALLY LOOK AT RXERR &
 1544                                                        ;OVRRUN BITS...THEY BOTH SHOULD BE SET
 1545  005230                          66$:
 1546  005230  104400                        SCOPE
 1547                                        ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 1548                                        ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
 1549                                        ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 1550                                        ;;(OVRRUN,RXERR)
 1551                                        ;;MODE:ISYMOD
 1552                                        ;;LENGTH:EIGHT
 1553                                        ;;CHAR:252
 1554                                        ;;
 1555  005232  012767  000012  173666  TST10: MOV     #10,TSTNO       ;SAVE THIS
 1556  005240  012767  005474  173650        MOV     #TST11,NEXT             ;GO TO THIS TEST WHEN THRU
 1557  005246  052777  000400  012274        BIS     #MRESET,@TXCSR  ;MASTER RESET
 1558  005254  012777  000000  012262        MOV     #ISYMOD,@PARCSR ;SET THE MODE
 1559  005262  052777  000400  012260        BIS     #MRESET,@TXCSR  ;MASTER RESET
 1560
 1561                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1562  005270  012777  064001  012252        MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1563
 1564                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1565  005276  012777  006000  012240        MOV     #ISYMOD!EIGHT!NOPAR!0,@PARCSR
 1566  005304  052777  000020  012222        BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 1567                                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 1568  005312  042777  020000  012230        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1569  005320  052777  020000  012222        BIS     #CLK,@TXCSR     ;POKE CLK UP
 1570                          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1571  005326  042777  020000  012214        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1572  005334  052777  020000  012206        BIS     #CLK,@TXCSR     ;POKE CLK UP
 1573  005342  016703  012172                MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1574  005346  012700  000252                MOV     #252,R0 ;EXPECTED
 1575  005352  012767  000012  173560        MOV     #10.,SHIFT      ;# OF SHIFTS
 1576  005360  012767  001524  173556        MOV     #1524,TEMP1     ;DATA CHAR
 1577  005366  004767  011646                JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1578  005372  105777  012136                TSTB    @RXCSR  ;RXDONE ?
 1579  005376  100401                        BMI     64$
 1580  005400  104000                        HLT     ;RXDONE SHOULD BE SET
 1581  005402                          64$:
 1582  005402  017701  012132                MOV     @RXDBUF,R1      ;ACTUAL
 1583  005406  020001                        CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
 1584  005410  001401                        BEQ     65$
 1585  005412  104002                        HLT     2       ;RECEIVED DATA DID NOT MATCH
 1586                                                        ;EXPECTED DATA - CHECK MAINT DATA
 1587                                                        ;OR RECEIVER LOGIC
 1588  005414                          65$:
 1589  005414  012767  000012  173516        MOV     #10.,SHIFT      ;# OF SHIFTS
 1590  005422  012767  001524  173514        MOV     #1524,TEMP1     ;DATA CHAR
 1591  005430  004767  011604                JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1592                          ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
 1593  005434  012767  000012  173476        MOV     #10.,SHIFT      ;# OF SHIFTS
 1594  005442  012767  001524  173474        MOV     #1524,TEMP1     ;DATA CHAR
 1595  005450  004767  011564                JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1596  005454  012700  140252                MOV     #140000!252,R0  ;EXPECTED DATA PLUS
```

```
1597
1598  005460  017701  012054                  MOV   @RXDBUF,R1          ;RXERR & OVRRUN
1599  005464  020001                          CMP   R0,R1    ;COMPARE EXP VS. ACT
1600  005466  001401                          BEQ   66$
1601  005470  104002                          HLT   2        ;SPECIFICALLY LOOK AT RXERR &
1602                                                         ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1603  005472                          66$:
1604  005472  104400                          SCOPE
1605                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1606                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1607                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1608                                          ;;(OVRRUN,RXERR)
1609                                          ;;MODE:ISYMOD
1610                                          ;;LENGTH:EIGHT
1611                                          ;;CHAR:377
1612                                          ;;
1613  005474  012767  000013  173424  TST11:  MOV   #11,TSTNO          ;SAVE THIS
1614  005502  012767  005736  173406          MOV   #TST12,NEXT                ;GO TO THIS TEST WHEN THRU
1615  005510  052777  000400  012032          BIS   #MRESET,@TXCSR     ;MASTER RESET
1616  005516  012777  000000  012020          MOV   #ISYMOD,@PARCSR    ;SET THE MODE
1617  005524  052777  000400  012016          BIS   #MRESET,@TXCSR     ;MASTER RESET
1618
1619                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1620  005532  012777  064001  012010          MOV   #MTDATA!CLK!MINT!BREAK,@TXCSR
1621
1622                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1623  005540  012777  006000  011776          MOV   #ISYMOD!EIGHT!NOPAR!0,@PARCSR
1624  005546  052777  000020  011760          BIS   #SYNSCH,@RXCSR   ;SET SYNC SEARCH
1625                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1626  005554  042777  020000  011766          BIC   #CLK,@TXCSR        ;POKE CLK DOWN
1627  005562  052777  020000  011760          BIS   #CLK,@TXCSR        ;POKE CLK UP
1628                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1629  005570  042777  020000  011752          BIC   #CLK,@TXCSR        ;POKE CLK DOWN
1630  005576  052777  020000  011744          BIS   #CLK,@TXCSR        ;POKE CLK UP
1631  005604  016703  011730                  MOV   RXDBUF,R3          ;SET UP FOR ERROR MESSAGE
1632  005610  012700  000377                  MOV   #377,R0 ;EXPECTED
1633  005614  012767  000012  173316          MOV   #10.,SHIFT         ;# OF SHIFTS
1634  005622  012767  001776  173314          MOV   #1776,TEMP1        ;DATA CHAR
1635  005630  004767  011404                  JSR   PC,RPOKE           ;SHIFT IN THIS CHAR
1636  005634  105777  011674                  TSTB  @RXCSR   ;RXDONE ?
1637  005640  100401                          BMI   64$
1638  005642  104000                          HLT   ;RXDONE SHOULD BE SET
1639  005644                          64$:
1640  005644  017701  011670                  MOV   @RXDBUF,R1         ;ACTUAL
1641  005650  020001                          CMP   R0,R1    ;COMPARE EXPECTED VS. ACTUAL
1642  005652  001401                          BEQ   65$
1643  005654  104002                          HLT   2        ;RECEIVED DATA DID NOT MATCH
1644                                                         ;EXPECTED DATA - CHECK MAINT DATA
1645                                                         ;OR RECEIVER LOGIC
1646  005656                          65$:
1647  005656  012767  000012  173254          MOV   #10.,SHIFT         ;# OF SHIFTS
1648  005664  012767  001776  173252          MOV   #1776,TEMP1        ;DATA CHAR
1649  005672  004767  011342                  JSR   PC,RPOKE           ;SHIFT IN THIS CHAR
1650                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1651  005676  012767  000012  173234          MOV   #10.,SHIFT         ;# OF SHIFTS
1652  005704  012767  001776  173232          MOV   #1776,TEMP1        ;DATA CHAR
```

```
 1653  005712  004767  011322                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1654  005716  012700  140377                  MOV     #140000!377,R0  ;EXPECTED DATA PLUS
 1655                                                                    ;RXERR & OVRRUN
 1656  005722  017701  011612                  MOV     @RXDBUF,R1      ;ACTUAL
 1657  005726  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
 1658  005730  001401                          BEQ     66$
 1659  005732  104002                          HLT     2               ;SPECIFICALLY LOOK AT RXERR &
 1660                                                                    ;OVRRUN BITS...THEY BOTH SHOULD BE SET
 1661  005734                          66$:
 1662  005734  104400                          SCOPE
 1663                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 1664                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
 1665                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 1666                                          ;;(OVRRUN,RXERR)
 1667                                          ;;MODE:ISYMOD
 1668                                          ;;LENGTH:EIGHT
 1669                                          ;;CHAR:0
 1670                                          ;;
 1671  005736  012767  000014  173162  TST12:  MOV     #12,TSTNO       ;SAVE THIS
 1672  005744  012767  006200  173144          MOV     #TST13,NEXT             ;GO TO THIS TEST WHEN THRU
 1673  005752  052777  000400  011570          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1674  005760  012777  000000  011556          MOV     #ISYMOD,@PARCSR ;SET THE MODE
 1675  005766  052777  000400  011554          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1676
 1677                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1678  005774  012777  064001  011546          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1679
 1680                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1681  006002  012777  006000  011534          MOV     #ISYMOD!EIGHT!NOPAR!0,@PARCSR
 1682  006010  052777  000020  011516          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 1683                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 1684  006016  042777  020000  011524          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1685  006024  052777  020000  011516          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1686                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1687  006032  042777  020000  011510          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1688  006040  052777  020000  011502          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1689  006046  016703  011466                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1690  006052  012700  000000                  MOV     #0,R0   ;EXPECTED
 1691  006056  012767  000012  173054          MOV     #10.,SHIFT      ;# OF SHIFTS
 1692  006064  012767  001000  173052          MOV     #1000,TEMP1     ;DATA CHAR
 1693  006072  004767  011142                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1694  006076  105777  011432                  TSTB    @RXCSR  ;RXDONE ?
 1695  006102  100401                          BMI     64$
 1696  006104  104000                          HLT     ;RXDONE SHOULD BE SET
 1697  006106                          64$:
 1698  006106  017701  011426                  MOV     @RXDBUF,R1      ;ACTUAL
 1699  006112  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
 1700  006114  001401                          BEQ     65$
 1701  006116  104002                          HLT     2               ;RECEIVED DATA DID NOT MATCH
 1702                                                                    ;EXPECTED DATA - CHECK MAINT DATA
 1703                                                                    ;OR RECEIVER LOGIC
 1704  006120                          65$:
 1705  006120  012767  000012  173012          MOV     #10.,SHIFT      ;# OF SHIFTS
 1706  006126  012767  001000  173010          MOV     #1000,TEMP1     ;DATA CHAR
 1707  006134  004767  011100                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1708                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
```

```
 1709  006140  012767  000012  172772          MOV     #10.,SHIFT      ;# OF SHIFTS
 1710  006146  012767  001000  172770          MOV     #1000,TEMP1     ;DATA CHAR
 1711  006154  004767  011060                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1712  006160  012700  140000                  MOV     #140000!0,R0    ;EXPECTED DATA PLUS
 1713                                                                  ;RXERR & OVRRUN
 1714  006164  017701  011350                  MOV     @RXDBUF,R1      ;ACTUAL
 1715  006170  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
 1716  006172  001401                          BEQ     66$
 1717  006174  104002                          HLT     2               ;SPECIFICALLY LOOK AT RXERR &
 1718                                                                  ;OVRRUN BITS...THEY BOTH SHOULD BE SET
 1719  006176                          66$:
 1720  006176  104400                          SCOPE
 1721                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 1722                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
 1723                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 1724                                          ;;(OVRRUN,RXERR)
 1725                                          ;;MODE:SYNEXT
 1726                                          ;;LENGTH:FIVE
 1727                                          ;;CHAR:25
 1728                                          ;;
 1729  006200  012767  000015  172720  TST13:  MOV     #13,TSTNO       ;SAVE THIS
 1730  006206  012767  006426  172702          MOV     #TST14,NEXT             ;GO TO THIS TEST WHEN THRU
 1731  006214  052777  000400  011326          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1732  006222  012777  020000  011314          MOV     #SYNEXT,@PARCSR ;SET THE MODE
 1733  006230  052777  000400  011312          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1734
 1735                          ;SET MAINT DATA,CLK,BREAK,&MAINTEN[illegible] MODE
 1736  006236  012777  064001  011304          MOV     #MTDATA!CLK!MINT!BR[illegible],@TXCSR
 1737
 1738                          ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1739  006244  012777  020000  011272          MOV     #SYNEXT!FIVE!NOPAR!0,@PARCSR
 1740  006252  052777  000020  011254          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
 1741                          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1742  006260  042777  020000  011262          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1743  006266  052777  020000  011254          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1744  006274  016703  011240                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1745  006300  012700  000025                  MOV     #25,R0  ;EXPECTED
 1746  006304  012767  000005  172626          MOV     #5,SHIFT        ;# OF SHIFTS
 1747  006312  012767  000025  172624          MOV     #25,TEMP1       ;DATA CHAR
 1748  006320  004767  010714                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1749  006324  105777  011204                  TSTB    @RXCSR  ;RXDONE ?
 1750  006330  100401                          BMI     64$
 1751  006332  104000                          HLT     ;RXDONE SHOULD BE SET
 1752  006334                          64$:
 1753  006334  017701  011200                  MOV     @RXDBUF,R1      ;ACTUAL
 1754  006340  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
 1755  006342  001401                          BEQ     65$
 1756  006344  104002                          HLT     2               ;RECEIVED DATA DID NOT MATCH
 1757                                                                  ;EXPECTED DATA - CHECK MAINT DATA
 1758                                                                  ;OR RECEIVER LOGIC
 1759  006346                          65$:
 1760  006346  012767  000005  172564          MOV     #5,SHIFT        ;# OF SHIFTS
 1761  006354  012767  000025  172562          MOV     #25,TEMP1       ;DATA CHAR
 1762  006362  004767  010652                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1763                          ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
 1764  006366  012767  000005  172544          MOV     #5,SHIFT        ;# OF SHIFTS
```

```
 1765  006374  012767  000025  172542          MOV     #25,TEMP1       ;DATA CHAR
 1766  006402  004767  010632                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1767  006406  012700  140025                  MOV     #140000!25,R0   ;EXPECTED DATA PLUS
 1768                                                                  ;RXERR & OVRRUN
 1769  006412  017701  011122                  MOV     @RXDBUF,R1      ;ACTUAL
 1770  006416  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
 1771  006420  001401                          BEQ     66$
 1772  006422  104002                          HLT     2               ;SPECIFICALLY LOOK AT RXERR &
 1773                                                                  ;OVRRUN BITS...THEY BOTH SHOULD BE SET
 1774  006424                          66$:
 1775  006424  104400                          SCOPE
 1776                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 1777                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
 1778                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 1779                                          ;;(OVRRUN,RXERR)
 1780                                          ;;MODE:SYNEXT
 1781                                          ;;LENGTH:FIVE
 1782                                          ;;CHAR:12
 1783                                          ;;
 1784  006426  012767  000016  172472  TST14:  MOV     #14,TSTNO       ;SAVE THIS
 1785  006434  012767  006654  172454          MOV     #TST15,NEXT             ;GO TO THIS TEST WHEN THRU
 1786  006442  052777  000400  011100          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1787  006450  012777  020000  011066          MOV     #SYNEXT,@PARCSR ;SET THE MODE
 1788  006456  052777  000400  011064          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1789
 1790                          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1791  006464  012777  064001  011056          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1792
 1793                          ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1794  006472  012777  020000  011044          MOV     #SYNEXT!FIVE!NOPAR!0,@PARCSR
 1795  006500  052777  000020  011026          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
 1796                          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1797  006506  042777  020000  011034          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1798  006514  052777  020000  011026          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1799  006522  016703  011012                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1800  006526  012700  000012                  MOV     #12,R0  ;EXPECTED
 1801  006532  012767  000005  172400          MOV     #5,SHIFT        ;# OF SHIFTS
 1802  006540  012767  000012  172376          MOV     #12,TEMP1       ;DATA CHAR
 1803  006546  004767  010466                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1804  006552  105777  010756                  TSTB    @RXCSR  ;RXDONE ?
 1805  006556  100401                          BMI     64$
 1806  006560  104000                          HLT     ;RXDONE SHOULD BE SET
 1807  006562                          64$:
 1808  006562  017701  010752                  MOV     @RXDBUF,R1      ;ACTUAL
 1809  006566  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
 1810  006570  001401                          BEQ     65$
 1811  006572  104002                          HLT     2               ;RECEIVED DATA DID NOT MATCH
 1812                                                                  ;EXPECTED DATA - CHECK MAINT DATA
 1813                                                                  ;OR RECEIVER LOGIC
 1814  006574                          65$:
 1815  006574  012767  000005  172336          MOV     #5,SHIFT        ;# OF SHIFTS
 1816  006602  012767  000012  172334          MOV     #12,TEMP1       ;DATA CHAR
 1817  006610  004767  010424                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1818                          ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
 1819  006614  012767  000005  172316          MOV     #5,SHIFT        ;# OF SHIFTS
 1820  006622  012767  000012  172314          MOV     #12,TEMP1       ;DATA CHAR
```

```
 1821  006630  004767  010404                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1822  006634  012700  140012                  MOV     #140000!12,R0   ;EXPECTED DATA PLUS
 1823                                                                    ;RXERR & OVRRUN
 1824  006640  017701  010674                  MOV     @RXDBUF,R1      ;ACTUAL
 1825  006644  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
 1826  006646  001401                          BEQ     66$
 1827  006650  104002                          HLT     2               ;SPECIFICALLY LOOK AT RXERR &
 1828                                                                    ;OVRRUN BITS...THEY BOTH SHOULD BE SET
 1829  006652                          66$:
 1830  006652  104400                          SCOPE
 1831                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 1832                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
 1833                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 1834                                          ;;(OVRRUN,RXERR)
 1835                                          ;;MODE:SYNEXT
 1836                                          ;;LENGTH:FIVE
 1837                                          ;;CHAR:37
 1838                                          ;;
 1839  006654  012767  000017  172244  TST15:  MOV     #15,TSTNO       ;SAVE THIS
 1840  006662  012767  007102  172226          MOV     #TST16,NEXT             ;GO TO THIS TEST WHEN THRU
 1841  006670  052777  000400  010652          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1842  006676  012777  020000  010640          MOV     #SYNEXT,@PARCSR ;SET THE MODE
 1843  006704  052777  000400  010636          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1844
 1845                          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1846  006712  012777  064001  010630          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1847
 1848                          ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1849  006720  012777  020000  010616          MOV     #SYNEXT!FIVE!NOPAR!0,@PARCSR
 1850  006726  052777  000020  010600          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
 1851                          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1852  006734  042777  020000  010606          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1853  006742  052777  020000  010600          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1854  006750  016703  010564                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1855  006754  012700  000037                  MOV     #37,R0  ;EXPECTED
 1856  006760  012767  000005  172152          MOV     #5,SHIFT        ;# OF SHIFTS
 1857  006766  012767  000037  172150          MOV     #37,TEMP1       ;DATA CHAR
 1858  006774  004767  010240                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1859  007000  105777  010530                  TSTB    @RXCSR  ;RXDONE ?
 1860  007004  100401                          BMI     64$
 1861  007006  104000                          HLT     ;RXDONE SHOULD BE SET
 1862  007010                          64$:
 1863  007010  017701  010524                  MOV     @RXDBUF,R1      ;ACTUAL
 1864  007014  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
 1865  007016  001401                          BEQ     65$
 1866  007020  104002                          HLT     2               ;RECEIVED DATA DID NOT MATCH
 1867                                                                    ;EXPECTED DATA - CHECK MAINT DATA
 1868                                                                    ;OR RECEIVER LOGIC
 1869  007022                          65$:
 1870  007022  012767  000005  172110          MOV     #5,SHIFT        ;# OF SHIFTS
 1871  007030  012767  000037  172106          MOV     #37,TEMP1       ;DATA CHAR
 1872  007036  004767  010176                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1873                          ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
 1874  007042  012767  000005  172070          MOV     #5,SHIFT        ;# OF SHIFTS
 1875  007050  012767  000037  172066          MOV     #37,TEMP1       ;DATA CHAR
 1876  007056  004767  010156                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
```

```
1877  007062  012700  140037                  MOV     #140000!37,R0   ;EXPECTED DATA PLUS
1878                                                                    ;RXERR & OVRRUN
1879  007066  017701  010446                  MOV     @RXDBUF,R1      ;ACTUAL
1880  007072  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
1881  007074  001401                          BEQ     66$
1882  007076  104002                          HLT     2               ;SPECIFICALLY LOOK AT RXERR &
1883                                                                    ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1884  007100                          66$:
1885  007100  104400                          SCOPE
1886                                           ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1887                                           ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1888                                           ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1889                                           ;;(OVRRUN,RXERR)
1890                                           ;;MODE:SYNEXT
1891                                           ;;LENGTH:FIVE
1892                                           ;;CHAR:0
1893                                           ;;
1894  007102  012767  000020  172016  TST16:   MOV     #16,TSTNO       ;SAVE THIS
1895  007110  012767  007330  172000          MOV     #TST17,NEXT             ;GO TO THIS TEST WHEN THRU
1896  007116  052777  000400  010424          BIS     #MRESET,@TXCSR  ;MASTER RESET
1897  007124  012777  020000  010412          MOV     #SYNEXT,@PARCSR ;SET THE MODE
1898  007132  052777  000400  010410          BIS     #MRESET,@TXCSR  ;MASTER RESET
1899
1900                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1901  007140  012777  064001  010402          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1902
1903                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1904  007146  012777  020000  010370          MOV     #SYNEXT!FIVE!NOPAR!0,@PARCSR
1905  007154  052777  000020  010352          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1906                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1907  007162  042777  020000  010360          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1908  007170  052777  020000  010352          BIS     #CLK,@TXCSR     ;POKE CLK UP
1909  007176  016703  010336                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1910  007202  012700  000000                  MOV     #0,R0   ;EXPECTED
1911  007206  012767  000005  171724          MOV     #5,SHIFT        ;# OF SHIFTS
1912  007214  012767  000000  171722          MOV     #0,TEMP1        ;DATA CHAR
1913  007222  004767  010012                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1914  007226  105777  010302                  TSTB    @RXCSR  ;RXDONE ?
1915  007232  100401                          BMI     64$
1916  007234  104000                          HLT     ;RXDONE SHOULD BE SET
1917  007236                          64$:
1918  007236  017701  010276                  MOV     @RXDBUF,R1      ;ACTUAL
1919  007242  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1920  007244  001401                          BEQ     65$
1921  007246  104002                          HLT     2               ;RECEIVED DATA DID NOT MATCH
1922                                                                    ;EXPECTED DATA - CHECK MAINT DATA
1923                                                                    ;OR RECEIVER LOGIC
1924  007250                          65$:
1925  007250  012767  000005  171662          MOV     #5,SHIFT        ;# OF SHIFTS
1926  007256  012767  000000  171660          MOV     #0,TEMP1        ;DATA CHAR
1927  007264  004767  007750                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1928                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1929  007270  012767  000005  171642          MOV     #5,SHIFT        ;# OF SHIFTS
1930  007276  012767  000000  171640          MOV     #0,TEMP1        ;DATA CHAR
1931  007304  004767  007730                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1932  007310  012700  140000                  MOV     #140000!0,R0    ;EXPECTED DATA PLUS
```

```
1933
1934  007314  017701  010220                    MOV     @RXDBUF,R1      ;RXERR & OVRRUN
1935  007320  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
1936  007322  001401                            BEQ     66$
1937  007324  104002                            HLT     2               ;SPECIFICALLY LOOK AT RXERR &
1938                                                                      ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1939  007326                          66$:
1940  007326  104400                            SCOPE
1941                                            ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1942                                            ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1943                                            ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1944                                            ;;(OVRRUN,RXERR)
1945                                            ;;MODE:SYNEXT
1946                                            ;;LENGTH:SIX
1947                                            ;;CHAR:25
1948                                            ;;
1949  007330  012767  000021  171570    TST17:  MOV     #17,TSTNO       ;SAVE THIS
1950  007336  012767  007556  171552            MOV     #TST18,NEXT             ;GO TO THIS TEST WHEN THRU
1951  007344  052777  000400  010176            BIS     #MRESET,@TXCSR  ;MASTER RESET
1952  007352  012777  020000  010164            MOV     #SYNEXT,@PARCSR ;SET THE MODE
1953  007360  052777  000400  010162            BIS     #MRESET,@TXCSR  ;MASTER RESET
1954
1955                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1956  007366  012777  064001  010154            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1957
1958                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1959  007374  012777  022000  010142            MOV     #SYNEXT!SIX!NOPAR!0,@PARCSR
1960  007402  052777  000020  010124            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1961                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1962  007410  042777  020000  010132            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1963  007416  052777  020000  010124            BIS     #CLK,@TXCSR     ;POKE CLK UP
1964  007424  016703  010110                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1965  007430  012700  000025                    MOV     #25,R0  ;EXPECTED
1966  007434  012767  000006  171476            MOV     #6,SHIFT        ;# OF SHIFTS
1967  007442  012767  000025  171474            MOV     #25,TEMP1       ;DATA CHAR
1968  007450  004767  007564                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1969  007454  105777  010054                    TSTB    @RXCSR  ;RXDONE ?
1970  007460  100401                            BMI     64$
1971  007462  104000                            HLT     ;RXDONE SHOULD BE SET
1972  007464                          64$:
1973  007464  017701  010050                    MOV     @RXDBUF,R1      ;ACTUAL
1974  007470  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1975  007472  001401                            BEQ     65$
1976  007474  104002                            HLT     2               ;RECEIVED DATA DID NOT MATCH
1977                                                                      ;EXPECTED DATA - CHECK MAINT DATA
1978                                                                      ;OR RECEIVER LOGIC
1979  007476                          65$:
1980  007476  012767  000006  171434            MOV     #5,SHIFT        ;# OF SHIFTS
1981  007504  012767  000025  171432            MOV     #25,TEMP1       ;DATA CHAR
1982  007512  004767  007522                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1983                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1984  007516  012767  000006  171414            MOV     #6,SHIFT        ;# OF SHIFTS
1985  007524  012767  000025  171412            MOV     #25,TEMP1       ;DATA CHAR
1986  007532  004767  007502                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1987  007536  012700  140025                    MOV     #140000!25,R0   ;EXPECTED DATA PLUS
1988                                                                    ;RXERR & OVRRUN
```

```
 1989  007542  017701  007772                    MOV     @RXDBUF,R1      ;ACTUAL
 1990  007546  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
 1991  007550  001401                            BEQ     66$
 1992  007552  104002                            HLT     2               ;SPECIFICALLY LOOK AT RXERR &
 1993                                                                    ;OVRRUN BITS...THEY BOTH SHOULD BE SET
 1994  007554                          66$:
 1995  007554  104400                            SCOPE
 1996                                            ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 1997                                            ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
 1998                                            ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 1999                                            ;;(OVRRUN,RXERR)
 2000                                            ;;MODE:SYNEXT
 2001                                            ;;LENGTH:SIX
 2002                                            ;;CHAR:52
 2003                                            ;;
 2004  007556  012767  000022  171342  TST18:    MOV     #18,TSTNO       ;SAVE THIS
 2005  007564  012767  010004  171324            MOV     #TST19,NEXT             ;GO TO THIS TEST WHEN THRU
 2006  007572  052777  000400  007750            BIS     #MRESET,@TXCSR  ;MASTER RESET
 2007  007600  012777  020000  007736            MOV     #SYNEXT,@PARCSR ;SET THE MODE
 2008  007606  052777  000400  007734            BIS     #MRESET,@TXCSR  ;MASTER RESET
 2009
 2010                          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 2011  007614  012777  064001  007726            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 2012
 2013                          ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 2014  007622  012777  022000  007714            MOV     #SYNEXT!SIX!NOPAR!0,@PARCSR
 2015  007630  052777  000020  007676            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
 2016                          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 2017  007636  042777  020000  007704            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 2018  007644  052777  020000  007676            BIS     #CLK,@TXCSR     ;POKE CLK UP
 2019  007652  016703  007662                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 2020  007656  012700  000052                    MOV     #52,R0  ;EXPECTED
 2021  007662  012767  000006  171250            MOV     #6,SHIFT        ;# OF SHIFTS
 2022  007670  012767  000052  171246            MOV     #52,TEMP1       ;DATA CHAR
 2023  007676  004767  007336                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 2024  007702  105777  007626                    TSTB    @RXCSR  ;RXDONE ?
 2025  007706  100401                            BMI     64$
 2026  007710  104000                            HLT     ;RXDONE SHOULD BE SET
 2027  007712                          64$:
 2028  007712  017701  007622                    MOV     @RXDBUF,R1      ;ACTUAL
 2029  007716  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
 2030  007720  001401                            BEQ     65$
 2031  007722  104002                            HLT     2               ;RECEIVED DATA DID NOT MATCH
 2032                                                                    ;EXPECTED DATA - CHECK MAINT DATA
 2033                                                                    ;OR RECEIVER LOGIC
 2034  007724                          65$:
 2035  007724  012767  000006  171206            MOV     #6,SHIFT        ;# OF SHIFTS
 2036  007732  012767  000052  171204            MOV     #52,TEMP1       ;DATA CHAR
 2037  007740  004767  007274                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 2038                          ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
 2039  007744  012767  000006  171166            MOV     #6,SHIFT        ;# OF SHIFTS
 2040  007752  012767  000052  171164            MOV     #52,TEMP1       ;DATA CHAR
 2041  007760  004767  007254                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 2042  007764  012700  140052                    MOV     #140000!52,R0   ;EXPECTED DATA PLUS
 2043                                                                    ;RXERR & OVRRUN
 2044  007770  017701  007544                    MOV     @RXDBUF,R1      ;ACTUAL
```

```
2045  007774  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
2046  007776  001401                          BEQ     66$
2047  010000  104002                          HLT     2       ;SPECIFICALLY LOOK AT RXERR &
2048                                                          ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2049  010002                          66$:
2050  010002  104400                          SCOPE
2051                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2052                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2053                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2054                                          ;;(OVRRUN,RXERR)
2055                                          ;;MODE:SYNEXT
2056                                          ;;LENGTH:SIX
2057                                          ;;CHAR:77
2058                                          ;;
2059  010004  012767  000023  171114  TST19:  MOV     #19,TSTNO       ;SAVE THIS
2060  010012  012767  010232  171076          MOV     #TST20,NEXT             ;GO TO THIS TEST WHEN THRU
2061  010020  052777  000400  007522          BIS     #MRESET,@TXCSR  ;MASTER RESET
2062  010026  012777  020000  007510          MOV     #SYNEXT,@PARCSR ;SET THE MODE
2063  010034  052777  000400  007506          BIS     #MRESET,@TXCSR  ;MASTER RESET
2064
2065                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2066  010042  012777  064001  007500          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2067
2068                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2069  010050  012777  022000  007466          MOV     #SYNEXT!SIX!NOPAR!0,@PARCSR
2070  010056  052777  000020  007450          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2071                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2072  010064  042777  020000  007456          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2073  010072  052777  020000  007450          BIS     #CLK,@TXCSR     ;POKE CLK UP
2074  010100  016703  007434                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2075  010104  012700  000077                  MOV     #77,R0  ;EXPECTED
2076  010110  012767  000006  171022          MOV     #6,SHIFT        ;# OF SHIFTS
2077  010116  012767  000077  171020          MOV     #77,TEMP1       ;DATA CHAR
2078  010124  004767  007110                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2079  010130  105777  007400                  TSTB    @RXCSR  ;RXDONE ?
2080  010134  100401                          BMI     64$
2081  010136  104000                          HLT     ;RXDONE SHOULD BE SET
2082  010140                          64$:
2083  010140  017701  007374                  MOV     @RXDBUF,R1      ;ACTUAL
2084  010144  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2085  010146  001401                          BEQ     65$
2086  010150  104002                          HLT     2       ;RECEIVED DATA DID NOT MATCH
2087                                                          ;EXPECTED DATA - CHECK MAINT DATA
2088                                                          ;OR RECEIVER LOGIC
2089  010152                          65$:
2090  010152  012767  000006  170760          MOV     #6,SHIFT        ;# OF SHIFTS
2091  010160  012767  000077  170756          MOV     #77,TEMP1       ;DATA CHAR
2092  010166  004767  007046                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2093                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2094  010172  012767  000006  170740          MOV     #6,SHIFT        ;# OF SHIFTS
2095  010200  012767  000077  170736          MOV     #77,TEMP1       ;DATA CHAR
2096  010206  004767  007026                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2097  010212  012700  140077                  MOV     #140000!77,R0   ;EXPECTED DATA PLUS
2098                                                          ;RXERR & OVRRUN
2099  010216  017701  007316                  MOV     @RXDBUF,R1      ;ACTUAL
2100  010222  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
```

```
2101  010224  001401                        BEQ     66$
2102  010226  104002                        HLT     2               ;SPECIFICALLY LOOK AT RXERR &
2103                                                                ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2104  010230                          66$:
2105  010230  104400                        SCOPE
2106                                        ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2107                                        ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2108                                        ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2109                                        ;;(OVRRUN,RXERR)
2110                                        ;;MODE:SYNEXT
2111                                        ;;LENGTH:SIX
2112                                        ;;CHAR:0
2113                                        ;;
2114  010232  012767  000024  170666  TST20: MOV     #20,TSTNO       ;SAVE THIS
2115  010240  012767  010460  170650        MOV     #TST21,NEXT             ;GO TO THIS TEST WHEN THRU
2116  010246  052777  000400  007274        BIS     #MRESET,@TXCSR  ;MASTER RESET
2117  010254  012777  020000  007262        MOV     #SYNEXT,@PARCSR ;SET THE MODE
2118  010262  052777  000400  007260        BIS     #MRESET,@TXCSR  ;MASTER RESET
2119
2120                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2121  010270  012777  064001  007252        MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2122
2123                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2124  010276  012777  022000  007240        MOV     #SYNEXT!SIX!NOPAR!0,@PARCSR
2125  010304  052777  000020  007222        BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2126                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2127  010312  042777  020000  007230        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2128  010320  052777  020000  007222        BIS     #CLK,@TXCSR     ;POKE CLK UP
2129  010326  016703  007206                MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2130  010332  012700  000000                MOV     #0,R0   ;EXPECTED
2131  010336  012767  000006  170574        MOV     #6,SHIFT        ;# OF SHIFTS
2132  010344  012767  000000  170572        MOV     #0,TEMP1        ;DATA CHAR
2133  010352  004767  006662                JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2134  010356  105777  007152                TSTB    @RXCSR  ;RXDONE ?
2135  010362  100401                        BMI     64$
2136  010364  104000                        HLT     ;RXDONE SHOULD BE SET
2137  010366                          64$:
2138  010366  017701  007146                MOV     @RXDBUF,R1      ;ACTUAL
2139  010372  020001                        CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2140  010374  001401                        BEQ     65$
2141  010376  104002                        HLT     2               ;RECEIVED DATA DID NOT MATCH
2142                                                                ;EXPECTED DATA - CHECK MAINT DATA
2143                                                                ;OR RECEIVER LOGIC
2144  010400                          65$:
2145  010400  012767  000006  170532        MOV     #6,SHIFT        ;# OF SHIFTS
2146  010406  012767  000000  170530        MOV     #0,TEMP1        ;DATA CHAR
2147  010414  004767  006620                JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2148                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2149  010420  012767  000006  170512        MOV     #6,SHIFT        ;# OF SHIFTS
2150  010426  012767  000000  170510        MOV     #0,TEMP1        ;DATA CHAR
2151  010434  004767  006600                JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2152  010440  012700  140000                MOV     #140000!0,R0    ;EXPECTED DATA PLUS
2153                                                                ;RXERR & OVRRUN
2154  010444  017701  007070                MOV     @RXDBUF,R1      ;ACTUAL
2155  010450  020001                        CMP     R0,R1   ;COMPARE EXP VS. ACT
2156  010452  001401                        BEQ     66$
```

```
2157  010454  104002                          HLT     2       ;SPECIFICALLY LOOK AT RXERR &
2158                                                          ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2159  010456                          66$:
2160  010456  104400                          SCOPE
2161                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2162                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2163                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2164                                          ;;(OVRRUN,RXERR)
2165                                          ;;MODE:SYNEXT
2166                                          ;;LENGTH:SEVEN
2167                                          ;;CHAR:125
2168                                          ;;
2169  010460  012767  000025  170440  TST21:  MOV     #21,TSTNO       ;SAVE THIS
2170  010466  012767  010706  170422          MOV     #TST22,NEXT             ;GO TO THIS TEST WHEN THRU
2171  010474  052777  000400  007046          BIS     #MRESET,@TXCSR  ;MASTER RESET
2172  010502  012777  020000  007034          MOV     #SYNEXT,@PARCSR ;SET THE MODE
2173  010510  052777  000400  007032          BIS     #MRESET,@TXCSR  ;MASTER RESET
2174
2175                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2176  010516  012777  064001  007024          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2177
2178                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2179  010524  012777  024000  007012          MOV     #SYNEXT!SEVEN!NOPAR!0,@PARCSR
2180  010532  052777  000020  006774          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2181                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2182  010540  042777  020000  007002          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2183  010546  052777  020000  006774          BIS     #CLK,@TXCSR     ;POKE CLK UP
2184  010554  016703  006760                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2185  010560  012700  000125                  MOV     #125,R0 ;EXPECTED
2186  010564  012767  000007  170346          MOV     #7,SHIFT        ;# OF SHIFTS
2187  010572  012767  000125  170344          MOV     #125,TEMP1      ;DATA CHAR
2188  010600  004767  006434                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2189  010604  105777  006724                  TSTB    @RXCSR  ;RXDONE ?
2190  010610  100401                          BMI     64$
2191  010612  104000                          HLT     ;RXDONE SHOULD BE SET
2192  010614                          64$:
2193  010614  017701  006720                  MOV     @RXDBUF,R1      ;ACTUAL
2194  010620  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2195  010622  001401                          BEQ     65$
2196  010624  104002                          HLT     2       ;RECEIVED DATA DID NOT MATCH
2197                                                          ;EXPECTED DATA - CHECK MAINT DATA
2198                                                          ;OR RECEIVER LOGIC
2199  010626                          65$:
2200  010626  012767  000007  170304          MOV     #7,SHIFT        ;# OF SHIFTS
2201  010634  012767  000125  170302          MOV     #125,TEMP1      ;DATA CHAR
2202  010642  004767  006372                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2203                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2204  010646  012767  000007  170264          MOV     #7,SHIFT        ;# OF SHIFTS
2205  010654  012767  000125  170262          MOV     #125,TEMP1      ;DATA CHAR
2206  010662  004767  006352                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2207  010666  012700  140125                  MOV     #140000!125,R0  ;EXPECTED DATA PLUS
2208                                                          ;RXERR & OVRRUN
2209  010672  017701  006642                  MOV     @RXDBUF,R1      ;ACTUAL
2210  010676  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
2211  010700  001401                          BEQ     66$
2212  010702  104002                          HLT     2       ;SPECIFICALLY LOOK AT RXERR &
```

```
                                                        ;OVRRUN BITS...THEY BOTH SHOULD BE SET
010704                                          66$:
010704  104400                                        SCOPE
                                                    ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
                                                    ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
                                                    ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
                                                    ;;(OVRRUN,RXERR)
                                                    ;;MODE:SYNEXT
                                                    ;;LENGTH:SEVEN
                                                    ;;CHAR:52
                                                    ;;
010706  012767  000026  170212            TST22:    MOV     #22,TSTNO       ;SAVE THIS
010714  012767  011134  170174                      MOV     #TST23,NEXT             ;GO TO THIS TEST WHEN THRU
010722  052777  000400  006620                      BIS     #MRESET,@TXCSR  ;MASTER RESET
010730  012777  020000  006606                      MOV     #SYNEXT,@PARCSR ;SET THE MODE
010736  052777  000400  006604                      BIS     #MRESET,@TXCSR  ;MASTER RESET

                                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
010744  012777  064001  006576                      MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
010752  012777  024000  006564                      MOV     #SYNEXT!SEVEN!NOPAR!0,@PARCSR
010760  052777  000020  006546                      BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
                                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
010766  042777  020000  006554                      BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010774  052777  020000  006546                      BIS     #CLK,@TXCSR     ;POKE CLK UP
011002  016703  006532                            MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
011006  012700  000052                            MOV     #52,R0  ;EXPECTED
011012  012767  000007  170120                    MOV     #7,SHIFT        ;# OF SHIFTS
011020  012767  000052  170116                    MOV     #52,TEMP1       ;DATA CHAR
011026  004767  006206                            JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
011032  105777  006476                            TSTB    @RXCSR  ;RXDONE ?
011036  100401                                    BMI     64$
011040  104000                                    HLT     ;RXDONE SHOULD BE SET
011042                                  64$:
011042  017701  006472                            MOV     @RXDBUF,R1      ;ACTUAL
011046  020001                                    CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
011050  001401                                    BEQ     65$
011052  104002                                    HLT     2               ;RECEIVED DATA DID NOT MATCH
                                                                            ;EXPECTED DATA - CHECK MAINT DATA
                                                                            ;OR RECEIVER LOGIC
011054                                  65$:
011054  012767  000007  170056                    MOV     #7,SHIFT        ;# OF SHIFTS
011062  012767  000052  170054                    MOV     #52,TEMP1       ;DATA CHAR
011070  004767  006144                            JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
                                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
011074  012767  000007  170036                    MOV     #7,SHIFT        ;# OF SHIFTS
011102  012767  000052  170034                    MOV     #52,TEMP1       ;DATA CHAR
011110  004767  006124                            JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
011114  012700  140052                            MOV     #140000!52,R0   ;EXPECTED DATA PLUS
                                                                            ;RXERR & OVRRUN
011120  017701  006414                            MOV     @RXDBUF,R1      ;ACTUAL
011124  020001                                    CMP     R0,R1   ;COMPARE EXP VS. ACT
011126  001401                                    BEQ     66$
011130  104002                                    HLT     2               ;SPECIFICALLY LOOK AT RXERR &
                                                                            ;OVRRUN BITS...THEY BOTH SHOULD BE SET
```

```
2269  011132                                66$:
2270  011132  104400                              SCOPE
2271                                              ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2272                                              ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2273                                              ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2274                                              ;;(OVRRUN,RXERR)
2275                                              ;;MODE:SYNEXT
2276                                              ;;LENGTH:SEVEN
2277                                              ;;CHAR:177
2278                                              ;;
2279  011134  012767  000027  167764  TST23:  MOV     #23,TSTNO       ;SAVE THIS
2280  011142  012767  011362  167746           MOV     #TST24,NEXT             ;GO TO THIS TEST WHEN THRU
2281  011150  052777  000400  006372           BIS     #MRESET,@TXCSR  ;MASTER RESET
2282  011156  012777  020000  006360           MOV     #SYNEXT,@PARCSR ;SET THE MODE
2283  011164  052777  000400  006356           BIS     #MRESET,@TXCSR  ;MASTER RESET
2284
2285                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2286  011172  012777  064001  006350           MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2287
2288                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2289  011200  012777  024000  006336           MOV     #SYNEXT!SEVEN!NOPAR!0,@PARCSR
2290  011206  052777  000020  006320           BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2291                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2292  011214  042777  020000  006326           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2293  011222  052777  020000  006320           BIS     #CLK,@TXCSR     ;POKE CLK UP
2294  011230  016703  006304                   MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2295  011234  012700  000177                   MOV     #177,R0 ;EXPECTED
2296  011240  012767  000007  167672           MOV     #7,SHIFT        ;# OF SHIFTS
2297  011246  012767  000177  167670           MOV     #177,TEMP1      ;DATA CHAR
2298  011254  004767  005760                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2299  011260  105777  006250                   TSTB    @RXCSR  ;RXDONE ?
2300  011264  100401                           BMI     64$
2301  011266  104000                           HLT     ;RXDONE SHOULD BE SET
2302  011270                          64$:
2303  011270  017701  006244                   MOV     @RXDBUF,R1      ;ACTUAL
2304  011274  020001                           CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2305  011276  001401                           BEQ     65$
2306  011300  104002                           HLT     2       ;RECEIVED DATA DID NOT MATCH
2307                                                    ;EXPECTED DATA - CHECK MAINT DATA
2308                                                    ;OR RECEIVER LOGIC
2309  011302                          65$:
2310  011302  012767  000007  167630           MOV     #7,SHIFT        ;# OF SHIFTS
2311  011310  012767  000177  167626           MOV     #177,TEMP1      ;DATA CHAR
2312  011316  004767  005716                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2313                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2314  011322  012767  000007  167610           MOV     #7,SHIFT        ;# OF SHIFTS
2315  011330  012767  000177  167606           MOV     #177,TEMP1      ;DATA CHAR
2316  011336  004767  005676                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2317  011342  012700  140177                   MOV     #140000!177,R0  ;EXPECTED DATA PLUS
2318                                                    ;RXERR & OVRRUN
2319  011346  017701  006166                   MOV     @RXDBUF,R1      ;ACTUAL
2320  011352  020001                           CMP     R0,R1   ;COMPARE EXP VS. ACT
2321  011354  001401                           BEQ     66$
2322  011356  104002                           HLT     2       ;SPECIFICALLY LOOK AT RXERR &
2323                                                    ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2324  011360                          66$:
```

```
2325  011360  104400                          SCOPE
2326                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2327                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2328                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2329                                          ;;(OVRRUN,RXERR)
2330                                          ;;MODE:SYNEXT
2331                                          ;;LENGTH:SEVEN
2332                                          ;;CHAR:0
2333                                          ;;
2334  011362  012767  000030  167536  TST24:  MOV     #24,TSTNO       ;SAVE THIS
2335  011370  012767  011610  167520          MOV     #TST25,NEXT             ;GO TO THIS TEST WHEN THRU
2336  011376  052777  000400  006144          BIS     #MRESET,@TXCSR  ;MASTER RESET
2337  011404  012777  020000  006132          MOV     #SYNEXT,@PARCSR ;SET THE MODE
2338  011412  052777  000400  006130          BIS     #MRESET,@TXCSR  ;MASTER RESET
2339
2340                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2341  011420  012777  064001  006122          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2342
2343                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2344  011426  012777  024000  006110          MOV     #SYNEXT!SEVEN!NOPAR!0,@PARCSR
2345  011434  052777  000020  006072          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2346                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2347  011442  042777  020000  006100          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2348  011450  052777  020000  006072          BIS     #CLK,@TXCSR     ;POKE CLK UP
2349  011456  016703  006056                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2350  011462  012700  000000                  MOV     #0,R0   ;EXPECTED
2351  011466  012767  000007  167444          MOV     #7,SHIFT        ;# OF SHIFTS
2352  011474  012767  000000  167442          MOV     #0,TEMP1        ;DATA CHAR
2353  011502  004767  005532                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2354  011506  105777  006022                  TSTB    @RXCSR  ;RXDONE ?
2355  011512  100401                          BMI     64$
2356  011514  104000                          HLT     ;RXDONE SHOULD BE SET
2357  011516                          64$:
2358  011516  017701  006016                  MOV     @RXDBUF,R1      ;ACTUAL
2359  011522  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2360  011524  001401                          BEQ     65$
2361  011526  104002                          HLT     2               ;RECEIVED DATA DID NOT MATCH
2362                                                                  ;EXPECTED DATA - CHECK MAINT DATA
2363                                                                  ;OR RECEIVER LOGIC
2364  011530                          65$:
2365  011530  012767  000007  167402          MOV     #7,SHIFT        ;# OF SHIFTS
2366  011536  012767  000000  167400          MOV     #0,TEMP1        ;DATA CHAR
2367  011544  004767  005470                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2368                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2369  011550  012767  000007  167362          MOV     #7,SHIFT        ;# OF SHIFTS
2370  011556  012767  000000  167360          MOV     #0,TEMP1        ;DATA CHAR
2371  011564  004767  005450                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2372  011570  012700  140000                  MOV     #140000!0,R0    ;EXPECTED DATA PLUS
2373                                                                  ;RXERR & OVRRUN
2374  011574  017701  005740                  MOV     @RXDBUF,R1      ;ACTUAL
2375  011600  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
2376  011602  001401                          BEQ     66$
2377  011604  104002                          HLT     2               ;SPECIFICALLY LOOK AT RXERR &
2378                                                                  ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2379  011606                          66$:
2380  011606  104400                          SCOPE
```

```
                                        ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
                                        ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
                                        ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
                                        ;;(OVRRUN,RXERR)
                                        ;;MODE:SYNEXT
                                        ;;LENGTH:EIGHT
                                        ;;CHAR:125
                                        ;;
011610  012767  000031  167310  TST25:  MOV     #25,TSTNO       ;SAVE THIS
011616  012767  012036  167272           MOV     #TST26,NEXT             ;GO TO THIS TEST WHEN THRU
011624  052777  000400  005716           BIS     #MRESET,@TXCSR  ;MASTER RESET
011632  012777  020000  005704           MOV     #SYNEXT,@PARCSR ;SET THE MODE
011640  052777  000400  005702           BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
011646  012777  064001  005674           MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
011654  012777  026000  005662           MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
011662  052777  000020  005644           BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
011670  042777  020000  005652           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
011676  052777  020000  005644           BIS     #CLK,@TXCSR     ;POKE CLK UP
011704  016703  005630                   MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
011710  012700  000125                   MOV     #125,R0 ;EXPECTED
011714  012767  000010  167216           MOV     #8.,SHIFT       ;# OF SHIFTS
011722  012767  000125  167214           MOV     #125,TEMP1      ;DATA CHAR
011730  004767  005304                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
011734  105777  005574                   TSTB    @RXCSR  ;RXDONE ?
011740  100401                           BMI     64$
011742  104000                           HLT     ;RXDONE SHOULD BE SET
011744                          64$:
011744  017701  005570                   MOV     @RXDBUF,R1      ;ACTUAL
011750  020001                           CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
011752  001401                           BEQ     65$
011754  104002                           HLT     2               ;RECEIVED DATA DID NOT MATCH
                                                                 ;EXPECTED DATA - CHECK MAINT DATA
                                                                 ;OR RECEIVER LOGIC
011756                          65$:
011756  012767  000010  167154           MOV     #8.,SHIFT       ;# OF SHIFTS
011764  012767  000125  167152           MOV     #125,TEMP1      ;DATA CHAR
011772  004767  005242                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
012000  012767  000010  167134           MOV     #8.,SHIFT       ;# OF SHIFTS
012004  012767  000125  167132           MOV     #125,TEMP1      ;DATA CHAR
012012  004767  005222                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
012016  012700  140125                   MOV     #140000!125,R0  ;EXPECTED DATA PLUS
                                                                 ;RXERR & OVRRUN
012022  017701  005512                   MOV     @RXDBUF,R1      ;ACTUAL
012026  020001                           CMP     R0,R1   ;COMPARE EXP VS. ACT
012030  001401                           BEQ     66$
012032  104002                           HLT     2               ;SPECIFICALLY LOOK AT RXERR &
                                                                 ;OVRRUN BITS...THEY BOTH SHOULD BE SET
012034                          66$:
012034  104400                           SCOPE
                                        ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
```

```
2437                                            ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2438                                            ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2439                                            ;;(OVRRUN,RXERR)
2440                                            ;;MODE:SYNEXT
2441                                            ;;LENGTH:EIGHT
2442                                            ;;CHAR:252
2443                                            ;;
2444  012036  012767  000032  167062  TST26:    MOV     #26,TSTNO       ;SAVE THIS
2445  012044  012767  012264  167044            MOV     #TST27,NEXT             ;GO TO THIS TEST WHEN THRU
2446  012052  052777  000400  005470            BIS     #MRESET,@TXCSR  ;MASTER RESET
2447  012060  012777  020000  005456            MOV     #SYNEXT,@PARCSR ;SET THE MODE
2448  012066  052777  000400  005454            BIS     #MRESET,@TXCSR  ;MASTER RESET
2449
2450                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2451  012074  012777  064001  005446            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2452
2453                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2454  012102  012777  026000  005434            MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
2455  012110  052777  000020  005416            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2456                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2457  012116  042777  020000  005424            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2458  012124  052777  020000  005416            BIS     #CLK,@TXCSR     ;POKE CLK UP
2459  012132  016703  005402                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2460  012136  012700  000252                    MOV     #252,R0 ;EXPECTED
2461  012142  012767  000010  166770            MOV     #8.,SHIFT       ;# OF SHIFTS
2462  012150  012767  000252  166766            MOV     #252,TEMP1      ;DATA CHAR
2463  012156  004767  005056                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2464  012162  105777  005346                    TSTB    @RXCSR  ;RXDONE ?
2465  012166  100401                            BMI     64$
2466  012170  104000                            HLT     ;RXDONE SHOULD BE SET
2467  012172                            64$:
2468  012172  017701  005342                    MOV     @RXDBUF,R1      ;ACTUAL
2469  012176  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2470  012200  001401                            BEQ     65$
2471  012202  104002                            HLT     2               ;RECEIVED DATA DID NOT MATCH
2472                                                                    ;EXPECTED DATA - CHECK MAINT DATA
2473                                                                    ;OR RECEIVER LOGIC
2474  012204                            65$:
2475  012204  012767  000010  166726            MOV     #8.,SHIFT       ;# OF SHIFTS
2476  012212  012767  000252  166724            MOV     #252,TEMP1      ;DATA CHAR
2477  012220  004767  005014                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2478                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2479  012224  012767  000010  166706            MOV     #8.,SHIFT       ;# OF SHIFTS
2480  012232  012767  000252  166704            MOV     #252,TEMP1      ;DATA CHAR
2481  012240  004767  004774                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2482  012244  012700  140252                    MOV     #140000!252,R0  ;EXPECTED DATA PLUS
2483                                                                    ;RXERR & OVRRUN
2484  012250  017701  005264                    MOV     @RXDBUF,R1      ;ACTUAL
2485  012254  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
2486  012256  001401                            BEQ     66$
2487  012260  104002                            HLT     2               ;SPECIFICALLY LOOK AT RXERR &
2488                                                                    ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2489  012262                            66$:
2490  012262  104400                            SCOPE
2491                                            ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2492                                            ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
```

```
2493                                            ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2494                                            ;;(OVRRUN,RXERR)
2495                                            ;;MODE:SYNEXT
2496                                            ;;LENGTH:EIGHT
2497                                            ;;CHAR:377
2498                                            ;;
2499  012264  012767  000033  166634  TST27:    MOV     #27,TSTNO       ;SAVE THIS
2500  012272  012767  012512  166616            MOV     #TST28,NEXT             ;GO TO THIS TEST WHEN THRU
2501  012300  052777  000400  005242            BIS     #MRESET,@TXCSR  ;MASTER RESET
2502  012306  012777  020000  005230            MOV     #SYNEXT,@PARCSR ;SET THE MODE
2503  012314  052777  000400  005226            BIS     #MRESET,@TXCSR  ;MASTER RESET
2504
2505                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2506  012322  012777  064001  005220            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2507
2508                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2509  012330  012777  026000  005206            MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
2510  012336  052777  000020  005170            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2511                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2512  012344  042777  020000  005176            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2513  012352  052777  020000  005170            BIS     #CLK,@TXCSR     ;POKE CLK UP
2514  012360  016703  005154                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2515  012364  012700  000377                    MOV     #377,R0 ;EXPECTED
2516  012370  012767  000010  166542            MOV     #8.,SHIFT       ;# OF SHIFTS
2517  012376  012767  000377  166540            MOV     #377,TEMP1      ;DATA CHAR
2518  012404  004767  004630                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2519  012410  105777  005120                    TSTB    @RXCSR  ;RXDONE ?
2520  012414  100401                            BMI     64$
2521  012416  104000                            HLT     ;RXDONE SHOULD BE SET
2522  012420                            64$:
2523  012420  017701  005114                    MOV     @RXDBUF,R1      ;ACTUAL
2524  012424  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2525  012426  001401                            BEQ     65$
2526  012430  104002                            HLT     2               ;RECEIVED DATA DID NOT MATCH
2527                                                                    ;EXPECTED DATA - CHECK MAINT DATA
2528                                                                    ;OR RECEIVER LOGIC
2529  012432                            65$:
2530  012432  012767  000010  166500            MOV     #8.,SHIFT       ;# OF SHIFTS
2531  012440  012767  000377  166476            MOV     #377,TEMP1      ;DATA CHAR
2532  012446  004767  004566                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2533                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2534  012452  012767  000010  166460            MOV     #8.,SHIFT       ;# OF SHIFTS
2535  012460  012767  000377  166456            MOV     #377,TEMP1      ;DATA CHAR
2536  012466  004767  004546                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2537  012472  012700  140377                    MOV     #140000!377,R0  ;EXPECTED DATA PLUS
2538                                                                    ;RXERR & OVRRUN
2539  012476  017701  005036                    MOV     @RXDBUF,R1      ;ACTUAL
2540  012502  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
2541  012504  001401                            BEQ     66$
2542  012506  104002                            HLT     2               ;SPECIFICALLY LOOK AT RXERR &
2543                                                                    ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2544  012510                            66$:
2545  012510  104400                            SCOPE
2546                                            ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2547                                            ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2548                                            ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
```

```
2549                                            ;;(OVRRUN,RXERR)
2550                                            ;;MODE:SYNEXT
2551                                            ;;LENGTH:EIGHT
2552                                            ;;CHAR:0
2553                                            ;;
2554  012512  012767  000034  166406  TST28:    MOV     #28,TSTNO       ;SAVE THIS
2555  012520  012767  012740  166370            MOV     #.EOP,NEXT              ;GO TO THIS TEST WHEN THRU
2556  012526  052777  000400  005014            BIS     #MRESET,@TXCSR  ;MASTER RESET
2557  012534  012777  020000  005002            MOV     #SYNEXT,@PARCSR ;SET THE MODE
2558  012542  052777  000400  005000            BIS     #MRESET,@TXCSR  ;MASTER RESET
2559
2560                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2561  012550  012777  064001  004772            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2562
2563                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2564  012556  012777  026000  004760            MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
2565  012564  052777  000020  004742            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2566                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2567  012572  042777  020000  004750            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2568  012600  052777  020000  004742            BIS     #CLK,@TXCSR     ;POKE CLK UP
2569  012606  016703  004726                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2570  012612  012700  000000                    MOV     #0,R0   ;EXPECTED
2571  012616  012767  000010  166314            MOV     #8.,SHIFT       ;# OF SHIFTS
2572  012624  012767  000000  166312            MOV     #0,TEMP1        ;DATA CHAR
2573  012632  004767  004402                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2574  012636  105777  004672                    TSTB    @RXCSR  ;RXDONE ?
2575  012642  100401                            BMI     64$
2576  012644  104000                            HLT     ;RXDONE SHOULD BE SET
2577  012646                            64$:
2578  012646  017701  004666                    MOV     @RXDBUF,R1      ;ACTUAL
2579  012652  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2580  012654  001401                            BEQ     65$
2581  012656  104002                            HLT     2               ;RECEIVED DATA DID NOT MATCH
2582                                                                    ;EXPECTED DATA - CHECK MAINT DATA
2583                                                                    ;OR RECEIVER LOGIC
2584  012660                            65$:
2585  012660  012767  000010  166252            MOV     #8.,SHIFT       ;# OF SHIFTS
2586  012666  012767  000000  166250            MOV     #0,TEMP1        ;DATA CHAR
2587  012674  004767  004340                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2588                                    ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2589  012700  012767  000010  166232            MOV     #8.,SHIFT       ;# OF SHIFTS
2590  012706  012767  000000  166230            MOV     #0,TEMP1        ;DATA CHAR
2591  012714  004767  004320                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2592  012720  012700  140000                    MOV     #140000!0,R0    ;EXPECTED DATA PLUS
2593                                                                    ;RXERR & OVRRUN
2594  012724  017701  004610                    MOV     @RXDBUF,R1      ;ACTUAL
2595  012730  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
2596  012732  001401                            BEQ     66$
2597  012734  104002                            HLT     2               ;SPECIFICALLY LOOK AT RXERR &
2598                                                                    ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2599  012736                            66$:
2600  012736  104400                            SCOPE
```

```
2601
2602                                        ;END OF PASS
2603                                        ;TYPE NAME OF TEST
2604                                        ;UPDATE PASS COUNT
2605                                        ;CHECK FOR EXIT TO ACT-11
2606                                        ;RESTART TEST
2607
2608  012740  104402              .EOP:     TYPE                        ;TYPE NAME OF TEST
2609  012742  016102                        MEPASS
2610  012744  104410  013176              CONVRT  ,OUTCRY
2611  012750  104402  015623              TYPE    ,DEVICE
2612  012754  105767  166222              TSTB    MULTD                 ;ARE YOU RUNNING MULTIPLE DEVICES ?
2613  012760  001511                      BEQ     CCC                   ;NO,JUMP AROUND
2614  012762  005767  166230              TST     ACTREG                ;ARE ANY DEVICES ACTIVE ?
2615  012766  001007                      BNE     RUNIT                 ;YES
2616  012770  104402  015635              TYPE    ,MCOW                 ;NO
2617  012774  016700  166216              MOV     ACTREG,R0             ;DISPLAY ACTREG
2618  013000  000000                      HALT                          ;SELECT SOMETHING TO RUN @ ACTREG:
2619                                      ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2620  013002  000167  166252              JMP     .START                ;START OVER AGAIN.....YOU DESELECTED EVERYTHING
2621  013006  062767  000010  166170  RUNIT:  ADD     #10,BASEADD           ;NEXT BLOCK (ADDRESSES)
2622  013014  062767  000010  166170  ZERO:   ADD     #10,BASEIV            ;NEXT BLOCK (VECTORS)
2623  013022  000241                      CLC
2624  013024  006167  166170              ROL     ROTADD                ;UP DATE ROTATING POINTER
2625  013030  103410                      BCS     2$                    ;IS IT THE LAST DEVICE
2626                                                                    ;TO BE TESTED IN THIS PASS ?
2627  013032  036767  166162  166156      BIT     ROTADD,ACTREG         ;TEST THIS DEVICE FOR ACTIVE STATUS
2628  013040  001762                      BEQ     RUNIT                 ;IF NOT ACTIVE, TRY NEXT ADDRESS
2629  013042  004767  000034              JSR     PC,REPLAY             ;CALCULATE NEW PARAMETERS
2630  013046  000167  000174              JMP     RESTRT                ;YES IT WAS ACTIVE,TEST THIS DEVICE
2631  013052  012767  000001  166140  2$:       MOV     #1,ROTADD             ;OK!,NOW SET UP  ROTATING
2632                                                                    ;POINTER FOR NEXT MULTIPLE PASS
2633  013060  016767  166122  166116          MOV     KEEPADD,BASEADD       ;RESTORE BASE ADDRESS
2634  013066  016767  166122  166116          MOV     KEEPIV,BASEIV         ;RESTORE BASE INTERRUPT VECTORS
2635  013074  004767  000002              JSR     PC,REPLAY             ;CALC NEW PARAMETERS
2636  013100  000441                      BR      CCC                   ;JUMP AROUND REPLAY
2637  013102  016767  166076  004126  REPLAY: MOV     BASEADD,DUBASE        ;SET UP FOR NEW ADDRESSES
2638  013110  004767  003770              JSR     PC,DUADDR             ;CREATE NEW ADDRESSES
2639  013114  016767  166072  004436          MOV     BASEIV,DURIV          ;CREATE DURIV
2640  013122  062767  000002  166062          ADD     #2,BASEIV
2641  013130  016767  166056  004424          MOV     BASEIV,DURIS          ;CREATE DURIS
2642  013136  062767  000002  166046          ADD     #2,BASEIV
2643  013144  016767  166042  004412          MOV     BASEIV,DUTIV          ;CREATE DUTIV
2644  013152  062767  000002  166032          ADD     #2,BASEIV
2645  013160  016767  166026  004400          MOV     BASEIV,DUTIS          ;CREATE DUTIS
2646  013166  016767  004366  166016          MOV     DURIV,BASEIV          ;RESTORE
2647  013174  000207                      RTS     PC
2648
2649  013176  000001              OUTCRY: 1
2650  013200     006      002                .BYTE   6,2
2651  013202  017534                        RXCSR
2652
2653  013204                      CCC:
2654  013204  005067  165724              CLR     LSTERR                ;CLEAR LAST ERROR PC
2655  013210  005067  166010              CLR     ERRFLG                ;CLEAR ERROR FLAG
2656  013214  005267  165710              INC     PASCNT                ;UPDATE PASS COUNT
```

```
2657  013220  016777  165704  165654          MOV     PASCNT,@LIGHTS          ;DISPLAY PASS COUNT
2658  013226  013701  000042                  MOV     @#42,R1                 ;CHECK FOR ACT-11 OR DDP
2659  013232  001405                          BEQ     RESTRT                  ;IF NOT, CONTINUE TESTING
2660  013234  000005                          RESET
2661  013236  004711                  LOGICAL:        JSR     PC,(R1)
2662  013240  000240                          NOP
2663  013242  000240                          NOP
2664  013244  000240                          NOP
2665  013246  012767  000340  164522  RESTRT: MOV     #340,PS                 ;PREVENT INTERRUPTS (PRIO: 7)
2666  013254  104413                          CKSWR                           ;CHECK FOR ↑G
2667  013256  012767  002350  165630          MOV     #TST1,RTRN
2668  013264  000167  167060                  JMP     TST1
2669
2670                                          ;SCOPE LOOP AND INTERATION HANDLER
2671
2672  013270                          .SCOPE:
2673                                  ;**** START OF CODE FOR THE X OR TESTER ******
2674  013270  000424                          BR      4$                      ;IF RUNNING ON THE X OR TESTER CHANGE
2675                                                                          ;THIS INSTRUCTION TO A "NOP"(NOP=240)
2676  013272  013746  000004                  MOV     @#4,-(SP)               ;SAVE CONTENTS OF ERROR VECTOR
2677  013276  012737  013316  000004          MOV     #1$,@#4                 ;SET FOR TIME OUT
2678  013304  005737  177060                  TST     @#177060                ;TIME OUT ON  X OR ?
2679  013310  012637  000004                  MOV     (SP)+,@#4               ;RESTORE ERROR VECTOR
2680  013314  000404                          BR      2$                      ;GO TO NEXT TEST
2681  013316  022626                  1$:     CMP     (SP)+,(SP)+             ;CLEAR THE STACK AFTER A TIMEOUT
2682  013320  012637  000004                  MOV     (SP)+,@#4               ;RESTORE ERROR VECTOR
2683  013324  000403                          BR      3$                      ;LOOP ON PRESENT TEST
2684  013326  016767  165564  165560  2$:     MOV     NEXT,RTRN               ;SET UP NEXT TEST IN RTRN
2685  013334  016716  165554          3$:     MOV     RTRN,(SP)               ;SET UP STACK FOR RTI
2686  013340  000002                          RTI
2687  013342                          4$:     ;**** END OF CODE FOR THE  X OR TESTER ******
2688  013342  104413                          CKSWR                           ;CHECK FOR ↑G
2689  013344  032777  040000  165526          BIT     #SW14,@SWR              ;LOOP ON CURRENT TEST ?
2690  013352  001407                  TTST:   BEQ     1$
2691  013354  000432                          BR      3$
2692  013356  105777  165522                  TSTB    @TKCSR                  ;TEST TTY FLAG
2693  013362  100027                          BPL     3$
2694  013364  017700  165516                  MOV     @TKDBR,R0               ;CLR DONE BIT
2695  013370  000412                          BR      2$                      ;IF A TTY KEY IS STRUCK GO TO NEXT TST
2696  013372  032777  004000  165500  1$:     BIT     #SW11,@SWR              ;INHIBIT ITERATIONS ?
2697  013400  001006                          BNE     2$
2698  013402  005267  165516                  INC     LPCNT
2699  013406  026767  165512  165506          CMP     LPCNT,ICOUNT            ;CHECK FOR ITERATION CNT FINISH
2700  013414  101412                          BLOS    3$
2701  013416  105067  165602          2$:     CLRB    ERRFLG
2702  013422  005067  165476                  CLR     LPCNT
2703  013426  012767  000005  165466          MOV     #5,ICOUNT               ;SET UP ITERATION COUNT
2704  013434  016767  165456  165452          MOV     NEXT,RTRN               ;SET UP NEXT TEST IN RTRN
2705  013442  016716  165446          3$:     MOV     RTRN,(SP)               ;SET UP STACK FOR RTI
2706  013446  000002                          RTI
2707  013450  001407                  BRW:    1407                            ;RESTORE "BEQ 1$" INSTRUCTION
2708  013452  000432                  BRX:    432                             ;RESTORE "BR 3$" INSTRUCTION
2709
2710                                          ;CHECK FOR FREEZE ON CURRENT DATA
2711
2712  013454  104413                  .SCOP1: CKSWR                           ;CHECK  FOR ↑G
```

```
2713  013456  032777  001000  165414          BIT     #SW09,@SWR
2714  013464  001402                          BEQ     1$
2715  013466  016716  165426                  MOV     LOCK,(SP)
2716  013472  000002                  1$:     RTI
2717
2718                                          ;TELETYPE OUTPUT ROUTINE
2719
2720  013474  010546                  .TYPE:  MOV     R5,-(SP)
2721  013476  017605  000002                  MOV     @2(SP),R5
2722  013502  062766  000002  000002          ADD     #2,2(SP)
2723  013510  105715                  1$:     TSTB    (R5)                    ;LOOK FOR "0"
2724  013512  001406                          BEQ     3$
2725  013514  105777  165370          2$:     TSTB    @TPCSR                  ;TEST DONE BIT
2726  013520  100375                          BPL     2$
2727  013522  112577  165364                  MOVB    (R5)+,@TPDBR            ;TYPE CHAR
2728  013526  000770                          BR      1$                      ;DO IT AGAIN UNTIL "0" IS SEEN
2729  013530  012605                  3$:     MOV     (SP)+,R5
2730  013532  000002                          RTI
2731
2732                                          ;ASCII STRING INPUT ROUTINE
2733
2734  013534  010346                  .INSTR: MOV     R3,-(SP)
2735  013536  010446                          MOV     R4,-(SP)
2736  013540  017667  000004  000010          MOV     @4(SP),.MSG
2737  013546  062766  000002  000004          ADD     #2,4(SP)
2738  013554  104402                  .INST1: TYPE
2739  013556  000000                  .MSG:   0
2740  013560  012704  016670                  MOV     #INBUF,R4
2741  013564  012703  000007                  MOV     #7,R3
2742  013570  105777  165310          1$:     TSTB    @TKCSR
2743  013574  100375                          BPL     1$
2744  013576  117714  165304                  MOVB    @TKDBR,(R4)
2745  013602  142714  000200                  BICB    #200,(R4)
2746  013606  121427  000025                  CMPB    (R4),#25                ;IS IT <↑U>
2747  013612  001003                          BNE     200$
2748  013614  104402  016012                  TYPE,MCRLF
2749  013620  000755                          BR      .INST1
2750  013622  122427  000015          200$:   CMPB    (R4)+,#15
2751  013626  001423                          BEQ     INSTR2
2752  013630  117777  165252  165254          MOVB    @TKDBR,@TPDBR
2753  013636  105777  165246          2$:     TSTB    @TPCSR
2754  013642  100375                          BPL     2$
2755  013644  005303                          DEC     R3
2756  013646  001350                          BNE     1$
2757  013650  000402                          BR      .INSTG
2758  013652  010346                  .INSTE: MOV     R3,-(SP)
2759  013654  010446                          MOV     R4,-(SP)
2760  013656  104402                  .INSTG: TYPE
2761  013660  016006                          MQM
2762  013662  005737  015150                  TST     @#RDSW
2763  013666  001402                          BEQ     400$
2764  013670  104402  016012                  TYPE,MCRLF
2765  013674  000727                  400$:   BR      .INST1
2766  013676  012604                  INSTR2: MOV     (SP)+,R4
2767  013700  012603                          MOV     (SP)+,R3
2768  013702  000002                          RTI
```

```
2769
2770                                        ;CONVERT ASCII STRING TO OCTAL
2771
2772  013704  010546            .PARAM:  MOV     R5,-(SP)
2773  013706  010446                     MOV     R4,-(SP)
2774  013710  016605  000004            MOV     4(SP),R5
2775  013714  012567  000170            MOV     (R5)+,LOLIM
2776  013720  012567  000166            MOV     (R5)+,HILIM
2777  013724  012567  000164            MOV     (R5)+,DEVADR
2778  013730  112567  000162            MOVB    (R5)+,LOBITS
2779  013734  112567  000157            MOVB    (R5)+,ADRCNT
2780  013740  010566  000004            MOV     R5,4(SP)
2781  013744  005005            PARAM1:  CLR     R5
2782  013746  012704  016670            MOV     #INBUF,R4
2783  013752  122714  000015            CMPB    #15,(R4)
2784  013756  001420                     BEQ     PARERR
2785  013760  121427  000060     1$:    CMPB    (R4),#60
2786  013764  002415                     BLT     PARERR
2787  013766  121427  000067            CMPB    (R4),#67
2788  013772  003012                     BGT     PARERR
2789  013774  142714  000060            BICB    #60,(R4)
2790  014000  152405                     BISB    (R4)+,R5
2791  014002  122714  000015            CMPB    #15,(R4)
2792  014006  001414                     BEQ     LIMITS
2793  014010  006305                     ASL     R5
2794  014012  006305                     ASL     R5
2795  014014  006305                     ASL     R5
2796  014016  000760                     BR      1$
2797  014020  122714  000015    PARERR:  CMPB    #15,(R4)        ;IS FIRST CHARACTER A <CR>
2798  014024  001003                     BNE     120$
2799  014026  005737  015150            TST     @#RDSW          ;IS CKSWR ROUTINE BEING USED
2800  014032  001023                     BNE     PARTI
2801  014034  104404            120$:    INSTER
2802  014036  000742                     BR      PARAM1
2803
2804                                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2805
2806  014040  020567  000046    LIMITS:  CMP     R5,HILIM
2807  014044  101365                     BHI     PARERR
2808  014046  020567  000036            CMP     R5,LOLIM
2809  014052  103762                     BLO     PARERR
2810  014054  136705  000036            BITB    LOBITS,R5
2811  014060  001357                     BNE     PARERR
2812
2813                                        ;STORE NUMBER AT SPECIFIED ADDRESS
2814
2815  014062  016704  000026            MOV     DEVADR,R4
2816  014066  010524            1$:      MOV     R5,(R4)+
2817  014070  062705  000002            ADD     #2,R5
2818  014074  105367  000017            DECB    ADRCNT
2819  014100  001372                     BNE     1$
2820  014102  012604            PARTI:   MOV     (SP)+,R4
2821  014104  012605                     MOV     (SP)+,R5
2822  014106  000002                     RTI
2823  014110  000000            LOLIM:   0
2824  014112  000000            HILIM:   0
```

```
2825  014114  000000                  DEVADR: 0
2826  014116  000000                  LOBITS: 0
2827          014117                  ADRCNT=LOBITS+1
2828
2829                                          ;SAVE PC OF TEST THAT FAILED AND R0-R5
2830
2831  014120  016667  000004  165046  .SAV05: MOV     4(SP),SAVPC
2832
2833                                          ;SAVE R0-R5
2834
2835  014126  010567  165036          SV05:   MOV     R5,SAVR5
2836  014132  010467  165030                  MOV     R4,SAVR4
2837  014136  010367  165022                  MOV     R3,SAVR3
2838  014142  010267  165014                  MOV     R2,SAVR2
2839  014146  010167  165006                  MOV     R1,SAVR1
2840  014152  010067  165000                  MOV     R0,SAVR0
2841  014156  000002                          RTI
2842
2843                                          ;RESTORE R0-R5
2844
2845  014160  016700  164772          .RES05: MOV     SAVR0,R0
2846  014164  016701  164770                  MOV     SAVR1,R1
2847  014170  016702  164766                  MOV     SAVR2,R2
2848  014174  016703  164764                  MOV     SAVR3,R3
2849  014200  016704  164762                  MOV     SAVR4,R4
2850  014204  016705  164760                  MOV     SAVR5,R5
2851  014210  000002                          RTI
2852
2853                                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2854
2855  014212  104402                  .CONVR: TYPE
2856  014214  016012                          MCRLF
2857  014216  010046                  .CNVRT: MOV     R0,-(SP)
2858  014220  010146                          MOV     R1,-(SP)
2859  014222  010346                          MOV     R3,-(SP)
2860  014224  010446                          MOV     R4,-(SP)
2861  014226  010546                          MOV     R5,-(SP)
2862  014230  017601  000012                  MOV     @12(SP),R1
2863  014234  016767  002470  164706          MOV     TEMP,TEMP3
2864  014242  062766  000002  000012          ADD     #2,12(SP)
2865  014250  012167  000154                  MOV     (R1)+,WRDCNT
2866  014254  112167  000152          1$:     MOVB    (R1)+,CHRCNT
2867  014260  112167  000147                  MOVB    (R1)+,SPACNT
2868  014264  013167  000144                  MOV     @(R1)+,BINWRD
2869  014270  016704  000140          2$:     MOV     BINWRD,R4
2870  014274  116705  000132                  MOVB    CHRCNT,R5
2871  014300  012700  016730                  MOV     #TEMP,R0
2872  014304  010403                  3$:     MOV     R4,R3
2873  014306  042703  177770                  BIC     #177770,R3
2874  014312  062703  000060                  ADD     #060,R3
2875  014316  110320                          MOVB    R3,(R0)+
2876  014320  006204                          ASR     R4              ;SHIFT FOR NEXT #
2877  014322  042704  100000                  BIC     #100000,R4      ;CLUGE TO STOP BIT 15 PROPAGATING.
2878  014326  006204                          ASR     R4              ;DITTO
2879  014330  006204                          ASR     R4              ;DITTO
2880  014332  005305                          DEC     R5
```

```
2881   014334  001363                          BNE     3$
2882   014336  012703  016770                  MOV     #MDATA,R3
2883   014342  114023                  4$:     MOVB    -(R0),(R3)+
2884   014344  105367  000062                  DECB    CHRCNT
2885   014350  001374                          BNE     4$
2886   014352  105767  000055                  TSTB    SPACNT
2887   014356  001405                          BEQ     6$
2888   014360  112723  000040          5$:     MOVB    #040,(R3)+
2889   014364  105367  000043                  DECB    SPACNT
2890   014370  001373                          BNE     5$
2891   014372  105013                  6$:     CLRB    (R3)
2892   014374  104402                          TYPE
2893   014376  016770                          MDATA
2894   014400  005367  000024                  DEC     WRDCNT
2895   014404  001323                          BNE     1$
2896   014406  016767  164536  002314          MOV     TEMP3,TEMP
2897   014414  012605                          MOV     (SP)+,R5
2898   014416  012604                          MOV     (SP)+,R4
2899   014420  012603                          MOV     (SP)+,R3
2900   014422  012601                          MOV     (SP)+,R1
2901   014424  012600                          MOV     (SP)+,R0
2902   014426  000002                          RTI
2903   014430  000000          WRDCNT: 0
2904   014432  000000          CHRCNT: 0
2905           014433          SPACNT=CHRCNT+1
2906   014434  000000          BINWRD: 0
2907
2908                                  ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
2909                                  ;BUFFER TO THE CHARACTERS "N" AND "Y".
2910                                  ;IF THE CHARACTER IS "N" CLEAR THE FLAG
2911                                  ;IF THE CHARACTER IS "Y" SET THE FLAG
2912
2913   014436  017605  000000  .SETFLG:MOV     @(SP),R5
2914   014442  122767  000116  002220          CMPB    #'N,INBUF       ;IS IT "N" ?
2915   014450  001002                          BNE     1$
2916   014452  105015                          CLRB    (R5)    ;000
2917   014454  000406                          BR      2$
2918   014456  122767  000131  002204  1$:     CMPB    #'Y,INBUF               ;IS IT "Y" ?
2919   014464  001005                          BNE     3$
2920   014466  112715  177777                  MOVB    #-1,(R5)                ;377
2921   014472  062716  000002          2$:     ADD     #2,(SP)
2922   014476  000002                          RTI
2923   014500  104404          3$:     INSTER                          ;RETRY
2924   014502  000755                          BR      .SETFLG
2925                                  ;TRAP DISPATCH SERVICE
2926                                  ;ARGUMENT OF TRAP IS EXTRACTED
2927                                  ;AND USED AS OFFSET TO OBTAIN POINTER
2928                                  ;TO SELECTED SUBROUTINE
2929
2930   014504  011646          .TRPSR: MOV     (SP),-(SP)              ;GET PC OF RETURN
2931   014506  162716  000002                  SUB     #2,(SP)                 ;=PC OF TRAP
2932   014512  017616  000000                  MOV     @(SP),(SP)              ;GET TRP
2933   014516  006316          TRPOK:  ASL     (SP)                    ;MULTIPLY TRAP ARG BY 2
2934   014520  042716  177001                  BIC     #177001,(SP)            ;CLEAR UNWANTED BITS
2935   014524  062716  001226                  ADD     #.TRPTAB,(SP)           ;POINTER TO SUBROUTINE ADDRESS
2936   014530  017616  000000                  MOV     @(SP),(SP)              ;SUBROUTINE ADDRESS
```

```
2937  014534  000136                        JMP     @(SP)+          ;GO TO SUBROUTINE
2938
2939                                        ;ERROR HANDLER
2940
2941  014536  104413                .HLT:   CKSWR                   ;CHECK FOR ↑G
2942  014540  032777  020000  164332        BIT     #SW13,@SWR      ;INHIBIT ERROR TYPE OUT ?
2943  014546  001061                        BNE     HALTS
2944  014550  021667  164360                CMP     (SP),LSTERR
2945  014554  001404                        BEQ     1$
2946  014556  011667  164352                MOV     (SP),LSTERR
2947  014562  105067  164436                CLRB    ERRFLG
2948  014566  104406                1$:     SAV05
2949  014570  011605                        MOV     (SP),R5
2950  014572  162705  000002                SUB     #2,R5
2951  014576  011504                        MOV     (R5),R4
2952  014600  006304                        ASL     R4
2953  014602  061504                        ADD     (R5),R4
2954  014604  006304                        ASL     R4
2955  014606  042704  177001                BIC     #177001,R4
2956  014612  062704  017504                ADD     #.ERRTAB,R4
2957  014616  012467  000040                MOV     (R4)+,ERRMSG
2958  014622  012467  000046                MOV     (R4)+,DATAHD
2959  014626  011467  000054                MOV     (R4),DATABP
2960  014632  105767  164366                TSTB    ERRFLG
2961  014636  001403                        BEQ     TYPMSG
2962  014640  005767  000042                TST     DATABP
2963  014644  001014                        BNE     TYPDAT
2964  014646  104410            TYPMSG:     CONVRT
2965  014650  015000                        ERTABO
2966  014652  112767  177777  164344        MOVB    #-1,ERRFLG
2967  014660  104402                        TYPE
2968  014662  000000            ERRMSG:     0
2969  014664  005767  000004                TST     DATAHD
2970  014670  001402                        BEQ     TYPDAT
2971  014672  104402                        TYPE
2972  014674  000000            DATAHD:     0
2973  014676  005767  000004    TYPDAT:     TST     DATABP
2974  014702  001402                        BEQ     RESREG
2975  014704  104410                        CONVRT
2976  014706  000000            DATABP:     0
2977  014710  104407            RESREG:     RES05
2978  014712  005777  164162    HALTS:      TST     @SWR
2979  014716  100005                        BPL     EXITER
2980  014720  010046                        PUSHR0
2981  014722  016600  000002                MOV     2(SP),R0
2982  014726  000000                        HALT
2983  014730  012600                        POPR0
2984  014732  104413            EXITER:     CKSWR                   ;CHECK FOR ↑G
2985  014734  005267  164172                INC     ERRCNT
2986  014740  032777  000400  164132        BIT     #SW08,@SWR      ;LOOP ON ERROR ?
2987  014746  001007                        BNE     1$
2988  014750  032777  002000  164122        BIT     #SW10,@SWR      ;ESCAPE TO NEXT ON ERROR ?
2989  014756  001407                        BEQ     2$
2990  014760  016767  164132  164126        MOV     NEXT,RTRN       ;SET UP FOR NEXT TEST
2991  014766  012706  001100        1$:     MOV     #STACK,SP       ;REINITIALIZE SP
2992  014772  000177  164116                JMP     @RTRN
```

```
 2993  014776  000002                 2$:     RTI
 2994  015000  000001                 ERTABO: 1
 2995  015002     006     002                 .BYTE   6,2
 2996  015004  001174                         SAVPC
 2997                                         ;ENTER HERE ON POWER FAILURE
 2998
 2999
 3000  015006  010046                 .PFAIL: MOV     R0,-(SP)                ;SAVE R0-R5 ON PROCESSOR STACK
 3001  015010  010146                         MOV     R1,-(SP)
 3002  015012  010246                         MOV     R2,-(SP)
 3003  015014  010346                         MOV     R3,-(SP)
 3004  015016  010446                         MOV     R4,-(SP)
 3005  015020  010546                         MOV     R5,-(SP)
 3006  015022  016746  162776                 MOV     24,-(SP)
 3007  015026  010667  164140                 MOV     SP,SAVSP                ;SAVE STACK POINTER
 3008  015032  012767  015044  162764         MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
 3009  015040  000000                         HALT                            ;HALT ON POWER DOWN NORMAL
 3010  015042  000777                 1$:     BR      1$
 3011
 3012                                         ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
 3013
 3014  015044  016706  164122         RESTAR: MOV     SAVSP,SP                ;RESTORE STACK POINTER
 3015  015050  012605                         MOV     (SP)+,R5                ;RESTORE R0-R5
 3016  015052  012604                         MOV     (SP)+,R4
 3017  015054  012603                         MOV     (SP)+,R3
 3018  015056  012602                         MOV     (SP)+,R2
 3019  015060  012601                         MOV     (SP)+,R1
 3020  015062  012600                         MOV     (SP)+,R0
 3021  015064  012767  015006  162732         MOV     #.PFAIL,24              ;SET UP FOR POWER FAILURE
 3022  015072  012767  000340  162676         MOV     #340,PS
 3023  015100  012706  001100                 MOV     #STACK,SP
 3024  015104  005067  001620                 CLR     TEMP
 3025  015110  005267  001614         1$:     INC     TEMP
 3026  015114  001375                         BNE     1$
 3027  015116  104410                         CONVRT
 3028  015120  015142                         PFTAB
 3029  015122  104402                         TYPE
 3030  015124  016015                         MPFAIL
 3031  015126  005067  164072                 CLR     ERRFLG
 3032  015132  005067  163776                 CLR     LSTERR
 3033  015136  000177  163752                 JMP     @RTRN
 3034  015142  000001                 PFTAB:  1
 3035  015144     006     002                 .BYTE   6,2
 3036  015146  001114                         RTRN
 3037
 3038
 3039                                         ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ↑G TO ALLOW CHANGING
 3040                                         ;OF LOC.176.
 3041                                         ;LOCATIONS USED:
 3042  015150  000000                 RDSW:   .WORD   0
 3043
 3044
 3045  015152  005737  000042         .CKSWR: TST     @#42
 3046  015156  001042                         BNE     OUT
 3047  015160  022767  000176  163712         CMP     #SWREG,SWR              ;SOFTWARE SWITCH REGISTER PRESENT
 3048  015166  001036                         BNE     OUT                     ;NO, GET OUT
```

```
3049  015170  105777  163710                  TSTB    @TKCSR          ;YES,WAIT FOR
3050  015174  100033                          BPL     OUT             ;READY, GET CHARACTER
3051  015176  017767  163704  176352          MOV     @TKDBR,.MSG     ;AND STRIP OFF
3052  015204  042767  177600  176344          BIC     #177600,.MSG    ;THE GARBAGE
3053  015212  122767  000007  176336          CMPB    #7,.MSG         ;IS IT A <↑G>
3054  015220  001021                          BNE     OUT
3055  015222  104402  015300                  TYPE,$CNTG
3056  015226  005137  015150          .CNTLU: COM     @#RDSW
3057  015232  104402  015305                  TYPE,$MSWR
3058  015236  104411  015272                  CNVRT,SWREGC
3059  015242  104403  015315                  INSTR,$MNEW
3060  015246  104405                          PARAM
3061  015250  000000                          0
3062  015252  177777                          177777
3063  015254  000176                          SWREG
3064  015256      000     001          .BYTE           0,1
3065  015260  104402  016012                  TYPE,MCRLF
3066  015264  005037  015150          OUT:    CLR     @#RDSW
3067  015270  000002                          RTI
3068  015272  000001                  SWREGC: 1
3069  015274      006     002          .BYTE   6,2
3070  015276  000176                          SWREG
3071  015300  005015  043536      000 $CNTG:  .ASCIZ  <15><12>/↑G/
3072  015305      015  051412  051127 $MSWR:  .ASCIZ  <15><12>/SWR= /
3073  015312  020075      000
3074  015315      040  047040  053505 $MNEW:  .ASCIZ  /  NEW= /
3075  015322  020075      000
3076          015326                  .EVEN
3077  015326  005015  042012  030525 MTITLE: .ASCIZ  <15><12><12>/DU11 DZDUB-C TAPE B /<15><12>
3078  015334  020061  055104  052504
3079  015342  026502  020103  040524
3080  015350  042520  041040  006440
3081  015356  000012
3082  015360  005015  042526  052103 MVECTO: .ASCIZ  <15><12>/VECTOR ADDRESS-/
3083  015366  051117  040440  042104
3084  015374  042522  051523  000055
3085  015402  005015  051461  020124 MREGAD: .ASCIZ  <15><12>/1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-/
3086  015410  042504  044526  042503
3087  015416  020072  042522  042503
3088  015424  053111  051105  041440
3089  015432  047117  051124  046117
3090  015440  051040  043505  051511
3091  015446  042524  020122  042101
3092  015454  051104  051505  026523
3093  015462      000
3094  015463      015  040412  042522 MMULT:  .ASCIZ  <15><12>/ARE YOU RUNNING MULTIPLE DEVICES ? (Y OR N)-/
3095  015470  054440  052517  051040
3096  015476  047125  044516  043516
3097  015504  046440  046125  044524
3098  015512  046120  020105  042504
3099  015520  044526  042503  020123
3100  015526  020077  054450  047440
3101  015534  020122  024516  000055
3102  015542  005015  040514  052123 MLASTD: .ASCIZ  <15><12>/LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-/
3103  015550  042040  053105  041511
3104  015556  035105  042522  042503
```

```
3105  015564  053111  051105  041440
3106  015572  047117  051124  046117
3107  015600  051040  043505  051511
3108  015606  042524  020122  042101
3109  015614  051104  051505  026523
3110  015622     000
3111  015623     075  042504  044526  DEVICE: .ASCIZ  /=DEVICE  /
3112  015630  042503  020040     000
3113  015635     015  044012  053517  MCOW:   .ASCIZ  <15><12>/HOW NOW BROWN COW? ...SELECT SOMETHING TO RUN @ACTREG/
3114  015642  047040  053517  041040
3115  015650  047522  047127  041440
3116  015656  053517  020077  027056
3117  015664  051456  046105  041505
3118  015672  020124  047523  042515
3119  015700  044124  047111  020107
3120  015706  047524  051040  047125
3121  015714  040040  041501  051124
3122  015722  043505     000
3123  015725     015  047412  052125  MRANGE: .ASCIZ  <15><12>/OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-/
3124  015732  047440  020106  040522
3125  015740  043516  035105  042522
3126  015746  054524  042520  046040
3127  015754  051501  020124  042504
3128  015762  044526  042503  051040
3129  015770  041530  051123  040440
3130  015776  042104  042522  051523
3131  016004  000055
3132  016006  020040  000077          MQM:    .ASCIZ  /  ?/
3133  016012  005015     000          MCRLF:  .ASCIZ   <15><12>
3134  016015     040  050040  053517  MPFAIL: .ASCIZ  /  POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
3135  016022  051105  043040  044501
3136  016030  052514  042522  020054
3137  016036  051120  043517  040522
3138  016044  020115  042522  052123
3139  016052  051101  020124  052101
3140  016060  052040  051505  020124
3141  016066  047111  050040  047522
3142  016074  051107  051505  000123
3143  016102  005015  047105  020104  MEPASS: .ASCIZ  <15><12>/END OF PASS TAPE B/
3144  016110  043117  050040  051501
3145  016116  020123  040524  042520
3146  016124  041040     000
3147  016127     015  051012     000  MR:     .ASCIZ  <15><12>/R/
3148  016133     015  052012  051505  MTSTPC: .ASCIZ  <15><12>/TEST PC-/
3149  016140  020124  041520  000055
3150  016146  005015  047514  045503  MLOCK:  .ASCIZ  <15><12>/LOCK ON SELECTED TEST? (Y OR N)-/
3151  016154  047440  020116  042523
3152  016162  042514  052103  042105
3153  016170  052040  051505  037524
3154  016176  024040  020131  051117
3155  016204  047040  026451     000
3156  016211     015  042012  020125  MLEVEL: .ASCIZ  <15><12>/DU PRIORITY LEVEL-/
3157  016216  051120  047511  044522
3158  016224  054524  046040  053105
3159  016232  046105  000055
3160  016236  005015  020043  043117  MSYNC:  .ASCIZ  <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/
```

```
3161  016244  051440  047131  020103
3162  016252  044103  051101  020123
3163  016260  042523  042514  052103
3164  016266  042105  024040  030440
3165  016274  047440  020122  024462
3166  016302  000055
3167  016304  005015  051511  051440  MWIRE6: .ASCIZ  <15><12>/IS SEC XMIT JUMPER #6 IN? (Y OR N)-/
3168  016312  041505  054040  044515
3169  016320  020124  052512  050115
3170  016326  051105  021440  020066
3171  016334  047111  020077  054450
3172  016342  047440  020122  024516
3173  016350  000055
3174  016352  005015  051511  051440  MWIRE5: .ASCIZ  <15><12>/IS SEC REC JUMPER #5 IN? (Y OR N)-/
3175  016360  041505  051040  041505
3176  016366  045040  046525  042520
3177  016374  020122  032443  044440
3178  016402  037516  024040  020131
3179  016410  051117  047040  026451
3180  016416     000
3181  016417     015  044412  020123  MWIRE4: .ASCIZ  <15><12>/IS OPT CLR ENABLE JUMPER #4 IN? (Y OR N)-/
3182  016424  050117  020124  046103
3183  016432  020122  047105  041101
3184  016440  042514  045040  046525
3185  016446  042520  020122  032043
3186  016454  044440  037516  024040
3187  016462  020131  051117  047040
3188  016470  026451     000
3189  016473     015  040412  042522  MEXTJ:  .ASCII  <15><12>/ARE YOU RUNNING IN MAINT MODE EXTERNAL?/
3190  016500  054440  052517  051040
3191  016506  047125  044516  043516
3192  016514  044440  020116  040515
3193  016522  047111  020124  047515
3194  016530  042504  042440  052130
3195  016536  051105  040516  037514
3196  016544  005015  040401  042116          .ASCII  <15><12><1>/AND ..... DO YOU HAVE THE EXTERNAL MODEM BYPASS/
3197  016552  027040  027056  027056
3198  016560  042040  020117  047531
3199  016566  020125  040510  042526
3200  016574  052040  042510  042440
3201  016602  052130  051105  040516
3202  016610  020114  047515  042504
3203  016616  020115  054502  040520
3204  016624  051523
3205  016626  005015  045001  046525          .ASCIZ  <15><12><1>/JUMPER CONNECTOR ON ?(Y OR N)-/
3206  016634  042520  020122  047503
3207  016642  047116  041505  047524
3208  016650  020122  047117  037440
3209  016656  054450  047440  020122
3210  016664  024516  000055
3211                                  .EVEN
3212
3213                                  ;BUFFERS FOR INPUT-OUTPUT
3214
3215  016670  000040                  INBUF:  .BLKB   40
3216  016730  000040                  TEMP:   .BLKB   40
```

```
3217  016770  000040                  MDATA:  .BLKB   40
3218                                  ;*****************************************************************
3219                                          ;UTILITIES
3220                                  ;*****************************************************************
3221
3222                                  ;THIS UTILITY CALCULATES PRIORITY LEVEL
3223  017030  006367  000044          DULEV:  ASL     DUPRT   ;SHIFT LEFT
3224  017034  006367  000040                  ASL     DUPRT   ;
3225  017040  006367  000034                  ASL     DUPRT   ;
3226  017044  006367  000030                  ASL     DUPRT   ;
3227  017050  006367  000024                  ASL     DUPRT   ;
3228  017054  016767  000020  000020          MOV     DUPRT,LESS1     ;MOVE THIS TO LESS1
3229  017062  162767  000001  000012          SUB     #1,LESS1        ;CREATE LESS1
3230  017070  042767  000037  000004          BIC     #37,LESS1       ;CLEAR TNZVC
3231  017076  000207                          RTS     PC
3232  017100  000240                  DUPRT:  LEVEL5
3233  017102  000200                  LESS1:  LEVEL4  ;LEVEL TO ALLOW INTERRUPTS
3234
3235                                          ;NEW DU ADDRESSES
3236  017104  016767  000126  000422  DUADDR: MOV     DUBASE,RXCSR    ;XXX0
3237  017112  005267  000120                  INC     DUBASE
3238  017116  016767  000114  000412          MOV     DUBASE,HRXCSR   ;XXX1
3239  017124  005267  000106                  INC     DUBASE
3240  017130  016767  000102  000402          MOV     DUBASE,RXDBUF   ;XXX2
3241  017136  016767  000074  000400          MOV     DUBASE,PARCSR   ;XXX2
3242  017144  005267  000066                  INC     DUBASE
3243  017150  016767  000062  000364          MOV     DUBASE,HRXDBUF  ;XXX3
3244  017156  016767  000054  000362          MOV     DUBASE,HPARCSR  ;XXX3
3245  017164  005267  000046                  INC     DUBASE
3246  017170  016767  000042  000352          MOV     DUBASE,TXCSR    ;XXX4
3247  017176  005267  000034                  INC     DUBASE
3248  017202  016767  000030  000342          MOV     DUBASE,HTXCSR   ;XXX5
3249  017210  005267  000022                  INC     DUBASE
3250  017214  016767  000016  000332          MOV     DUBASE,TXDBUF   ;XXX6
3251  017222  005267  000010                  INC     DUBASE
3252  017226  016767  000004  000322          MOV     DUBASE,HTXDBUF  ;XXX7
3253  017234  000207                          RTS     PC
3254  017236  000000                  DUBASE: 0
3255
3256                                  ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3257                                  ;INFORMATION CONTAINED IN TEMP1 AND IT IS
3258                                  ;SHIFTED IN BY THE CONTENTS OF SHIFT
3259  017240  042777  040000  000302  RPOKE:  BIC     #MTDATA,@TXCSR
3260  017246  005067  161674                  CLR     TEMP2
3261  017252  006067  161666                  ROR     TEMP1   ;FORCE CARRY
3262  017256  006067  161664                  ROR     TEMP2   ;PICK UP CARRY IN BIT 15
3263  017262  006267  161660                  ASR     TEMP2   ;SHIFT INTO BIT 14
3264  017266  042767  100000  161652          BIC     #BIT15,TEMP2    ;CLR BIT 15
3265  017274  056777  161646  000246          BIS     TEMP2,@TXCSR    ;POKE MAINT DATA
3266  017302  042777  020000  000240          BIC     #CLK,@TXCSR     ;POKE CLK
3267  017310  052777  020000  000232          BIS     #CLK,@TXCSR     ;
3268  017316  005367  161616                  DEC     SHIFT
3269  017322  001346                          BNE     RPOKE
3270  017324  000207                          RTS     PC
3271                                  ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3272  017326  016767  161612  161612  ODD8:   MOV     TEMP1,TEMP2     ;SAVE TEMP1
```

```
3273  017334  005067  161610                  CLR     TEMP3
3274  017340  012727  000010                  MOV     #8.,(PC)+
3275  017344  000000                  4$:     0
3276  017346  006067  161574          1$:     ROR     TEMP2
3277  017352  005567  161572                  ADC     TEMP3
3278  017356  005367  177762                  DEC     4$
3279  017362  001371                          BNE     1$
3280  017364  006067  161560                  ROR     TEMP3
3281  017370  103404                          BCS     2$
3282  017372  052767  000400  161544          BIS     #BIT8,TEMP1     ;SET ODD PARITY
3283  017400  000403                          BR      3$
3284  017402  042767  000400  161534  2$:     BIC     #BIT8,TEMP1     ;CLR EVEN PARITY
3285                                          ;TEMP1 NOW HAS ODD PARITY CHARACTER
3286  017410  000207                  3$:     RTS     PC
3287
3288                                  ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3289  017412  016767  161526  161526  EVEN8:  MOV     TEMP1,TEMP2     ;SAVE TEMP1
3290  017420  005067  161524                  CLR     TEMP3
3291  017424  012727  000010                  MOV     #8.,(PC)+
3292  017430  000000                  4$:     0
3293  017432  006067  161510          1$:     ROR     TEMP2
3294  017436  005567  161506                  ADC     TEMP3
3295  017442  005367  177762                  DEC     4$
3296  017446  001371                          BNE     1$
3297  017450  006067  161474                  ROR     TEMP3
3298  017454  103004                          BCC     2$
3299  017456  052767  000400  161460          BIS     #BIT8,TEMP1     ;SET EVEN PARITY
3300  017464  000403                          BR      3$
3301  017466  042767  000400  161450  2$:     BIC     #BIT8,TEMP1     ;CLR ODD PARITY
3302                                          ;TEMP1 NOW HAS EVEN PARITY CHARACTER
3303  017474  000207                  3$:     RTS     PC
3304
3305  017476  062716  000002          TRPREG: ADD     #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3306                                          ;IN MAIN PART OF THE PROGRAM
3307  017502  000002                          RTI
3308                                  ;ERROR HLT TABLE
3309  017504  017570                  .ERRTAB:        EM0     ;HLT 0  BIT ERROR (GENERAL)
3310  017506  000000                          0
3311  017510  000000                          0
3312  017512  017604                          EM1     ;HLT 1  REGISTER ERROR
3313  017514  017755                          DH1
3314  017516  017776                          DT1
3315  017520  017646                          EM2     ;HLT 2  RECEIVER ERROR
3316  017522  017755                          DH1
3317  017524  017776                          DT1
3318  017526  017710                          EM3     ;HLT 3  TRANSMITTER ERROR
3319  017530  017755                          DH1
3320  017532  017776                          DT1
3321                                  ;DEFAULT DU ADDRESSES
3322  017534  160040                  RXCSR:  160040
3323  017536  160041                  HRXCSR: 160041
3324  017540  160042                  RXDBUF: 160042
3325  017542  160043                  HRXDBUF:        160043
3326  017544  160042                  PARCSR: 160042
3327  017546  160043                  HPARCSR:        160043
3328  017550  160044                  TXCSR:  160044
```

```
 3329  017552  160045                  HTXCSR: 160045
 3330  017554  160046                  TXDBUF: 160046
 3331  017556  160047                  HTXDBUF:        160047
 3332                                          ;DEFAULT DU VECTORS
 3333  017560  000770                  DURIV:  770     ;REC INTR VECTOR
 3334  017562  000772                  DURIS:  772     ;REC INTR STATUS
 3335  017564  000774                  DUTIV:  774     ;XMIT INTR VECTOR
 3336  017566  000776                  DUTIS:  776     ;XMIT INTR STATUS
 3337                                          ;ERROR MESSAGES
 3338  017570  036440  042440  051122  EM0:    .ASCIZ  / = ERROR PC/
 3339  017576  051117  050040  000103
 3340  017604  036440  051040  043505  EM1:    .ASCIZ  / = REGISTER ERROR PC/<15><12><1>/REGISTER  /
 3341  017612  051511  042524  020122
 3342  017620  051105  047522  020122
 3343  017626  041520  005015  051001
 3344  017634  043505  051511  042524
 3345  017642  020122  000040
 3346  017646  036440  051040  041505  EM2:    .ASCIZ  / = RECEIVER ERROR PC/<15><12><1>/REGISTER  /
 3347  017654  044505  042526  020122
 3348  017662  051105  047522  020122
 3349  017670  041520  005015  051001
 3350  017676  043505  051511  042524
 3351  017704  020122  000040
 3352  017710  036440  052040  040522  EM3:    .ASCIZ  / = TRANSMITTER ERROR PC/<15><12><1>/REGISTER  /
 3353  017716  051516  044515  052124
 3354  017724  051105  042440  051122
 3355  017732  051117  050040  006503
 3356  017740  000412  042522  044507
 3357  017746  052123  051105  020040
 3358  017754     000
 3359                                          ;DATA HEADERS FOR ERROR MESSAGES
 3360  017755     105  050130  041505  DH1:    .ASCIZ  /EXPECTED  ACTUAL/
 3361  017762  042524  020104  040440
 3362  017770  052103  040525  000114
 3363                                  .EVEN
 3364                                          ;DATA TABLES FOR ERROR MESSAGES
 3365  017776  000003                  DT1:    3
 3366  020000     006       004                .BYTE   6,4
 3367  020002  001164                          SAVR3   ;REGISTER
 3368  020004     006       004                .BYTE   6,4
 3369  020006  001156                          SAVR0   ;EXPECTED DATA
 3370  020010     006       002                .BYTE   6,2
 3371  020012  001160                          SAVR1   ;ACTUAL DATA
 3372          000001                  .END
```

```
AAA     002046           954#
ACTREG  001216           772#   902*   916*   917*   924*  2614   2617   2627
ADRCNT= 014117          2779*  2818*  2827#
BASEAD  001204           767#   884*   921*   922    928*   930*  2621*  2633*  2637
BASEIV  001212           770#   894*  2622*  2634*  2639   2640*  2641   2642*  2643   2644*  2645   2646*
BBB     001664           901    905#
BINWRD  014434          2868*  2869   2906#
BITW  = 002000           682#
BIT0  = 000001           630#   658    689
BIT1  = 000002           629#   657
BIT10 = 002000           620#   648    682
BIT11 = 004000           619#   647
BIT12 = 010000           618#   646    663
BIT13 = 020000           617#   645    662    681
BIT14 = 040000           616#   644    661    680
BIT15 = 100000           615#   643    660    679    3264
BIT2  = 000004           628#   656    997
BIT3  = 000010           627#   655    688
BIT4  = 000020           626#   654    687
BIT5  = 000040           625#   653    686
BIT6  = 000100           624#   652    685
BIT7  = 000200           623#   651    684
BIT8  = 000400           622#   650    666    683    3282   3284   3299   3301
BIT9  = 001000           621#   649    665
BREAK = 000001           689#  1040   1098   1156   1214   1272   1330   1388   1446   1504   1562   1620   1678
                        1736   1791   1846   1901   1956   2011   2066   2121   2176   2231   2286   2341   2396
                        2451   2506   2561
BRW     013450           994   1008   2707#
BRX     013452           995   1009   2708#
CARDET= 010000           646#
CCC     013204          2613   2636   2653#
CHRCNT  014432          2866*  2870   2884*  2904#  2905
CKSWR = 104413           814#  2666   2688   2712   2941   2984
CLK   = 020000           681#  1040   1046   1047   1049   1050   1098   1104   1105   1107   1108   1156   1162
                        1163   1165   1166   1214   1220   1221   1223   1224   1272   1278   1279   1281   1282
                        1330   1336   1337   1339   1340   1388   1394   1395   1397   1398   1446   1452   1453
                        1455   1456   1504   1510   1511   1513   1514   1562   1568   1569   1571   1572   1620
                        1626   1627   1629   1630   1678   1684   1685   1687   1688   1736   1742   1743   1791
                        1797   1798   1846   1852   1853   1901   1907   1908   1956   1962   1963   2011   2017
                        2018   2066   2072   2073   2121   2127   2128   2176   2182   2183   2231   2237   2238
                        2286   2292   2293   2341   2347   2348   2396   2402   2403   2451   2457   2458   2506
                        2512   2513   2561   2567   2568   3266   3267
CNTLU = 104414           816#   861
CNVRT = 104411           810#  3058
CONVRT= 104410           808#  2610   2964   2975   3027
COUNT   001142           743#
CTS   = 020000           645#
DATABP  014706          2959*  2962   2973   2976#
DATAHD  014674          2958*  2969   2972#
DEVADR  014114          2777*  2815   2825#
DEVICE  015623          2611   3111#
DH1     017755          3313   3316   3319   3360#
DISPRE  000174           710#   854
DLIGHT= 177570           600#   845
DNA   = 100000           679#
DNAINT= 000040           686#
```

```
DSC    = 100000  643#
DSINTE= 000040  653#
DSR    = 001000  649#
DSWR   = 177570  599#  844
DTR    = 000002  657#
DTI     017776  3314  3317  3320  3365#
DUADDR  017104  883  2638  3236#
DUBASE  017236  879  882  2637*  3236  3237*  3238  3239*  3240  3241  3242*  3243  3244  3245*
                3246  3247*  3248  3249*  3250  3251*  3252  3254#
DULEV   017030  949  3223#
DUPRT   017100  946  3223*  3224*  3225*  3226*  3227*  3228  3232#
DURIS   017562  2641*  3334#
DURIV   017560  890  893  894  2639*  2646  3333#
DUTIS   017566  2645*  3336#
DUTIV   017564  2643*  3335#
EIGHT  = 006000  674#  1507  1565  1623  1681  2399  2454  2509  2564
EM0     017570  3309  3338#
EM1     017604  3312  3340#
EM2     017646  3315  3346#
EM3     017710  3318  3352#
ERRCNT  001132  736#  835*  2985*
ERRFLG  001224  782#  834*  2655*  2701*  2947*  2960  2966*  3031*
ERRMSG  014662  2957*  2968#
ERTABO  015000  2965  2994#
EVENB   017412  3289#
EVEPAR= 001400  677#
EVPAR  = 000400  666#
EXITER  014732  2979  2984#
FIVE   = 000000  671#  1739  1794  1849  1904
FRMERR= 020000  662#
HALTS   014712  2943  2978#
HDXEN  = 000010  688#
HILIM   014112  2776*  2806  2824#
HOLD    001136  741#
HPARCS  017546  3244*  3327#
HRXCSR  017536  3238*  3323#
HRXDBU  017542  3243*  3325#
HTXCSR  017552  3248*  3329#
HTXDBU  017556  3252*  3331#
ICOUNT  001122  732#  2699  2703*
INBUF   016670  957  961  2740  2782  2914  2918  3215#
INIFLG  001222  780#  840  843*
INSTER= 104404  800#  965  2801  2923
INSTR  = 104403  798#  874  885  895  906  931  941  955  968  972  976  980  999
                1012  3059
INSTR2  013676  2751  2766#
ISYMOD= 000000  670#  1036  1043  1094  1101  1152  1159  1210  1217  1268  1275  1326  1333
                1384  1391  1442  1449  1500  1507  1558  1565  1616  1623  1674  1681
JMRBY   001203  763#  983
KEEPAD  001206  768#  882*  884  928  930  2633
KEEPIV  001214  771#  893*  2634
LASTAD  001210  769#  911  922  936
LESS1   017102  3228*  3229*  3230*  3233#
LEVEL0= 000000  640#
LEVEL1= 000040  639#
LEVEL2= 000100  638#
```

```
LEVEL3= 000140           637#
LEVEL4= 000200           636#  3233
LEVEL5= 000240           635#  3232
LEVEL6= 000300           634#
LEVEL7= 000340           633#
LIGHTS  001102           721#   845*   854*  2657*
LIMITS  014040          2792   2806#
LOBITS  014116          2778*  2810   2826#  2827
LOCK    001120           731#  2715
LOGICA  013236          2661#
LOKFLG  001225           783#  1002   1003
LOLIM   014110          2775*  2808   2823#
LPCNT   001124           733#   831*  2698*  2699   2702*
LSTERR  001134           737#   836*  2654*  2944   2946*  3032*
MCOW    015635          2616   3113#
MCRLF   016012          2748   2764   2856   3065   3133#
MDATA   016770          2882   2893   3217#
MEPASS  016102          2609   3143#
MEXT  = 010000           693#
MEXTJ   016473           981   3189#
MINT  = 004000           692#  1040   1098   1156   1214   1272   1330   1388   1446   1504   1562   1620   1678
                        1736   1791   1846   1901   1956   2011   2066   2121   2176   2231   2286   2341   2396
                        2451   2506   2561
MLASTD  015542           907   3102#
MLEVEL  016211           942   3156#
MLOCK   016146          1000   3150#
MMULT   015463           896   3094#
MPFAIL  016015          3030   3134#
MQM     016006          2761   3132#
MR      016127          1022   3147#
MRANGE  015725           932   3123#
MREGAD  015402           875   3085#
MRESET= 000400           683#  1035   1037   1093   1095   1151   1153   1209   1211   1267   1269   1325   1327
                        1383   1385   1441   1443   1499   1501   1557   1559   1615   1617   1673   1675   1731
                        1733   1786   1788   1841   1843   1896   1898   1951   1953   2006   2008   2061   2063
                        2116   2118   2171   2173   2226   2228   2281   2283   2336   2338   2391   2393   2446
                        2448   2501   2503   2556   2558
MSYNC   016236           956   3160#
MTDATA= 040000           680#  1040   1098   1156   1214   1272   1330   1388   1446   1504   1562   1620   1678
                        1736   1791   1846   1901   1956   2011   2066   2121   2176   2231   2286   2341   2396
                        2451   2506   2561   3259
MTITLE  015326           842   3077#
MTSTPC  016133          1013   3148#
MULTD   001202           762#   898    899    991   2612
MVECTO  015360           886   3082#
MWIRE4  016417           977   3181#
MWIRE5  016352           973   3174#
MWIRE6  016304           969   3167#
NEXT    001116           730#  1034*  1092*  1150*  1208*  1266*  1324*  1382*  1440*  1498*  1556*  1614*  1672*
                        1730*  1785*  1840*  1895*  1950*  2005*  2060*  2115*  2170*  2225*  2280*  2335*  2390*
                        2445*  2500*  2555*  2684   2704   2990
NOPAR = 000000           675#  1043   1101   1159   1217   1275   1333   1391   1449   1507   1565   1623   1681
                        1739   1794   1849   1904   1959   2014   2069   2124   2179   2234   2289   2344   2399
                        2454   2509   2564
ODDPAR= 001000           676#
ODDB    017326          3272#
```

```
ONCE    001362            841    844#
OPTCLR  001201            761#   979
OUT     015264           3046   3048   3050   3054   3066#
OUTCRY  013176           2610   2649#
OUTMUL  002024            904    929    940#
OVRRUN= 040000            661#
PARAM = 104405            802#   876    887    908    933    943   1014   3060
PARAM1  013744           2781#  2802
PARCSR  017544           1036*  1043*  1094*  1101*  1152*  1159*  1210*  1217*  1268*  1275*  1326*  1333*  1384*
                         1391*  1442*  1449*  1500*  1507*  1558*  1565*  1616*  1623*  1674*  1681*  1732*  1739*
                         1787*  1794*  1842*  1849*  1897*  1904*  1952*  1959*  2007*  2014*  2062*  2069*  2117*
                         2124*  2172*  2179*  2227*  2234*  2282*  2289*  2337*  2344*  2392*  2399*  2447*  2454*
                         2502*  2509*  2557*  2564*  3241*  3326#
PAREN = 001000            665#
PARER = 010000            663#
PARERR  014020           2784   2786   2788   2797#  2807   2809   2811
PARTI   014102           2800   2820#
PASCNT  001130            735#   833*   2656*  2657
PFTAB   015142           3028   3034#
POPRO = 012600            609#  2983
POP1SP= 005726            607#
POP2SP= 022626            611#
PS    = 177776            601#   828*   987*   2665*  3022*
PUSHRO= 010046            608#  2980
PUSH1S= 005746            606#
PUSH2S= 024646            610#
RDSW    015150           2762   2799   3042#  3056*  3066*
RECACT= 004000            647#
REPLAY  013102           2629   2635   2637#
RESREG  014710           2974   2977#
RESTAR  015044           3008   3014#
RESTRT  013246           2630   2659   2665#
RESOS = 104407            806#  2977
RING  = 040000            644#
RINTEN= 000100            652#
ROTADD  001220            775#   903*   915*   917    919*   924    927*  2624*  2627   2631*
RPOKE   017240           1055   1069   1073   1113   1127   1131   1171   1185   1189   1229   1243   1247   1287
                         1301   1305   1345   1359   1363   1403   1417   1421   1461   1475   1479   1519   1533
                         1537   1577   1591   1595   1635   1649   1653   1693   1707   1711   1748   1762   1766
                         1803   1817   1821   1858   1872   1876   1913   1927   1931   1968   1982   1986   2023
                         2037   2041   2078   2092   2096   2133   2147   2151   2188   2202   2206   2243   2257
                         2261   2298   2312   2316   2353   2367   2371   2408   2422   2426   2463   2477   2481
                         2518   2532   2536   2573   2587   2591   3259#  3269
RTRN    001114            729#   838*   1017   1021*  1023   2667*  2684*  2685   2704*  2705   2990*  2992   3033
                         3036
RTS   = 000004            656#
RUNA  = ****** U            1     33     45    108    546
RUNB  = ****** U            1     33     45    108    546
RUNC  = ****** U            1     33     45    108    546
RUND  = ****** U            1     33     45    108    546
RUNE  = ****** U            1     33     45    108    546
RUNF  = ****** U            1     33     45    108    546
RUNIT   013006           2615   2621#  2628
RXCSR   017534           1044*  1056   1102*  1114   1160*  1172   1218*  1230   1276*  1288   1334*  1346   1392*
                         1404   1450*  1462   1508*  1520   1566*  1578   1624*  1636   1682*  1694   1740*  1749
                         1795*  1804   1850*  1859   1905*  1914   1960*  1969   2015*  2024   2070*  2079   2125*
```

```
                        2134    2180*   2189    2235*   2244    2290*   2299    2345*   2354    2400*   2409    2455*   2464
                        2510*   2519    2565*   2574    2651    3236*   3322#
RXDBUF  017540          1051    1060    1076    1109    1118    1134    1167    1176    1192    1225    1234    1250    1283
                        1292    1308    1341    1350    1366    1399    1408    1424    1457    1466    1482    1515    1524
                        1540    1573    1582    1598    1631    1640    1656    1689    1698    1714    1744    1753    1769
                        1799    1808    1824    1854    1863    1879    1909    1918    1934    1964    1973    1989    2019
                        2028    2044    2074    2083    2099    2129    2138    2154    2184    2193    2209    2239    2248
                        2264    2294    2303    2319    2349    2358    2374    2404    2413    2429    2459    2468    2484
                        2514    2523    2539    2569    2578    2594    3240*   3324#
RXDONE= 000200           651#
RXERR = 100000           660#
SAVPC   001174           756#    2831*   2996
SAVR0   001156           749#    2840*   2845    3369
SAVR1   001160           750#    2839*   2846    3371
SAVR2   001162           751#    2838*   2847
SAVR3   001164           752#    2837*   2848    3367
SAVR4   001166           753#    2836*   2849
SAVR5   001170           754#    2835*   2850
SAVSP   001172           755#    3007*   3014
SAVUS = 104406           804#    2948
SCOPE = 104400           792#    1082    1140    1198    1256    1314    1372    1430    1488    1546    1604    1662    1720
                        1775    1830    1885    1940    1995    2050    2105    2160    2215    2270    2325    2380    2435
                        2490    2545    2600
SCOP1 = 104401           794#
SEND  = 000020           687#
SEREC   001200           760#    975
SETFLG= 104412           812#    897     970     974     978     982     1001
SEVEN = 004000           673#    1275    1333    1391    1449    2179    2234    2289    2344
SEXMIT  001177           759#    971
SHIFT   001140           742#    1053*   1067*   1071*   1111*   1125*   1129*   1169*   1183*   1187*   1227*   1241*   1245*
                        1285*   1299*   1303*   1343*   1357*   1361*   1401*   1415*   1419*   1459*   1473*   1477*   1517*
                        1531*   1535*   1575*   1589*   1593*   1633*   1647*   1651*   1691*   1705*   1709*   1746*   1760*
                        1764*   1801*   1815*   1819*   1856*   1870*   1874*   1911*   1925*   1929*   1966*   1980*   1984*
                        2021*   2035*   2039*   2076*   2090*   2094*   2131*   2145*   2149*   2186*   2200*   2204*   2241*
                        2255*   2259*   2296*   2310*   2314*   2351*   2365*   2369*   2406*   2420*   2424*   2461*   2475*
                        2479*   2516*   2530*   2534*   2571*   2585*   2589*   3268*
SIX   = 002000           672#    1043    1101    1159    1217    1959    2014    2069    2124
SPACNT= 014433           2867*   2886    2889*   2905#
SRD   = 002000           648#
STACK = 001100           602#    829     988     2991    3023
STD   = 000010           655#
STFLG   001223           781#    832*
STPSYN= 000400           650#
SVOS    014126           2835#
SWR     001100           720#    844*    849     853*    859     862     997     1010    2689    2696    2713    2942    2978
                        2986    2988    3047
SWREG   000176           711#    853     859     3047    3063    3070
SWREGC  015272           3058    3068#
SW00  = 000001           582#    862
SW01  = 000002           581#    1010
SW02  = 000004           580#
SW03  = 000010           579#
SW04  = 000020           578#
SW05  = 000040           577#
SW06  = 000100           576#
SW08  = 000400           575#    2986
```

E06


```
SW09  = 001000          574#  2713
SW10  = 002000          573#  2988
SW11  = 004000          572#  2696
SW12  = 010000          571#
SW13  = 020000          570#  2942
SW14  = 040000          569#  2689
SW15  = 100000          568#
SYNCNO  001176          758#   959*   963*
SYNEXT= 020000          669#  1732   1739   1787   1794   1842   1849   1897   1904   1952   1959   2007   2014
                       2062  2069   2117   2124   2172   2179   2227   2234   2282   2289   2337   2344   2392
                       2399  2447   2454   2502   2509   2557   2564
SYNINT= 030000          668#
SYNSCH= 000020          654#  1044   1102   1160   1218   1276   1334   1392   1450   1508   1566   1624   1682
                       1740  1795   1850   1905   1960   2015   2070   2125   2180   2235   2290   2345   2400
                       2455  2510   2565
SYSTST= 014000          694#
TEMP    016730         2863  2871   2896*  3024*  3025*  3216#
TEMP1   001144          744#  1054*  1068*  1072*  1112*  1126*  1130*  1170*  1184*  1188*  1228*  1242*  1246*
                       1286*  1300*  1304*  1344*  1358*  1362*  1402*  1416*  1420*  1460*  1474*  1478*  1518*
                       1532*  1536*  1576*  1590*  1594*  1634*  1648*  1652*  1692*  1706*  1710*  1747*  1761*
                       1765*  1802*  1816*  1820*  1857*  1871*  1875*  1912*  1926*  1930*  1967*  1981*  1985*
                       2022*  2036*  2040*  2077*  2091*  2095*  2132*  2146*  2150*  2187*  2201*  2205*  2242*
                       2256*  2260*  2297*  2311*  2315*  2352*  2366*  2370*  2407*  2421*  2425*  2462*  2476*
                       2480*  2517*  2531*  2535*  2572*  2586*  2590*  3261*  3272    3282*  3284*  3289    3299*
                       3301*
TEMP2   001146          745#  3260*  3262*  3263*  3264*  3265   3272*  3276*  3289*  3293*
TEMP3   001150          746#  2863*  2896   3273*  3277*  3280*  3290*  3294*  3297*
TEMP4   001152          747#
TEMP5   001154          748#
TKCSR   001104          722#  2692   2742   3049
TKDBR   001106          723#  2694   2744   2752   3051
TLAST = 012512         1016   2602#
TPCSR   001110          724#  2725   2753
TPDBR   001112          725#  2727*  2752*
TRPOK   014516         2933#
TRPREG  017476         3305#
TSTNO   001126          734#   837*  1033*  1091*  1149*  1207*  1265*  1323*  1381*  1439*  1497*  1555*  1613*
                       1671*  1729*  1784*  1839*  1894*  1949*  2004*  2059*  2114*  2169*  2224*  2279*  2334*
                       2389*  2444*  2499*  2554*
TST1    002350         1015   1021   1033#  2667   2668
TST10   005232         1498   1555#
TST11   005474         1556   1613#
TST12   005736         1614   1671#
TST13   006200         1672   1729#
TST14   006426         1730   1784#
TST15   006654         1785   1839#
TST16   007102         1840   1894#
TST17   007330         1895   1949#
TST18   007556         1950   2004#
TST19   010004         2005   2059#
TST2    002612         1034   1091#
TST20   010232         2060   2114#
TST21   010460         2115   2169#
TST22   010706         2170   2224#
TST23   011134         2225   2279#
TST24   011362         2280   2334#
```

```
TST25   011610          2335    2389#
TST26   012036          2390    2444#
TST27   012264          2445    2499#
TST28   012512          2500    2554#   2602
TST29 = ****** U        2555
TST3    003054          1092    1149#
TST4    003316          1150    1207#
TST5    003560          1208    1265#
TST6    004022          1266    1323#
TST7    004264          1324    1381#
TST8    004526          1382    1439#
TST9    004770          1440    1497#
TTST    013352           994*    995*   1005*   1006*   1008*   1009*   2690#
TXCSR   017550          1035*   1037*   1040*   1046*   1047*   1049*   1050*   1093*   1095*   1098*   1104*   1105*   1107*
                        1108*   1151*   1153*   1156*   1162*   1163*   1165*   1166*   1209*   1211*   1214*   1220*   1221*
                        1223*   1224*   1267*   1269*   1272*   1278*   1279*   1281*   1282*   1325*   1327*   1330*   1336*
                        1337*   1339*   1340*   1383*   1385*   1388*   1394*   1395*   1397*   1398*   1441*   1443*   1446*
                        1452*   1453*   1455*   1456*   1499*   1501*   1504*   1510*   1511*   1513*   1514*   1557*   1559*
                        1562*   1568*   1569*   1571*   1572*   1615*   1617*   1620*   1626*   1627*   1629*   1630*   1673*
                        1675*   1678*   1684*   1685*   1687*   1688*   1731*   1733*   1736*   1742*   1743*   1786*   1788*
                        1791*   1797*   1798*   1841*   1843*   1846*   1852*   1853*   1896*   1898*   1901*   1907*   1908*
                        1951*   1953*   1956*   1962*   1963*   2006*   2008*   2011*   2017*   2018*   2061*   2063*   2066*
                        2072*   2073*   2116*   2118*   2121*   2127*   2128*   2171*   2173*   2176*   2182*   2183*   2226*
                        2228*   2231*   2237*   2238*   2281*   2283*   2286*   2292*   2293*   2336*   2338*   2341*   2347*
                        2348*   2391*   2393*   2396*   2402*   2403*   2446*   2448*   2451*   2457*   2458*   2501*   2503*
                        2506*   2512*   2513*   2556*   2558*   2561*   2567*   2568*   3246*   3259*   3265*   3266*   3267*
                        3328#
TXDBUF  017554          3250*   3330#
TXDONE= 000200           684#
TXINTE= 000100           685#
TYPDAT  014676          2963    2970    2973#
TYPE  = 104402           796#    842    1022    2608    2611    2616    2738    2748    2760    2764    2855    2892    2967
                        2971    3029    3055    3057    3065
TYPMSG  014646          2961    2964#
USER  = 000000           691#
VOID  = 000001           658#
WRDCNT  014430          2865*   2894*   2903#
ZERO    013014          2622#
$CNTG   015300          3055    3071#
$E    = 000036           552#   1034    1035#   1092    1093#   1150    1151#   1208    1209#   1266    1267#   1324    1325#
                        1382    1383#   1440    1441#   1498    1499#   1556    1557#   1614    1615#   1672    1673#   1730
                        1731#   1785    1786#   1840    1841#   1895    1896#   1950    1951#   2005    2006#   2060    2061#
                        2115    2116#   2170    2171#   2225    2226#   2280    2281#   2335    2336#   2390    2391#   2445
                        2446#   2500    2501#   2555    2556#
$MNEW   015315          3059    3074#
$MSWR   015305          3057    3072#
$N    = 000034           552#   1033    1035#   1091    1093#   1149    1151#   1207    1209#   1265    1267#   1323    1325#
                        1381    1383#   1439    1441#   1497    1499#   1555    1557#   1613    1615#   1671    1673#   1729
                        1731#   1784    1786#   1839    1841#   1894    1896#   1949    1951#   2004    2006#   2059    2061#
                        2114    2116#   2169    2171#   2224    2226#   2279    2281#   2334    2336#   2389    2391#   2444
                        2446#   2499    2501#   2554    2556#   2602#
$Y    = 000015           784#    792     794#    796#    798#    800#    802#    804#    806#    808#    810#    812#    814#
                         816#    818#
.     = 020014           696#    699#    709#    716#   3076#   3215#   3216#   3217#
.BEGIN  002156           864     987#
.CKSWR  015152           815    3045#
```

```
.CNTLU  015226            817    3056#
.CNVRT  014216            811    2857#
.CONVR  014212            809    2855#
.EOP    012740           2555    2608#
.ERRTA  017504           2956    3309#
.HLT    014536            702    2941#
.INSTE  013652            801    2758#
.INSTG  013656           2757    2760#
.INSTR  013534            799    2734#
.INSTI  013554           2738#   2749    2765
.MSG    013556           2736*   2739#   3051*   3052*   3053
.PARAM  013704            803    2772#
.PFAIL  015006            700     830    3000#   3021
.RESOS  014160            807    2845#
.SAVOS  014120            805    2831#
.SCOPE  013270            793    2672#
.SCOPI  013454            795    2712#
.SETFL  014436            813    2913#   2924
.START  001260            712     828#    838    2620
.TRPSR  014504            704    2930#
.TRPTA  001226            789#   2935
.TYPE   013474            797    2720#
```

```
HLT      612#  1058  1063  1079  1116  1121  1137  1174  1179  1195  1232  1237  1253  1290  1295
         1311  1348  1353  1369  1406  1411  1427  1464  1469  1485  1522  1527  1543  1580  1585
         1601  1638  1643  1659  1696  1701  1717  1751  1756  1772  1806  1811  1827  1861  1866
         1882  1916  1921  1937  1971  1976  1992  2026  2031  2047  2081  2086  2102  2136  2141
         2157  2191  2196  2212  2246  2251  2267  2301  2306  2322  2356  2361  2377  2411  2416
         2432  2466  2471  2487  2521  2526  2542  2576  2581  2597
PRGEND   552#  2601
PRGFRT   552#   553
PUSSYF   552#
RSETUP   552#  1035  1093  1151  1209  1267  1325  1383  1441  1499  1557  1615  1673  1731  1786
         1841  1896  1951  2006  2061  2116  2171  2226  2281  2336  2391  2446  2501  2556
TSETUP   552#
$BEGIN   552#   984
$BINAR   552#
$BUFFE   552#  3212
$CABLE   552#
$CATCH   552#   695
$CLRVE   552#   844
$CONVR   552#  2852
$DNA     552#
$EOP     552#  2601
$GETFL   552#   895   968   972   976   980   999
$GETPA   552#   874   885   905   931   940  1012
$GETSY   552#   950
$HEADE   552#   553
$HLT     552#  2938
$INSTR   552#  2731
$ISOB    552#
$MATCH   552#
$MRR     552#
$MRRW    552#
$MRW     552#
$MSG     552#  3077
$PARAM   552#  2769
$PFAIL   552#  2997
$POKE    552#
$POKER   552#  1048  1106  1164  1222  1280  1338  1396  1454  1512  1570  1628  1686  1741  1796
         1851  1906  1961  2016  2071  2126  2181  2236  2291  2346  2401  2456  2511  2566
$RCNET   552#
$RECAC   552#
$REG     552#  2828
$RESET   552#  1035  1037  1093  1095  1151  1153  1209  1211  1267  1269  1325  1327  1383  1385
         1441  1443  1499  1501  1557  1559  1615  1617  1673  1675  1731  1733  1786  1788  1841
         1843  1896  1898  1951  1953  2006  2008  2061  2063  2116  2118  2171  2173  2226  2228
         2281  2283  2336  2338  2391  2393  2446  2448  2501  2503  2556  2558
$RXACT   552#
$SCOPE   552#  2669
$SCOPI   552#  2709
$SETFL   552#  2907
$SETVE   552#   696
$START   552#   820
$STRIP   552#
$SYMBO   552#   565
$SYNCR   552#  1044  1102  1160  1218  1276  1334  1392  1450  1508  1566  1624  1682
$TRAPS   552#   784
$TRPAR   552#
```

```
DZDUB-C MACY11 27(1006)  01-OCT-76  09:17  PAGE 78
DZDUBC.P11    21-MAY-76 00:00          CROSS REFERENCE TABLE -- MACRO NAMES                                    SEQ 0073

$TRPDE    552#    792    794    796    798    800    802    804    806    808    810    812    814    816
$TRPSR    552#   2925
$TSTNO    552#   1033   1091   1149   1207   1265   1323   1381   1439   1497   1555   1613   1671   1729   1784
         1839    1894   1949   2004   2059   2114   2169   2224   2279   2334   2389   2444   2499   2554
$TYPE     552#   2717
$UNIBU    552#
$VARIA    552#    715
$WORDF    552#
$WORDO    552#   1025   1083   1141   1199   1257   1315   1373   1431   1489   1547   1605   1663   1721   1776
         1831    1886   1941   1996   2051   2106   2161   2216   2271   2326   2381   2436   2491   2546
$WORDP    552#

. ABS.  020014     000


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZDUBC,DZDUBCV/CRF/SOL=HELLO.P11,PARA.P11,KEET.P11,DZDUBC.P11
 RUN-TIME: 23 34 3 SECONDS
 RUN-TIME RATIO: 228/60=3.7
 CORE USED:  18K  (35 PAGES)
```

JOB

Spooler runtime 10 Seconds, 47 KCS, 308 disk reads, 3 disk writes, 73 pages

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]