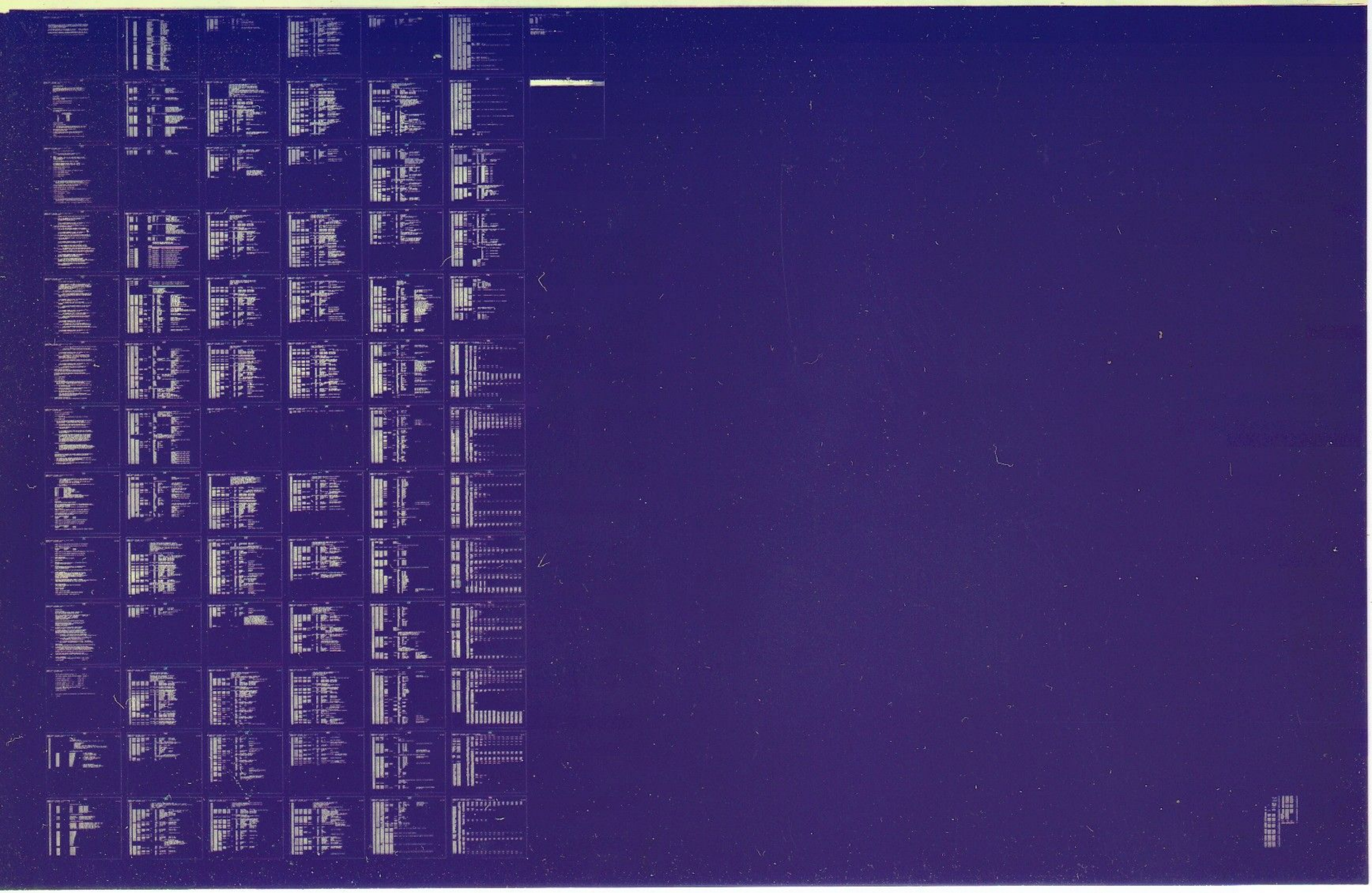
DU11

TIMING&INTERFACE
MD-11-DZDUE-C

EP-DZDUE-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 SW01=1
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
,,,IT WILL NOT WORK IF MULTIPLE DEVICES ARE SELECTED

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

4.3.4.1 LOAD 000200

4.3.4.2 SET SW01=1

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

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

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

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

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

4.3.5 PROGRAM RESTART WITH SW02 =1
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED

SEE NOTE IN 4.3.4 FOR MORE DETAILS

4.3.5.1 LOAD ADDRESS 000200

4.3.5.2 SET SWD2 =1
NOTE: IT MAY BE ADVANTAGEOUS TO SET SWD1=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 SWD1 =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 DZDUE-C TAPE E
                                ;COPYRIGHT 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                        ;STARTING PROCEDURE
                                        ;LOAD PROGRAM
                                        ;PRESS START
                                        ;PROGRAM WILL TYPE "DU11 DZDUE-C TAPE E "
                                        ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                                        ;AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE E"
                                        ;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                          DMAINTE=BIT5            ;DNA INTR ENAB
000020                          SEND=BIT4               ;SEND
000010                          MDXEN=BIT3              ;MDX/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  014422                          .PFAIL          ;POWER FAIL HANDLER
000026  000340                          340             ;SERVICE AT LEVEL 7
000030  014152                          .HLT            ;ERROR HANDLER
000032  000340                          340             ;SERVICE AT LEVEL 7
000034  014120                          .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  012704      .SCOPE
            104401          SCOP1=TRAP+1    ;CALL TO LOOP ON CURRENT DATA HANDLER
001230  013070      .SCOP1
            104402          TYPE=TRAP+2     ;CALL TO TELETYPE OUTPUT ROUTINE
001232  013110      .TYPE
            104403          INSTR=TRAP+3    ;CALL TO ASCII STRING INPUT ROUTINE
001234  013150      .INSTR
            104404          INSTER=TRAP+4   ;CALL TO INPUT ERROR HANDLER
001236  013266      .INSTER
            104405          PARAM=TRAP+5    ;CALL TO NUMERICAL DATA INPUT ROUTINE
001240  013320      .PARAM
            104406          SAVOS=TRAP+6    ;CALL TO REGISTER SAVE ROUTINE
001242  013534      .SAVOS
            104407          RESOS=TRAP+7    ;CALL TO REGISTER RESTORE ROUTINE
001244  013574      .RESOS
            104410          CONVRT=TRAP+10  ;CALL TO DATA OUTPUT ROUTINE
001246  013626      .CONVRT
            104411          CNVRT=TRAP+11   ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF
001250  013632      .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  014422  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  014742          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  015016                          MREGAD                          ;MESSAGE
 876  001546  104405                          PARAM                           ;CONVERT STRING
 877  001550  160000                          160000                          ;LOW LIMIT
 878  001552  167776                          167776                          ;HIGH LIMIT
 879  001554  016652                          DUBASE                          ;STORE AT THIS LOCATION
 880  001556     001                  .BYTE   1                               ;MASK
 881  001557     001                  .BYTE   1                               ;HOW MANY TIMES + 2
 882  001560  016767  015066  177420          MOV     DUBASE,KEEPADD  ;SAVE
 883  001566  004767  014726                  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  014774                          MVECTO                          ;MESSAGE
 887  001604  104405                          PARAM                           ;CONVERT STRING
 888  001606  000300                          300                             ;LOW LIMIT
 889  001610  000776                          776                             ;HIGH LIMIT
 890  001612  017174                          DURIV                           ;STORE AT THIS LOCATION
 891  001614     001                  .BYTE   1                               ;MASK
 892  001615     004                  .BYTE   4                               ;HOW MANY TIMES + 2
 893  001616  016767  015352  177370          MOV     DURIV,KEEPIV    ;SAVE
 894  001624  016767  015344  177360          MOV     DURIV,BASEIV    ;SET UP FOR ROTATION
 895  001632  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
 896  001634  015077                          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  015156                          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  015341                        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  015625                        MLEVEL                          ;MESSAGE
002030  104405                        PARAM                           ;CONVERT STRING
002032  000004                        4                               ;LOW LIMIT
002034  000007                        7                               ;HIGH LIMIT
002036  016514                        DUPRT                           ;STORE AT THIS LOCATION
002040     000                .BYTE   0                               ;MASK
002041     001                .BYTE   1                               ;HOW MANY TIMES + 2
002042  004767  014376                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  015652                        MSYNC                           ;MESSAGE
002052  122767  000061  014224  3$:   CMPB    #'1,INBUF               ;IS IT "1" ?
002060  001003                        BNE     1$
002062  105067  177110                CLRB    SYNCNO                  ;000
002066  000412                        BR      4$
002070  122767  000062  014206  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  015720                        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  015766                        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  016033                        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  016107                          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  010654  010554          MOV     BRW,TTST                ;RESTORE NORMAL SCOPE LOOP
  995  002212  016767  010650  010550          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  015562                          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  010510          MOV     #NOP,TTST
 1006  002256  012767  000240  010504          MOV     #NOP,TTST+2             ;SET UP TO LOCK
 1007  002264  000406                          BR      2$
 1008  002266  016767  010572  010472  1$:     MOV     BRW,TTST
 1009  002274  016767  010566  010466          MOV     BRX,TTST+2              ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
 1010  002302  032777  000002  176570  2$:     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  015547                          MTSTPC                          ;MESSAGE
 1014  002316  104405                          PARAM                           ;CONVERT STRING
 1015  002320  002350                          TST1                            ;LOW LIMIT
 1016  002322  011512                          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$:     MOV     #TST1,RTRN              ;START AT TEST 1
 1022  002340  104402  015543          4$:     TYPE    ,MR                     ;TYPE R
 1023  002344  000177  176544                  JMP     @RTRN                   ;START TESTING
 1024
 1025
```

```
                                        ;;THIS TEST VERIFYS THAT BY SENDING ONLY ONE SYNC
                                        ;;CHARACTER (TWO SELECTED BY STRAPPING ) RECACT =0
                                        ;;THEN SEND ONE ORDINARY CHARACTER (TO BREAK UP THE SEQUENCE)
                                        ;;RECACT =0...............IT WILL TAKE TWO MORE SYNC CHARS
                                        ;;BEFORE RECACT =1
                                        ;;NOTE: THIS TEST WILL ONLY WORK WHEN TWO SYNC CHARS
                                        ;;HAS BEEN BEEN SELECTED ...OTHERWISE JUMP AROUND THIS TEST
                                        ;;MODE:SYNC INTERNAL (SYNINT)
                                        ;;PARITY: NOPAR
                                        ;;LENGTH: EIGHT
                                        ;;
                                        ;;THIS TEST CHECKS ONLY THE RECEIVER SECTION
                                        ;;
002350  012767  000001  176550  TST1:   MOV     #1,TSTNO        ;SAVE THIS
002356  012767  002652  176532          MOV     #TST2,NEXT              ;GO TO THIS TEST WHEN THRU
002364  105767  176606                  TSTB    SYNCNO  ;TEST FOR  # OF SYNC CHARS REQUIRED
002370  100127                          BPL     4$      ;IF NOT TWO  GET OUT OF TEST
002372  052777  000400  014564          BIS     #MRESET,@TXCSR  ;MASTER RESET
002400  012777  030000  014552          MOV     #SYNINT,@PARCSR ;SET THE MODE
002406  052777  000400  014550          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINT DATA,CLK.BREAK,&MAINTENANCE MODE
002414  012777  064001  014542          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                ;SET MODE  ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
002422  012777  036026  014530          MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
002430  052777  000020  014512          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                        ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
002436  042777  020000  014520          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
002444  052777  020000  014512          BIS     #CLK,@TXCSR     ;POKE CLK UP
                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
002452  042777  020000  014504          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
002460  052777  020000  014476          BIS     #CLK,@TXCSR     ;POKE CLK UP
002466  012767  000010  176444          MOV     #8.,SHIFT       ;# OF SHIFTS
002474  012767  000026  176442          MOV     #26,TEMP1       ;SYNC CHAR TO BE SHIFTED IN
002502  004767  014146                  JSR     PC,RPOKE        ;SHIFT IN THIS SYNC CHAR
002506  032777  004000  014434          BIT     #RECACT,@RXCSR  ;RECACT = 0 ?
002514  001401                          BEQ     1$
002516  104000                          HLT     ;RECACT SHOULD BE 0
002520                          1$:
002520  012767  000010  176412          MOV     #8.,SHIFT       ;# OF SHIFTS
002526  012767  000025  176410          MOV     #25,TEMP1       ;ANY CHARACTER
002534  004767  014114                  JSR     PC,RPOKE        ;SHIFT IN THIS CHARACTER
                                ;YOU HAVE JUST LOST SYNCRONIZATION......................
                                ;POKE THE CLK TWICE TO GET INTO SYNCRONIZATION
                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
002540  012777  020000  014416          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
002546  052777  020000  014410          BIS     #CLK,@TXCSR     ;POKE CLK UP
                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
002554  042777  020000  014402          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
002562  052777  020000  014374          BIS     #CLK,@TXCSR     ;POKE CLK UP
002570  012767  000002  176344          MOV     #2,COUNT        ;# OF SYNC CHARS
002576  032777  004000  014344  2$:     BIT     #RECACT,@RXCSR  ;RECACT = 0 ?
002604  001401                          BEQ     3$
002606  104000                          HLT             ;RECACT SHOULD BE  0
002610                          3$:
```

```
 1082  002610  012767  000010  176322          MOV     #8.,SHIFT       ;# OF SHIFTS
 1083  002616  012767  000026  176320          MOV     #26,TEMP1       ;SYNC CHAR
 1084  002624  004767  014024                  JSR     PC,RPOKE        ;SHIFT IN THIS SYNC CHAR
 1085  002630  005367  176306                  DEC     COUNT
 1086  002634  001360                          BNE     2$      ;IS COUNT = 0 ? NO GO AGAIN
 1087  002636  032777  004000  014304          BIT     #RECACT,@RXCSR  ;RECACT = 1 ?
 1088  002644  001001                          BNE     4$
 1089  002646  104000                          HLT     ;RECACT SHOULD BE ASSERTED
 1090  002650                          4$:
 1091  002650  104400                          SCOPE
```

```
 1092                                   ;;THIS TEST VERIFYS MODE SELECT.....
 1093                                   ;;SYNC EXTERNAL VS. SYNC INTERNAL
 1094                                   ;;
 1095                                   ;BASICALLY THE TEST CHECKS THAT THE RECEIVED
 1096                                   ;DATA FREEZES  IN SYNC INTERNAL
 1097                                   ;IN SYNC EXTERNAL THIS DATA IS TRANSPARENT
 1098                                   ;THIS TEST ONLY APPLIES TO THE RECEIVER SECTION
 1099                                   ;;LENGTH: EIGHT
 1100                                   ;NOTE:SEARCH SYNC IS NOT SET
 1101  002652  012767  000002  176246  TST2:   MOV     #2,TSTNO          ;SAVE THIS
 1102  002660  012767  003250  176230          MOV     #TST3,NEXT                ;GO TO THIS TEST WHEN THRU
 1103  002666  052777  000400  014270          BIS     #MRESET,@TXCSR    ;MASTER RESET
 1104  002674  012777  030000  014256          MOV     #SYNINT,@PARCSR   ;SET THE MODE
 1105  002702  052777  000400  014254          BIS     #MRESET,@TXCSR    ;MASTER RESET
 1106
 1107                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1108  002710  012777  064001  014246          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1109
 1110                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1111  002716  012777  036026  014234          MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
 1112                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1113  002724  042777  020000  014232          BIC     #CLK,@TXCSR       ;POKE CLK DOWN
 1114  002732  052777  020000  014224          BIS     #CLK,@TXCSR       ;POKE CLK UP
 1115                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1116  002740  042777  020000  014216          BIC     #CLK,@TXCSR       ;POKE CLK DOWN
 1117  002746  052777  020000  014210          BIS     #CLK,@TXCSR       ;POKE CLK UP
 1118  002754  012767  000010  176156          MOV     #8.,SHIFT         ;# OF SHIFTS
 1119  002762  012767  000125  176154          MOV     #125,TEMP1        ;DATA CHARACTER
 1120  002770  016703  014160                  MOV     RXDBUF,R3         ;FOR ERROR MESSAGE
 1121  002774  042777  040000  014162  1$:     BIC     #MTDATA,@TXCSR    ;CLEAR MAINT DATA
 1122  003002  000241                          CLC
 1123  003004  006067  176134                  ROR     TEMP1   ;FORCE CARRY
 1124  003010  103003                          BCC     2$
 1125  003012  052777  040000  014144          BIS     #MTDATA,@TXCSR    ;SET MTDATA
 1126  003020  042777  020000  014136  2$:     BIC     #CLK,@TXCSR       ;POKE CLK
 1127  003026  052777  020000  014130          BIS     #CLK,@TXCSR       ;
 1128  003034  012700  000377                  MOV     #377,R0 ;EXPECTED
 1129  003040  017701  014110                  MOV     @RXDBUF,R1        ;ACTUAL
 1130  003044  020001                          CMP     R0,R1
 1131  003046  001401                          BEQ     3$
 1132  003050  104002                          HLT     2
 1133                                         ;DATA CHARACTER SHOULD COMPARE......
 1134  003052                         3$:
 1135  003052  005367  176062                  DEC     SHIFT   ;IS IT THE LAST SHIFT ?
 1136  003056  001346                          BNE     1$      ;NO ...SHIFT SOME MORE
 1137  003060  052777  000400  014076          BIS     #MRESET,@TXCSR    ;MASTER RESET
 1138  003066  012777  020000  014064          MOV     #SYNEXT,@PARCSR   ;SET THE MODE
 1139  003074  052777  000400  014062          BIS     #MRESET,@TXCSR    ;MASTER RESET
 1140
 1141                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1142  003102  012777  064001  014054          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1143
 1144                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1145  003110  012777  026026  014042          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
 1146                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1147  003116  042777  020000  014040          BIC     #CLK,@TXCSR       ;POKE CLK DOWN
```

```
1148   003124  052777  020000  014032          BIS     #CLK,@TXCSR     ;POKE CLK UP
1149   003132  012767  000010  176000          MOV     #8.,SHIFT       ;# OF SHIFTS
1150   003140  012767  000125  175776          MOV     #125,TEMP1      ;DATA CHARACTER
1151   003146  005067  175774                  CLR     TEMP2
1152   003152  105167  175770                  COMB    TEMP2   ;MAKE LOW BYTE ALL 1'S
1153                                           ;TO MATCH RXDBUF'S CONTENTS AFTER A MASTER RESET
1154   003156  042777  040000  014000  4$:     BIC     #MTDATA,@TXCSR  ;CLR MAINT DATA
1155   003164  000241                          CLC
1156   003166  006067  175752                  ROR     TEMP1   ;FORCE CARRY
1157   003172  103003                          BCC     5$
1158   003174  052777  040000  013762          BIS     #MTDATA,@TXCSR
1159   003202  106067  175740          5$:     RORB    TEMP2   ;PICK UP CARRY BIT
1160   003206  042777  020000  013750          BIC     #CLK,@TXCSR
1161   003214  052777  020000  013742          BIS     #CLK,@TXCSR
1162   003222  016700  175720                  MOV     TEMP2,R0        ;EXPECTED
1163   003226  017701  013722                  MOV     @RXDBUF,R1      ;ACTUAL
1164   003232  020001                          CMP     R0,R1
1165   003234  001401                          BEQ     6$
1166   003236  104002                          HLT     2       ;DATA CHARACTER SHOULD COMPARE...
1167                                           ;THE DATA CHARACTER SHOULD BE SEEN AS IT
1168                                           ;SHIFTS ACROSS THE RECEIVER DATA OUTPUT
1169   003240                          6$:
1170   003240  005367  175674                  DEC     SHIFT
1171   003244  001344                          BNE     4$
1172   003246  104400                          SCOPE
```

```
 1173                                   ;;THIS TEST VERIFYS TX DONE FUNCTION, DONE = 1
 1174                                   ;;ALSO   VERIFYS THAT THE TRANSMITTER CHIP IDLES "SYNC" CHARACTER
 1175                                   ;;WHEN NO NEW CHARACTER IS LOADED INTO TXDBUF("SYNC" = BINARY COUNT PATTERN)
 1176                                   ;;MODE:  SYNC INTERNAL
 1177                                   ;;PARITY:  NO PARITY (NOPAR)
 1178                                   ;;LENGTH:  EIGHT
 1179                                   ;;
 1180  003250  012767  000003  175650  TST3:   MOV     #3,TSTNO        ;SAVE THIS
 1181  003256  012767  003546  175632          MOV     #TST4,NEXT              ;GO TO THIS TEST WHEN THRU
 1182  003264  012767  003450  175626          MOV     #6$,LOCK        ;SET UP FOR SCOPE LOOP
 1183  003272  012704  036000                  MOV     #SYNINT!EIGHT!NOPAR!0,R4        ;MODE ETC.
 1184  003276  052777  000400  013660  1$:     BIS     #MRESET,@TXCSR  ;MASTER RESET
 1185  003304  012777  030000  013646          MOV     #SYNINT,@PARCSR ;SET THE MODE
 1186  003312  052777  000400  013644          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1187  003320  012777  004020  013636          MOV     #MINT!SEND,@TXCSR
 1188  003326  010477  013626                  MOV     R4,@PARCSR
 1189  003332  105777  013626                  TSTB    @TXCSR          ;TXDONE?
 1190  003336  100401                          BMI     2$
 1191  003340  104000                          HLT                     ;TXDONE SHOULD BE SET
 1192  003342                          2$:
 1193  003342  112777  000021  013620          MOVB    #21,@TXDBUF     ;LOAD ANY CHAR
 1194                                          ;POKE   CLK TO GET INTO SYNCIRONIZATION
 1195  003350  052777  020000  013606          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1196  003356  042777  020000  013600          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1197  003364  105777  013574                  TSTB    @TXCSR
 1198  003370  100001                          BPL     3$
 1199  003372  104000                          HLT                     ;TXDONE SHOULD BE CLR
 1200  003374                          3$:
 1201  003374  052777  020000  013562          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1202  003402  042777  020000  013554          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1203  003410  105777  013550                  TSTB    @TXCSR
 1204  003414  100401                          BMI     4$
 1205  003416  104000                          HLT                     ;TXDONE SHOULD BE SET
 1206  003420                          4$:
 1207  003420  012767  000007  175512          MOV     #7,SHIFT
 1208  003426                          5$:
 1209  003426  052777  020000  013530          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1210  003434  042777  020000  013522          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1211  003442  005367  175472                  DEC     SHIFT
 1212  003446  001367                          BNE     5$      ;SHIFT OUT THE "21"
 1213  003450  016703  013510          6$:     MOV     TXCSR,R3        ;FOR ERROR MESSAGE
 1214                                                  ;THE BIT WINDOW IS RE GENERATED INTO
 1215                                                  ;A CHARACTER AND LEFT PRESENTED IN R1
 1216                                                  ;FOR THE COMPARE OPERATION. IF YOU WANT TO
 1217                                                  ;LOCK ON A PARTICULAR SYNC CHARACTER...
 1218                                                  ;SET SWR09=1
 1219  003454  005000                          CLR     R0
 1220  003456  150400                          BISB    R4,R0   ;EXPECT "SYNC"
 1221  003460  012767  000010  175452          MOV     #8.,SHIFT       ;# OF SHIFTS
 1222  003466  005001                          CLR     R1
 1223  003470                          7$:
 1224  003470  052777  020000  013466          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1225  003476  042777  020000  013460          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1226  003504  000241                          CLC
 1227  003506  032777  002000  013450          BIT     #BITW,@TXCSR    ;BITW = ?
 1228  003514  001401                          BEQ     8$
```

```
 1229   003516  000261                          SEC     ;SET CARRY
 1230   003520  106001                  8$:     RORB    R1      ;PICK UP CARRY
 1231   003522  005367  175412                  DEC     SHIFT
 1232   003526  001360                          BNE     7$      ;FINISH THAT CHARACTER
 1233   003530  020001                          CMP     R0,R1   ;CMP EXPECTED VS ACTUAL
 1234   003532  001401                          BEQ     9$
 1235   003534  104003                          HLT     3       ;SYNC CHAR IS NOT CORRECT
 1236   003536                          9$:
 1237   003536  104401                          SCOP1
 1238   003540  105204                          INCB    R4      ;SET UP FOR NEXT SYNC HOLDING REG.
 1239   003542  001255                          BNE     1$      ;FINISHED WITH BINARY COUNT PATTERN ?
 1240
 1241   003544  104400                          SCOPE
 1242
```

```
                                        ;;THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER
                                        ;;RESET" WHILE IN STRIP SYNC MODE.
                                        ;;THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR
                                        ;;WHEN STRIP SYNC IS SET AND SYNC CHARACTERS ARE SENT
                                        ;;BUT IF AN ERROR SHOULD OCCUR...THIS AUTOMATIC RESET
                                        ;;IS DISCOMBOBULATED
                                        ;;IE:  FORCE OVERRUN (OVERRUN) WHILE STRIP SYNC IS SET
                                        ;;BY TRANSMITTING A DATA CHARACTER THEN TRANSMIT A SYNC CHARACTER
                                        ;;AND DON'T READ THAT DATA CHARACTER.  NOTE:  NORMALLY THE LOGIC
                                        ;;RESETS THE RXDONE & ERROR FLAGS PROVIDING THAT ONLY SYNC CHARACTERS ARE
                                        ;;STRIPPED
                                        ;;MODE:  SYNC EXTERNAL (SYNEXT)
                                        ;;LENGTH:  EIGHT
                                        ;;NOTE:  THIS TEST USES BOTH RECEIVER AND TRANSMITTER LOGIC
                                        ;;
003546  012767  000004  175352  TST4:   MOV     #4,TSTNO        ;SAVE THIS
003554  012767  004050  175334          MOV     #TST5,NEXT              ;GO TO THIS TEST WHEN THRU

003562  052777  000400  013374          BIS     #MRESET,@TXCSR  ;MASTER RESET
003570  012777  020000  013362          MOV     #SYNEXT,@PARCSR ;SET THE MODE
003576  052777  000400  013360          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINTENANCE MODE & SEND
                                        ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
003604  012777  004020  013352          MOV     #MINT!SEND,@TXCSR

                                ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
003612  012777  026026  013340          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
003620  112777  000026  013342          MOVB    #26,@TXDBUF     ;LOAD SYNC CHAR
003626  052777  000420  013314          BIS     #SYNSCH!STPSYN,@RXCSR   ;SET SYNC SEARCH & STRIP SYNC
003634  016703  013314                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
003640  012767  000003  175274          MOV     #3,COUNT        ;# OF TIMES SYNC WILL BE SENT
003646  052777  020000  013310          BIS     #CLK,@TXCSR     ;POKE CLK UP
003654  042777  020000  013302          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003662  012767  000010  175250  1$:     MOV     #8.,SHIFT       ;# OF SHIFTS
003670                          2$:
003670  052777  020000  013266          BIS     #CLK,@TXCSR     ;POKE CLK UP
003676  042777  020000  013260          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003704  005367  175230                  DEC     SHIFT
003710  001367                          BNE     2$
003712  105777  013232                  TSTB    @RXCSR          ;RXDONE?
003716  100001                          BPL     3$
003720  104000                          HLT                     ;RXDONE SHOULD NOT ASSERT
003722                          3$:
003722  005367  175214                  DEC     COUNT
003726  001355                          BNE     1$
003730  012700  000026                  MOV     #26,R0          ;EXPECTED
003734  017701  013214                  MOV     @RXDBUF,R1      ;ACTUAL
003740  020001                          CMP     R0,R1
003742  001401                          BEQ     4$
003744  104002                          HLT     2               ;NOTE THAT OVERRUN SHOULD NOT OCCUR, ALSO
                                                                ;SECOND & 3RD SYNC CHARACTER CAME FROM
                                                                ;SYNC HOLDING REGISTER
003746                          4$:
003746  012767  000003  175166          MOV     #3,COUNT        ;# OF TIMES
```

```
1299   003754  112777  000025  013206          MOVB    #25,@TXDBUF     ;LOAD ANY CHAR....HOWEVER...
1300                                                    ;ONE MORE SYNC CHAR WILL BE SENT BEFORE
1301                                                    ;THE "25" CHAR IS SENT (THE DNA BIT IS
1302                                                    ;ALREADY UP)
1303
1304   003762  012767  000010  175150  5$:     MOV     #8.,SHIFT       ;# OF SHIFTS
1305
1306   003770                          6$:
1307   003770  052777  020000  013166          BIS     #CLK,@TXCSR     ;POKE CLK UP
1308   003776  042777  020000  013160          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1309   004004  005367  175130                  DEC     SHIFT
1310   004010  001367                          BNE     6$
1311   004012  005367  175124                  DEC     COUNT
1312   004016  001361                          BNE     5$
1313   004020  105777  013124                  TSTB    @RXCSR  ;RXDONE = 1 ?
1314   004024  100401                          BMI     7$
1315   004026  104000                          HLT     ;RXDONE SHOULD BE SET
1316   004030                          7$:
1317   004030  012700  140026                  MOV     #RXERR!OVRRUN!26,R0 ;EXPECTED
1318   004034  017701  013114                  MOV     @RXDBUF,R1      ;ACTUAL
1319   004040  020001                          CMP     R0,R1
1320   004042  001401                          BEQ     8$
1321   004044  104002                          HLT     2               ;NOTE THAT OVRRUN SHOULD OCCUR,
1322                                                                   ;ALSO SECOND SYNC CHARACTER CAME
1323                                                                   ;FROM SYNC HOLDING REGISTER
1324                                                                   ;SUMMARY:  THE OVRRUN STOPPED
1325                                                                   ;THE AUTOMATIC RESETTING OF
1326                                                                   ;RXDONE & ERROR FLAGS.....CHECK THIS
1327   004046                          8$:
1328   004046  104400                          SCOPE
```

```
                                        ;;THIS TEST VERIFYS THAT EITHER SEQUENCE OF
                                        ;;LOADING TXDBUF AND SETTING SEND
                                        ;;DOES CAUSE TRANSMISSION
                                        ;;MODE:   SYNC EXT
                                        ;;LENGTH:   EIGHT
                                        ;;
004050  012767  000005  175050  TST5:   MOV     #5,TSTNO        ;SAVE THIS
004056  012767  004256  175032          MOV     #TST6,NEXT              ;GO TO THIS TEST WHEN THRU

004064  052777  000400  013072          BIS     #MRESET,@TXCSR  ;MASTER RESET
004072  012777  020000  013060          MOV     #SYNEXT,@PARCSR ;SET THE MODE
004100  052777  000400  013056          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINTENANCE MODE & SEND
                                        ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
004106  012777  004020  013050          MOV     #MINT!SEND,@TXCSR

                                ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
004114  012777  026026  013036          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
004122  016703  013036                  MOV     TXCSR,R3        ;SET UP FOR ERROR MESSAGE
004126  042777  000020  013030          BIC     #SEND,@TXCSR    ;DROP SEND
004134  012767  000025  175002          MOV     #25,TEMP1
004142  112777  000025  013020          MOVB    #25,@TXDBUF     ;LOAD CHARACTER
004150  052777  000020  013006          BIS     #SEND,@TXCSR    ;SET SEND
                                ;GET INTO SYNCRONIZATION
004156  052777  020000  013000          BIS     #CLK,@TXCSR     ;POKE CLK UP
004164  042777  020000  012772          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
004172  012767  000010  174740          MOV     #8.,SHIFT       ;# OF SHIFTS
004200  005000                  1$:     CLR     R0
004202  006067  174736                  ROR     TEMP1
004206  103002                          BCC     2$
004210  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BIT WINDOW

004214                          2$:
004214  052777  020000  012742          BIS     #CLK,@TXCSR     ;POKE CLK UP
004222  042777  020000  012734          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
004230  017701  012730                  MOV     @TXCSR,R1       ;ACTUAL
004234  042701  075777                  BIC     #075777,R1      ;SAVE BIT WINDOW & DNA
004240  020001                          CMP     R0,R1
004242  001401                          BEQ     3$
004244  104003                          HLT     3               ;BIT WINDOW DID NOT MATCH ACTUAL DATA BIT
                                                                ;ALSO CHECK DNA
004246                          3$:
004246  005367  174666                  DEC     SHIFT
004252  001352                          BNE     1$

004254  104400                          SCOPE
```

```
1377                                    ;;THIS TEST VERIFYS THAT DROPPING OF SEND IN THE
1378                                    ;;MIDDLE OF TRANSMITTING A CHARACTER DOES INDEED
1379                                    ;;FINISH TRANSMITTING THAT CHARACTER
1380                                    ;;MODE:  SYNC EXT
1381                                    ;;LENGTH:  EIGHT
1382                                    ;;
1383
1384  004256  012767  000006  174642  TST6:   MOV     #6,TSTNO        ;SAVE THIS
1385  004264  012767  004472  174624          MOV     #TST7,NEXT              ;GO TO THIS TEST WHEN THRU
1386
1387  004272  052777  000400  012664          BIS     #MRESET,@TXCSR  ;MASTER RESET
1388  004300  012777  020000  012652          MOV     #SYNEXT,@PARCSR ;SET THE MODE
1389  004306  052777  000400  012650          BIS     #MRESET,@TXCSR  ;MASTER RESET
1390
1391                                    ;SET MAINTENANCE MODE & SEND
1392                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1393  004314  012777  004020  012642          MOV     #MINT!SEND,@TXCSR
1394
1395                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1396  004322  012777  026026  012630          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
1397  004330  016703  012630                  MOV     TXCSR,R3        ;SETUP FOR ERROR MESSAGE
1398  004334  112777  000252  012626          MOVB    #252,@TXDBUF    ;LOAD DATA CHAR.
1399  004342  012767  000252  174574          MOV     #252,TEMP1      ;SHIFTED CHAR
1400  004350  012767  000010  174562          MOV     #8.,SHIFT       ;# OF SHIFTS
1401                                          ;GET INTO SYNCRONIZATION
1402  004356  052777  020000  012600          BIS     #CLK,@TXCSR     ;POKE CLK UP
1403  004364  042777  020000  012572          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1404
1405  004372  005000                  1$:     CLR     R0
1406  004374  006067  174544                  ROR     TEMP1           ;FORCE CARRY
1407  004400  103002                          BCC     2$
1408  004402  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BIT WINDOW
1409
1410  004406                          2$:
1411  004406  052777  020000  012550          BIS     #CLK,@TXCSR     ;POKE CLK UP
1412  004414  042777  020000  012542          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1413  004422  017701  012536                  MOV     @TXCSR,R1       ;ACTUAL
1414  004426  042701  075777                  BIC     #075777,R1      ;SAVE ONLY BIT WINDOW & DNA
1415  004432  020001                          CMP     R0,R1
1416  004434  001401                          BEQ     3$
1417  004436  104003                          HLT     3               ;BIT WINDOW DID NOT MATCH
1418                                                                  ;ACTUAL DATA BIT
1419  004440                          3$:
1420  004440  005367  174474                  DEC     SHIFT
1421  004444  022767  000003  174466          CMP     #3,SHIFT
1422  004452  001003                          BNE     4$
1423  004454  042777  000020  012502          BIC     #SEND,@TXCSR    ;DROP SEND
1424  004462  005767  174452          4$:     TST     SHIFT
1425  004466  001341                          BNE     1$              ;DO IT AGAIN?
1426
1427  004470  104400                  SCOPE
1428
```

```
 1429                                   ;;THIS TEST VERIFYS THAT RXDONE ASSERTS WHEN STRIP SYNC IS SET
 1430                                   ;;MODE: SYNC INTERNAL
 1431                                   ;;LENGTH: EIGHT
 1432                                   ;;
 1433  004472  012767  000007  174426  TST7:   MOV     #7,TSTNO        ;SAVE THIS
 1434  004500  012767  004746  174410          MOV     #TST8,NEXT              ;GO TO THIS TEST WHEN THRU
 1435  004506  052777  000400  012450          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1436  004514  012777  030000  012436          MOV     #SYNINT,@PARCSR ;SET THE MODE
 1437  004522  052777  000400  012434          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1438
 1439                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1440  004530  012777  064001  012426          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1441
 1442                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1443  004536  012777  036026  012414          MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
 1444
 1445  004544  052777  000420  012376          BIS     #SYNSCH!STPSYN,@RXCSR   ;SET SYNC SEARCH &
 1446                                                                  ;STRIP SYNC
 1447                                          ;POKE CLK TO GET RECEIVER INTO SYNCRONIZATION.....
 1448  004552  042777  020000  012404          BIC     #CLK,@TXCSR     ;POKE CLK
 1449  004560  052777  020000  012376          BIS     #CLK,@TXCSR     ;
 1450                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1451  004566  042777  020000  012370          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1452  004574  052777  020000  012362          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1453  004602  016703  012346                  MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
 1454  004606  012767  000002  174326          MOV     #2,COUNT        ;# OF TIMES OF SYNC CHARS.
 1455                                          ;TEST TO SEE HOW MANY SYNC CHARS  NEEDED
 1456  004614  105767  174356                  TSTB    SYNCNO
 1457  004620  100402                          BMI     1$      ;WILL IT BE ONE OR TWO ?
 1458  004622  005367  174314                  DEC     COUNT   ;MAKE IT ONE LESS
 1459  004626  012767  000010  174304  1$:     MOV     #8.,SHIFT       ;#OF SHIFTS
 1460  004634  012767  000026  174302          MOV     #26,TEMP1       ;SYNC CHAR
 1461  004642  004767  012006                  JSR     PC,RPOKE
 1462  004646  005367  174270                  DEC     COUNT   ;IS IT THE LAST SYNC CHAR ?
 1463  004652  001365                          BNE     1$      ;GO AGAIN AND SHIFT IN ANOTHER SYNC CHAR
 1464  004654  032777  004000  012266          BIT     #RECACT,@RXCSR  ;RECACT=1?
 1465  004662  001001                          BNE     2$
 1466  004664  104000                          HLT     ;RECACT SHOULD BE ASSERTED
 1467  004666                          2$:
 1468  004666  105777  012256                  TSTB    @RXCSR          ;RXDONE=0?
 1469  004672  100001                          BPL     3$
 1470  004674  104000                          HLT                     ;RXDONE SHOULD NOT BE ASSERTED
 1471  004676                          3$:
 1472  004676  012767  000010  174234          MOV     #8.,SHIFT       ;#OF SHIFTS
 1473  004704  012767  000025  174232          MOV     #25,TEMP1       ;ANY CHARACTER
 1474  004712  004767  011736                  JSR     PC,RPOKE
 1475  004716  105777  012226                  TSTB    @RXCSR          ;RXDONE=1?
 1476  004722  100401                          BMI     4$
 1477  004724  104000                          HLT                     ;RXDONE SHOULD NOW BE ASSERTED
 1478  004726                          4$:
 1479  004726  012700  000025                  MOV     #25,R0          ;EXPECTED
 1480  004732  017701  012216                  MOV     @RXDBUF,R1      ;ACTUAL
 1481  004736  020001                          CMP     R0,R1
 1482  004740  001401                          BEQ     5$
 1483  004742  104002                          HLT     2               ;CHARACTERS SHOULD BE MATCHED
 1484  004744                          5$:
```

H03

SEQ 0033

```
 1485  004744  104400                          SCOPE
 1486
```

```
1487                                    ;;THIS TEST VERIFYS THAT BY DROPPING SYNSCN
1488                                    ;;IN THE MIDDLE OF A CHARACTER, SYNC CHARACTER SEQUENCE
1489                                    ;;IS NEEDED BEFORE RECACT, RXDONE ASSERT AGAIN.
1490                                    ;;ALSO NOTE:  SINCE RECACT IS DEPENDENT ON MATCH DETECT,
1491                                    ;;AND IF SYNSCH IS DROPPED IN THE MIDDLE OF
1492                                    ;;A SYNC CHARACTER AND THEN RAISED AGAIN ;RXDONE SHOULD
1493                                    ;;NOT ASSERT UNTIL NEW SYNC CHARACTER SEQUENCE
1494                                    ;;MODE:  SYNC INTERNAL (SYNINT)
1495                                    ;;LENGTH:  EIGHT
1496                                    ;;
1497
1498  004746  012767  000010  174152  TST8:   MOV     #8,TSTNO        ;SAVE THIS
1499  004754  012767  005476  174134          MOV     #TST9,NEXT              ;GO TO THIS TEST WHEN THRU
1500
1501  004762  105767  174210                  TSTB    SYNCNO          ;TWO SYNC CHARACTERS SELECTED  ?
1502  004766  100002                          BPL     1$      ;IF ANSWER WAS NO  DO THIS TEST
1503  004770  000167  000500                  JMP     16$     ;IF ANSWER WAS YES  JUMP OVER THIS TEST
1504  004774                          1$:
1505  004774  052777  000400  012162          BIS     #MRESET,@TXCSR  ;MASTER RESET
1506  005002  012777  030000  012150          MOV     #SYNINT,@PARCSR ;SET THE MODE
1507  005010  052777  000400  012146          BIS     #MRESET,@TXCSR  ;MASTER RESET
1508
1509                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1510  005016  012777  064001  012140          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1511
1512                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1513  005024  012777  036026  012126          MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
1514
1515  005032  052777  000020  012110          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1516                                          .POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1517  005040  042777  020000  012116          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1518  005046  052777  020000  012110          BIS     #CLK,@TXCSR     ;POKE CLK UP
1519                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1520  005054  042777  020000  012102          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1521  005062  052777  020000  012074          BIS     #CLK,@TXCSR     ;POKE CLK UP
1522  005070  012767  000002  174044          MOV     #2,COUNT        ;# OF TIMES
1523  005076  016703  012052                  MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
1524  005102  012767  000010  174030  2$:     MOV     #8.,SHIFT       ;# OF SHIFTS
1525  005110  012767  000026  174026          MOV     #26,TEMP1       ;SYNC CHAR.
1526  005116  004767  011532                  JSR     PC,RPOKE
1527  005122  005367  174014                  DEC     COUNT
1528  005126  001403                          BEQ     3$
1529                                  ;TEST TO SEE HOW MANY SYNC CHARACTERS NEEDED
1530  005130  105767  174042                  TSTB    SYNCNO
1531  005134  100762                          BMI     2$
1532
1533  005136  032777  004000  012004  3$:     BIT     #RECACT,@RXCSR  ;RECACT=1?
1534  005144  001001                          BNE     4$
1535  005146  104000                          HLT                     ;RECACT SHOULD BE SET
1536  005150                          4$:
1537  005150  012767  000004  173762          MOV     #4,SHIFT        ;# OF SHIFTS
1538  005156  012767  000026  173760          MOV     #26,TEMP1       ;SYNC CHAR.
1539  005164  004767  011464                  JSR     PC,RPOKE
1540  005170  032777  004000  011752          BIT     #RECACT,@RXCSR  ;RECACT=1?
1541  005176  001001                          BNE     5$
1542  005200  104000                          HLT                     ;RECACT SHOULD  STILL BE SET
```

```
1543  005202                                  5$:
1544  005202  042777  000020  011740                BIC     #SYNSCH,@RXCSR  ;DROP SEARCH SYNC
1545  005210  032777  004000  011732                BIT     #RECACT,@RXCSR  ;RECACT=0?
1546  005216  001401                                BEQ     6$
1547  005220  104000                                HLT                     ;RECACT SHOULD NOT BE SET
1548                                          ;NOW SHIFT TWO BITS TO ALLOW SEARCH SYNC =0 TO TAKE
1549                                          ;EFFECT IN THE LOGIC(THIS ALLOWS THE RECEIVER CHIP TO SEE
1550                                          ;THE DROPPING OF SEARCH SYNC)...MATCH DETECT IN THE  REC.CHIP SHOULD ALSO DROP
1551  005222                                  6$:
1552  005222  012767  000002  173710                MOV     #2,SHIFT        ;# OF SHIFTS
1553  005230  004767  011420                        JSR     PC,RPOKE
1554  005234  052777  000020  011706                BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1555  005242  032777  004000  011700                BIT     #RECACT,@RXCSR
1556  005250  001401                                BEQ     7$
1557  005252  104000                                HLT     ;RECACT =0 ?
1558  005254                                  7$:
1559  005254  105777  011670                        TSTB    @RXCSR
1560  005260  100001                                BPL     8$
1561  005262  104000                                HLT     ;RXDONE = 0 ?
1562  005264                                  8$:
1563  005264  012767  000002  173646                MOV     #2,SHIFT        ;# OF SHIFTS
1564  005272  004767  011356                        JSR     PC,RPOKE
1565  005276  032777  004000  011644                BIT     #RECACT,@RXCSR  ;RECACT=0?
1566  005304  001401                                BEQ     9$
1567  005306  104000                                HLT                     ;RECACT SHOULD NOT BE SET
1568  005310                                  9$:
1569  005310  105777  011634                        TSTB    @RXCSR          ;RXDONE=0?
1570  005314  100001                                BPL     10$
1571  005316  104000                                HLT                     ;RXDONE SHOULD NOT BE ASSERTED
1572  005320                                  10$:
1573  005320  012700  000026                        MOV     #26,R0          ;EXPECTED
1574  005324  017701  011624                        MOV     @RXDBUF,R1      ;ACTUAL
1575  005330  020001                                CMP     R0,R1
1576  005332  001401                                BEQ     11$
1577  005334  104002                                HLT     2               ;CHARACTERS SHOULD BE MATCHED
1578  005336                                  11$:
1579  005336  012767  000002  173576                MOV     #2,COUNT        ;# OF TIMES OF SYNC CHARS.
1580                                                ;TEST TO SEE HOW MANY SYNC CHARS  NEEDED
1581  005344  105767  173626                        TSTB    SYNCNO
1582  005350  100402                                BMI     12$     ;WILL IT BE TWO OR ONE ?
1583  005352  005367  173564                        DEC     COUNT   ;IT WAS ONLY ONE NEEDED
1584  005356  012767  000010  173554  12$:         MOV     #8.,SHIFT       ;#OF SHIFTS
1585  005364  012767  000026  173552                MOV     #26,TEMP1       ;SYNC CHAR
1586  005372  004767  011256                        JSR     PC,RPOKE
1587  005376  005367  173540                        DEC     COUNT   ;IS IT THE LAST SYNC CHAR ?
1588  005402  001365                                BNE     12$     ;GO AGAIN AND SHIFT IN ANOTHER SYNC CHAR
1589  005404  032777  004000  011536                BIT     #RECACT,@RXCSR  ;RECACT=1?
1590  005412  001001                                BNE     13$
1591  005414  104000                                HLT     ;RECACT SHOULD BE ASSERTED
1592  005416                                  13$:
1593  005416  105777  011526                        TSTB    @RXCSR          ;RXDONE=0?
1594  005422  100001                                BPL     14$
1595  005424  104000                                HLT                     ;RXDONE SHOULD NOT BE ASSERTED
1596  005426                                  14$:
1597  005426  012767  000010  173504                MOV     #8.,SHIFT       ;#OF SHIFTS
1598  005434  012767  000025  173502                MOV     #25,TEMP1       ;ANY CHARACTER
```

```
1599   005442  004767  011206                  JSR     PC,RPOKE
1600   005446  105777  011476                  TSTB    @RXCSR          ;RXDONE=1?
1601   005452  100401                          BMI     15$
1602   005454  104000                          HLT                     ;RXDONE SHOULD NOW BE ASSERTED
1603   005456                          15$:
1604   005456  012700  000025                  MOV     #25,R0          ;EXPECTED
1605   005462  017701  011466                  MOV     @RXDBUF,R1      ;ACTUAL
1606   005466  020001                          CMP     R0,R1
1607   005470  001401                          BEQ     16$
1608   005472  104002                          HLT     2
                                                        ;CHARACTERS SHOULD BE MATCHED
1609                                           ;IF THIS  FAILS THEN CHECK THAT THE CORRECT
1610                                           ;RECEIVER CHIP IS BEING USED...WHAT IS
1611                                           ;HAPPENING IS THAT MATCH DETECT IS ASSERTING
1612                                           ;BEFORE A NEW SYNC CHARACTER SEQUENCE
1613                                           ;TRANSPIRES  THUS RXDONE ASSERTS TOO SOON
1614                                           ;AND OVER RUN OCCURS SINCE THE RECEIVER WAS NOT READ
1615                                           ;CONCLUSION::: IF OLDER RECEIVER CHIP
1616                                           ;IS BEING USED THEN REPLACE IT WITH A NEW
1617                                           ;RECEIVER CHIP IF 1 SYNC CHARACTER SEQUENCE IS DESIRED
1618   005474                          16$:
1619   005474  104400                          SCOPE
1620
```

```
 1621                                   ;;THIS TEST VERIFYS THAT HDX MODE DISQUALIFIES THE
 1622                                   ;;RECEIVER WHEN SEND IS ASSERTED
 1623                                   ;;MODE:  SYNC EXT
 1624                                   ;;LENGTH:  EIGHT
 1625                                   ;;NOTE:  THIS TEST WORKS ONLY IN MAINT. EXTERNAL MODE
 1626                                   ;;THIS TEST USES BOTH RECEIVER & TRANSMITTER LOGIC
 1627  005476  012767  000011  173422  TST9:   MOV     #9,TSTNO        ;SAVE THIS
 1628  005504  012767  006054  173404          MOV     #TST10,NEXT             ;GO TO THIS TEST WHEN THRU
 1629  005512  105767  173465                  TSTB    JMRBY
 1630  005516  100155                          BPL     7$              ;GET OUT OF THIS TEST IF "NO"
 1631  005520  016703  011430                  MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
 1632  005524  052777  000400  011432          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1633  005532  012777  020000  011420          MOV     #SYNEXT,@PARCSR ;SET THE MODE
 1634  005540  052777  000400  011416          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1635
 1636                                  ;SET MAINTENANCE MODE & SEND
 1637                                          ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
 1638  005546  012777  010020  011410          MOV     #MEXT!SEND,@TXCSR
 1639
 1640                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
 1641  005554  012777  026026  011376          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
 1642  005562  052777  000020  011360          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
 1643  005570  112777  000025  011372          MOVB    #25,@TXDBUF     ;ANY CHARACTER
 1644                                          ;POKE CLK FOR SYNCRONIZATION
 1645  005576  052777  020000  011360          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1646                                          ;WAIT FOR CABLE & DRIVER DELAYS
 1647  005604  016702  173326                  MOV     HOLD,R2 ;WAIT THIS AMT
 1648  005610                          64$:
 1649  005610  005302                          DEC     R2      ;WAIT
 1650  005612  001376                          BNE     64$
 1651                                          ;EXIT...
 1652  005614  042777  020000  011342          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1653                                          ;WAIT FOR CABLE & DRIVER DELAYS
 1654  005622  016702  173310                  MOV     HOLD,R2 ;WAIT THIS AMT
 1655  005626                          65$:
 1656  005626  005302                          DEC     R2      ;WAIT
 1657  005630  001376                          BNE     65$
 1658                                          ;EXIT...
 1659  005632  012767  000010  173300          MOV     #8.,SHIFT       ;# OF SHIFTS
 1660  005640                          1$:
 1661  005640  052777  020000  011316          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1662                                          ;WAIT FOR CABLE & DRIVER DELAYS
 1663  005646  016702  173264                  MOV     HOLD,R2 ;WAIT THIS AMT
 1664  005652                          66$:
 1665  005652  005302                          DEC     R2      ;WAIT
 1666  005654  001376                          BNE     66$
 1667                                          ;EXIT...
 1668  005656  042777  020000  011300          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1669                                          ;WAIT FOR CABLE & DRIVER DELAYS
 1670  005664  016702  173246                  MOV     HOLD,R2 ;WAIT THIS AMT
 1671  005670                          67$:
 1672  005670  005302                          DEC     R2      ;WAIT
 1673  005672  001376                          BNE     67$
 1674                                          ;EXIT...
 1675  005674  005367  173240                  DEC     SHIFT   ;# OF SHIFTS
 1676  005700  022767  000003  173232          CMP     #3,SHIFT        ;IS IT TIME TO LOAD NEXT CHAR ?
```

```
 1677  005706  001003                        BNE     2$
 1678  005710  112777  000024  011252        MOVB    #24,@TXDBUF     ;LOAD NEXT CHAR.
 1679  005716  005767  173216          2$:   TST     SHIFT
 1680  005722  001346                        BNE     1$
 1681  005724  105777  011220                TSTB    @RXCSR          ;RXDONE=1?
 1682  005730  100401                        BMI     3$
 1683  005732  104000                        HLT                     ;RXDONE SHOULD BE SET
 1684  005734                          3$:
 1685  005734  012700  000025                MOV     #25,R0          ;EXPECTED
 1686  005740  017701  011210                MOV     @RXDBUF,R1      ;ACTUAL
 1687  005744  020001                        CMP     R0,R1
 1688  005746  001401                        BEQ     4$
 1689  005750  104002                        HLT     2               ;CHARACTERS SHOULD COMPARE
 1690  005752                          4$:
 1691  005752  052777  000010  011204        BIS     #HDXEN,@TXCSR   ;SET HALF DUPLEX HDX
 1692  005760  012767  000010  173152        MOV     #8.,SHIFT       ;# OF SHIFT
 1693  005766                          5$:
 1694  005766  052777  020000  011170        BIS     #CLK,@TXCSR     ;POKE CLK UP
 1695                                        ;WAIT FOR CABLE & DRIVER DELAYS
 1696  005774  016702  173136                MOV     HOLD,R2 ;WAIT THIS AMT
 1697  006000                          68$:
 1698  006000  005302                        DEC     R2      ;WAIT
 1699  006002  001376                        BNE     68$
 1700                                        ;EXIT...
 1701  006004  042777  020000  011152        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1702                                        ;WAIT FOR CABLE & DRIVER DELAYS
 1703  006012  016702  173120                MOV     HOLD,R2 ;WAIT THIS AMT
 1704  006016                          69$:
 1705  006016  005302                        DEC     R2      ;WAIT
 1706  006020  001376                        BNE     69$
 1707                                        ;EXIT...
 1708  006022  005367  173112                DEC     SHIFT
 1709  006026  001357                        BNE     5$
 1710  006030  105777  011114                TSTB    @RXCSR          ;RXDONE=0?
 1711  006034  100001                        BPL     6$
 1712  006036  104000                        HLT                     ;RXDONE SHOULD NOT BE ASSERTED
 1713                                        ;CHECK OUT HDX LOGIC
 1714  006040                          6$:
 1715  006040  017701  011110                MOV     @RXDBUF,R1      ;ACTUAL
 1716  006044  020001                        CMP     R0,R1
 1717  006046  001401                        BEQ     7$
 1718  006050  104002                        HLT     2               ;CHARACTERS SHOULD COMPARE
 1719                                        ;NOTE THAT CHARACTER 25 WILL BE FROZEN
 1720                                        ;IN THE RXDBUF EVEN THOUGH CHARACTER 24 WAS
 1721                                        ;SENT TO THE RECEIVER
 1722  006052                          7$:
 1723  006052  104400                        SCOPE
 1724
```

```
1725                                    ;;THIS TEST VERIFYS THAT BREAK FORCES A SPACE CONDITION
1726                                    ;;ON THE LINE WHILE TRANSMITTING
1727                                    ;;THIS TEST USES BOTH THE RECEIVER AND TRANSMITTER LOGIC
1728                                    ;;MODE:  SYNC EXT (SYNEXT)
1729                                    ;;LENGTH:  EIGHT
1730                                    ;;
1731
1732  006054  012767  000012  173044  TST10:  MOV     #10,TSTNO       ;SAVE THIS
1733  006062  012767  006312  173026           MOV     #TST11,NEXT             ;GO TO THIS TEST WHEN THRU
1734  006070  052777  000400  011066           BIS     #MRESET,@TXCSR  ;MASTER RESET
1735  006076  012777  020000  011054           MOV     #SYNEXT,@PARCSR ;SET THE MODE
1736  006104  052777  000400  011052           BIS     #MRESET,@TXCSR  ;MASTER RESET
1737
1738                                    ;SET MAINTENANCE MODE & SEND
1739                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1740  006112  012777  004020  011044           MOV     #MINT!SEND,@TXCSR
1741
1742                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1743  006120  012777  026026  011032           MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
1744  006126  052777  000020  011014           BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1745  006134  016703  011014                   MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
1746  006140  012767  000002  172774           MOV     #2,COUNT        ;# OF TIMES
1747  006146  112777  000025  011014           MOVB    #25,@TXDBUF     ;ANY CHARACTER
1748                                           ;POKE CLK FOR SYNCRONIZATION
1749  006154  052777  020000  011002           BIS     #CLK,@TXCSR     ;POKE CLK UP
1750  006162  042777  020000  010774           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1751  006170  012700  000025                   MOV     #25,R0          ;EXPECTED
1752  006174  012767  000010  172736  1$:      MOV     #8.,SHIFT       ;# OF SHIFTS
1753
1754  006202                          2$:
1755  006202  052777  020000  010754           BIS     #CLK,@TXCSR     ;POKE CLK UP
1756  006210  042777  020000  010746           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1757  006216  005367  172716                   DEC     SHIFT
1758  006222  022767  000003  172710           CMP     #3,SHIFT
1759  006230  001003                           BNE     3$
1760  006232  112777  000024  010730           MOVB    #24,@TXDBUF     ;LOAD NEXT CHAR
1761  006240  005767  172674          3$:      TST     SHIFT
1762  006244  001356                           BNE     2$
1763  006246  105777  010676                   TSTB    @RXCSR          ;RXDONE=1?
1764  006252  100401                           BMI     4$
1765  006254  104000                           HLT
1766  006256                          4$:
1767  006256  017701  010672                   MOV     @RXDBUF,R1      ;ACTUAL
1768  006262  020001                           CMP     R0,R1
1769  006264  001401                           BEQ     5$
1770  006266  104003                           HLT     3
1771  006270                          5$:
1772  006270  052777  000001  010666           BIS     #BREAK,@TXCSR   ;SET BREAK
1773  006276  012700  000000                   MOV     #0,R0           ;EXPECTED
1774  006302  005367  172634                   DEC     COUNT
1775  006306  001332                           BNE     1$
1776  006310  104400                           SCOPE
1777
```

```
1778                                    ;;THIS TEST VERIFYS THAT DSC CAUSES AN INTERRUPT
1779                                    ;;THIS TEST ONLY WORKS IN MAINT EXTERNAL MODE
1780                                    ;;INTERRUPT VECTOR:  DURIV
1781  006312  012767  000013  172606  TST11:  MOV     #11,TSTNO       ;SAVE THIS
1782  006320  012767  006532  172570          MOV     #TST12,NEXT             ;GO TO THIS TEST WHEN THRU
1783  006326  105767  172651                  TSTB    JMRBY           ;IN MAINT EXTERNAL?
1784  006332  100076                          BPL     7$              ;IF ANSWER NO JUMP AROUND TEST
1785  006334  052777  000400  010622          BIS     #MRESET,@TXCSR  ;MASTER RESET
1786  006342  105767  172633                  TSTB    OPTCLR  ;IS THE OPTIONAL CLR JUMPER IN ?
1787  006346  100405                          BMI     1$      ;YES
1788  006350  012777  000000  010572          MOV     #0,@RXCSR       ;CLR THE UNRESETTABLE BITS
1789  006356  005777  010566                  TST     @RXCSR  ;GET RID OF DSC BY READING RXCSR
1790  006362  012777  006406  010604  1$:     MOV     #2$,@DURIV      ;SET UP TRAPCATCHER
1791  006370  016777  010120  010600          MOV     DUPRT,@DURIS    ;
1792  006376  016767  010114  171372          MOV     LESS1,PS        ;ALLOW INTERRUPTS
1793  006404  000423                          BR      4$      ;JUMP AROUND INTERRUPT SVC ROUTINE
1794                                  ;THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
1795  006406  012767  000340  171362  2$:     MOV     #LEVEL7,PS      ;DON'T ALLOW ANYMORE INTERRUPTS
1796  006414  005777  010530                  TST     @RXCSR          ;DSC=1?
1797  006420  100401                          BMI     3$
1798  006422  104000                          HLT                     ;FALSE INTERRUPT
1799  006424                          3$:
1800  006424  042777  000040  010516          BIC     #DSINTE,@RXCSR  ;CLEAR INTERRUPT ENABLE
1801  006432  012716  006522                  MOV     #6$,(SP)        ;SET UP RETURN LOCATION
1802  006436  016777  010534  010530          MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
1803  006444  012777  000000  010524          MOV     #0,@DURIS       ;
1804  006452  000002                          RTI
1805
1806  006454  052777  000040  010466  4$:     BIS     #DSINTE,@RXCSR  ;SET INTERRUPT ENABLE
1807  006462  052777  000002  010460          BIS     #DTR,@RXCSR     ;TRY TO CAUSE INTERRUPT
1808  006470  005000                          CLR     R0
1809  006472                          5$:
1810  006472  005200                          INC     R0              ;WAIT FOR INTERRUPT
1811  006474  001376                          BNE     5$
1812  006476  016777  010474  010470          MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
1813  006504  012777  000000  010464          MOV     #0,@DURIS       ;
1814
1815  006512  042777  000040  010430          BIC     #DSINTE,@RXCSR  ;CLEAR INTERRUPT ENABLE
1816  006520  104000                          HLT                     ;INTERRUPT FAILED TO OCCUR
1817  006522  012767  000340  171246  6$:     MOV     #LEVEL7,PS
1818  006530  104400                  7$:     SCOPE
1819
```

```
 1820                                   ;;THIS TEST VERIFYS THAT RXDONE CAUSES AN INTERRUPT
 1821                                   ;;MODE:  SYNC EXTERNAL
 1822                                   ;;INTERRUPT VECTOR:  DURIV
 1823                                   ;;LENGTH:  EIGHT
 1824
 1825  006532  012767  000014  172366  TST12:  MOV     #12,TSTNO       ;SAVE THIS
 1826  006540  012767  007100  172350           MOV     #TST13,NEXT             ;GO TO THIS TEST WHEN THRU
 1827
 1828  006546  052777  000400  010410           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1829  006554  012777  020000  010376           MOV     #SYNEXT,@PARCSR ;SET THE MODE
 1830  006562  052777  000400  010374           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1831
 1832                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1833  006570  012777  064001  010366           MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1834
 1835                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1836  006576  012777  026026  010354           MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
 1837  006604  052777  000020  010336           BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
 1838                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1839  006612  042777  020000  010344           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1840  006620  052777  020000  010336           BIS     #CLK,@TXCSR     ;POKE CLK UP
 1841  006626  012777  006652  010340           MOV     #1$,@DURIV      ;SET UP TRAPCATCHER
 1842  006634  016777  007654  010334           MOV     DUPRT,@DURIS    ;
 1843  006642  016767  007650  171126           MOV     LESS1,PS        ;ALLOW INTERRUPTS
 1844  006650  000425                           BR      3$      ;JUMP AROUND INTERRUPT SVC ROUTINE
 1845                                   ;THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
 1846  006652  012767  000340  171116  1$:      MOV     #LEVEL7,PS      ;DON'T ALLOW ANYMORE INTERRUPTS
 1847  006660  042777  000100  010262           BIC     #RINTEN,@RXCSR  ;CLEAR INTERRUPT ENABLE
 1848  006666  105777  010256                   TSTB    @RXCSR          ;RXDONE=1?
 1849  006672  100401                           BMI     2$
 1850  006674  104000                           HLT                     ;FALSE INTERRUPT
 1851  006676                           2$:
 1852  006676  012716  007070                   MOV     #8$,(SP)        ;SET UP RETURN LOCATION
 1853  006702  016777  010270  010264           MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
 1854  006710  012777  000000  010260           MOV     #0,@DURIS       ;
 1855  006716  017701  010232                   MOV     @RXDBUF,R1      ;CLEAR INTERRUPT
 1856  006722  000002                           RTI
 1857
 1858  006724  052777  000100  010216  3$:      BIS     #RINTEN,@RXCSR  ;SET INTERRUPT ENABLE
 1859  006732  012767  000010  172200           MOV     #8.,SHIFT       ;# OF SHIFTS
 1860  006740  012767  000025  172176           MOV     #25,TEMP1       ;TO BE SHIFTED CHARACTER
 1861                                   ;THE FOLLOWING POKES THE MAINT DATA BASED UPON THE
 1862                                   ;INFORMATION CONTAINED IN TEMP1 AND IT IS
 1863                                   ;SHIFTED IN BY THE CONTENTS OF SHIFT
 1864  006746  042777  040000  010210  4$:      BIC     #MTDATA,@TXCSR
 1865  006754  000241                           CLC
 1866  006756  006067  172162                   ROR     TEMP1   ;FORCE CARRY
 1867  006762  103003                           BCC     5$
 1868  006764  052777  040000  010172           BIS     #MTDATA,@TXCSR
 1869  006772  042777  020000  010164  5$:      BIC     #CLK,@TXCSR
 1870  007000  052777  020000  010156           BIS     #CLK,@TXCSR
 1871  007006  005367  172126                   DEC     SHIFT
 1872  007012  001355                           BNE     4$
 1873                                   ;INTERRUPT SHOULD NOW OCCUR
 1874  007014  005000                           CLR     R0
 1875  007016                           6$:
```

```
1876   007016   005200                          INC     R0              ;WAIT FOR INTERRUPT
1877   007020   001376                          BNE     6$
1878   007022   016777   010150   010144        MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
1879   007030   012777   000000   010140        MOV     #0,@DURIS       ;
1880   007036   016703   010112                MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
1881   007042   012700   000025                MOV     #25,R0          ;EXPECTED
1882   007046   017701   010102                MOV     @RXDBUF,R1
1883   007052   042777   000100   010070        BIC     #RINTEN,@RXCSR  ;CLR INTR ENABLE
1884   007060   020001                          CMP     R0,R1
1885   007062   001401                          BEQ     7$
1886   007064   104002                          HLT     2               ;CHARACTERS SHOULD COMPARE
1887   007066                           7$:
1888   007066   104000                          HLT                     ;INTERRUPT FAILED TO OCCUR
1889
1890   007070   012767   000340   170700  8$:    MOV     #LEVEL7,PS
1891   007076   104400                          SCOPE
1892
```

```
 1893                                   ;;THIS TEST VERIFYS THAT TWO INTERRUPTS THAT TRAP TO
 1894                                   ;;THE SAME VECTOR ARE BOTH EXECUTED
 1895                                   ;;INTERRUPT VECTOR:  DURIV
 1896                                   ;;THIS TEST ONLY WORKS IN MAINT EXTERNAL MODE
 1897
 1898  007100  012767  000015  172020  TST13:  MOV     #13,TSTNO       ;SAVE THIS
 1899  007106  012767  007556  172002           MOV     #TST14,NEXT             ;GO TO THIS TEST WHEN THRU
 1900  007114  105767  172063                   TSTB    JMRBY           ;IN MAINT. EXTERNAL?
 1901  007120  100402                           BMI     1$      ;IF ANSWER WAS YES DO THIS TEST
 1902  007122  000167  000426                   JMP     15$     ;IF ANSWER WAS NO JUMP AROUND TEST
 1903  007126                           1$:
 1904  007126  105767  172C47                   TSTB    OPTCLR  ;IS THE OPTIONAL CLEAR JUMPER IN ?
 1905  007132  100402                           BMI     2$      ;YES
 1906  007134  005077  010010                   CLR     @RXCSR  ;NO CLEAR UNRESETTABLE BITS
 1907  007140                           2$:
 1908  007140  052777  000400  010016           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1909  007146  012777  020000  010004           MOV     #SYNEXT,@PARCSR ;SET THE MODE
 1910  007154  052777  000400  010002           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1911
 1912                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1913  007162  012777  064001  007774           MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1914
 1915                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1916  007170  012777  026026  007762           MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
 1917  007176  052777  000020  007744           BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
 1918                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1919  007204  042777  020000  007752           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1920  007212  052777  020000  007744           BIS     #CLK,@TXCSR     ;POKE CLK UP
 1921  007220  012777  007244  007746           MOV     #3$,@DURIV      ;SET UP TRAPCATCHER
 1922  007226  016777  007262  007742           MOV     DUPRT,@DURIS    ;
 1923  007234  016767  007256  170534           MOV     LESS1,PS        ;ALLOW INTERRUPT
 1924  007242  000457                           BR      9$      ;JUMP AROUND SVC ROUTINE
 1925
 1926                                   ;THE FOLLOWING IS THE 1ST INTERRUPT SVC ROUTINE
 1927  007244  012767  000340  170524  3$:      MOV     #LEVEL7,PS      ;DON'T ALLOW ANY MORE INTERRUPTS
 1928  007252  105777  007672                   TSTB    @RXCSR          ;RXDONE = 1 ?
 1929  007256  100401                           BMI     4$
 1930  007260  104000                           HLT                     ;FALSE INTERRUPT
 1931  007262                           4$:
 1932  007262  012716  007546                   MOV     #14$,(SP)       ;SET UP RETURN LOCATION
 1933  007266  012777  007350  007700           MOV     #7$,@DURIV      ;SET UP TRAPCATCHER FOR SECOND
 1934                                                                   ;INTERRUPT
 1935  007274  052777  000002  007646           BIS     #DTR,@RXCSR     ;TRY TO CAUSE SECOND INTERRUPT
 1936  007302  017701  007646                   MOV     @RXDBUF,R1      ;JUST READ RXDBUF TO CLR RXDONE
 1937                                                                   ;TO ALLOW SECOND INTERRUPT
 1938  007306  016767  007204  170462           MOV     LESS1,PS        ;ALLOW INTERRUPT
 1939  007314  005000                           CLR     R0
 1940  007316                           5$:
 1941  007316  005200                           INC     R0              ;WAIT FOR INTERRUPT
 1942  007320  001376                           BNE     5$
 1943  007322  042777  000140  007620           BIC     #RINTEN!DSINTE,@RXCSR   ;CLR INTR ENABLES
 1944  007330  104000                           HLT                     ;2ND INTERRUPT FAILED TO OCCUR
 1945
 1946  007332  016777  007640  007634  6$:      MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
 1947  007340  012777  000000  007630           MOV     #0,@DURIS       ;
 1948  007346  000002                           RTI
```

```
                                        ;THE FOLLOWING IS THE 2ND INTERRUPT SVC ROUTINE
007350  012767  000340  170420  7$:     MOV     #LEVEL7,PS      ;DON'T ALLOW ANYMORE INTERRUPTS
007356  005777  007566                  TST     @RXCSR          ;DSC = 1 ?
007362  100401                          BMI     8$
007364  104000                          HLT                     ;FALSE INTERRUPT
007366                          8$:
007366  042777  000140  007554          BIC     #RINTEN!DSINTE,@RXCSR   ;CLR BOTH INTR ENABLES
007374  012716  007332                  MOV     #6$,(SP)        ;SET UP RETURN LOCATION
007400  000002                          RTI
007402  052777  000140  007540  9$:     BIS     #RINTEN!DSINTE,@RXCSR ;SET INTERRUPT ENABLES
007410  012767  000010  171522          MOV     #8.,SHIFT       ;# OF SHIFTS
007416  012767  000025  171520          MOV     #25,TEMP1
                                        ;THE FOLLOWING POKES THE MAINT DATA BASED UPON THE
                                        ;INFORMATION CONTAINED IN TEMP1 AND IT IS
                                        ;SHIFTED IN BY THE CONTENTS OF SHIFT
007424  042777  040000  007532  10$:    BIC     #MTDATA,@TXCSR
007432  000241                          CLC
007434  006067  171504                  ROR     TEMP1   ;FORCE CARRY
007440  103003                          BCC     11$
007442  052777  040000  007514          BIS     #MTDATA,@TXCSR
007450  042777  020000  007506  11$:    BIC     #CLK,@TXCSR
007456  052777  020000  007500          BIS     #CLK,@TXCSR
007464  005367  171450                  DEC     SHIFT
007470  001355                          BNE     10$
                                        ;1ST INTERRUPT SHOULD NOW OCCUR
007472  005000                          CLR     R0
007474                          12$:
007474  005200                          INC     R0              ;WAIT FOR INTERRUPT
007476  001376                          BNE     12$             ;
007500  016777  007472  007466          MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
007506  012777  000000  007462          MOV     #0,@DURIS       ;
007514  016703  007434                  MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
007520  012700  000025                  MOV     #25,R0          ;EXPECTED
007524  017701  007424                  MOV     @RXDBUF,R1
007530  042777  000140  007412          BIC     #RINTEN!DSINTE,@RXCSR   ;CLR INTERRUPT ENABLES
007536  020001                          CMP     R0,R1
007540  001401                          BEQ     13$
007542  104002                          HLT     2               ;CHARACTERS SHOULD COMPARE
007544                          13$:
007544  104000                          HLT                     ;INTERRUPT FAILED TO OCCUR

007546  012767  000340  170222  14$:    MOV     #LEVEL7,PS      ;DON'T ALLOW ANY MORE INTERRUPTS
007554  104400                  15$:    SCOPE
```

```
 1995                                   ;;THIS TEST VERIFYS THAT DNA CAUSES AN INTERRUPT
 1996                                   ;;MODE:  SYNC EXTERNAL
 1997                                   ;;INTERRUPT VECTOR:  DUTIV
 1998
 1999  007556  012767  000016  171342  TST14:  MOV     #14,TSTNO       ;SAVE THIS
 2000  007564  012767  010052  171324          MOV     #TST15,NEXT             ;GO TO THIS TEST WHEN THRU
 2001
 2002  007572  052777  000400  007364          BIS     #MRESET,@TXCSR  ;MASTER RESET
 2003  007600  012777  020000  007352          MOV     #SYNEXT,@PARCSR ;SET THE MODE
 2004  007606  052777  000400  007350          BIS     #MRESET,@TXCSR  ;MASTER RESET
 2005
 2006                                   ;SET "AINTENANCE MODE & SEND
 2007                                           ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
 2008  007614  012777  004020  007342          MOV     #MINT!SEND,@TXCSR
 2009
 2010                                   ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
 2011  007622  012777  026026  007330          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
 2012  007630  112777  000025  007332          MOVB    #25,@TXDBUF     ;LOAD CHARACTER
 2013  007636  012767  000010  171274          MOV     #8.,SHIFT
 2014                                   ;POKE CLK TO GET INTO SYNCRONIZATION
 2015  007644  052777  020000  007312          BIS     #CLK,@TXCSR     ;POKE CLK UP
 2016  007652  042777  020000  007304          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 2017
 2018  007660                                 1$:
 2019  007660  052777  020000  007276          BIS     #CLK,@TXCSR     ;POKE CLK UP
 2020  007666  042777  020000  007270          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 2021  007674  005367  171240                  DEC     SHIFT           ;LAST SHIFT?
 2022  007700  001367                          BNE     1$
 2023  007702  012777  007752  007270          MOV     #3$,@DUTIV      ;SET UP TRAPCATCHER
 2024  007710  016777  006600  007264          MOV     DUPRT,@DUTIS    ;
 2025  007716  016767  006574  170052          MOV     LESS1,PS        ;ALLOW INTERUPTS
 2026  007724  052777  000040  007232          BIS     #DNAINTE,@TXCSR ;ENABLE INTERRUPT
 2027                                           ;NOW POKE CLK TO GET DNA
 2028  007732  052777  020000  007224          BIS     #CLK,@TXCSR     ;POKE CLK
 2029  007740  005000                          CLR     R0
 2030  007742                                 2$:
 2031  007742  005200                          INC     R0              ;WAIT FOR INTERRUPT
 2032  007744  001376                          BNE     2$
 2033  007746  104000                          HLT                     ;INTERRUPT FAILED TO OCCUR
 2034  007750  000423                          BR      5$      ;JUMP AROUND SVC ROUTINE
 2035                                   ;THE FOLLOWING IS THE INTERRUPT SERVICE ROUTINE
 2036  007752  012767  000340  170016  3$:     MOV     #LEVEL7,PS      ;DON'T ALLOW ANYMORE INTERRUPTS
 2037  007760  005777  007200                  TST     @TXCSR          ;DNA?
 2038  007764  100401                          BMI     4$
 2039  007766  104000                          HLT                     ;FALSE INTERRUPT
 2040  007770                                 4$:
 2041  007770  042777  000040  007166          BIC     #DNAINTE,@TXCSR ;CLR INTR ENABLE
 2042  007776  012716  010042                  MOV     #6$,(SP)        ;SET UP RETURN LOCATION
 2043  010002  016777  007174  007170          MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
 2044  010010  012777  000000  007164          MOV     #0,@DUTIS       ;
 2045  010016  000002                          RTI
 2046
 2047  010020  016777  007156  007152  5$:     MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
 2048  010026  012777  000000  007146          MOV     #0,@DUTIS       ;
 2049
 2050  010034  042777  000040  007122          BIC     #DNAINTE,@TXCSR ;CLR INTERRUPT ENABLE
```

H04

SEQ 0046

```
 2051  010042  012767  000340  167726  6$:     MOV     #LEVEL7,PS      ;RESTORE NO INTERRUPT STATUS
 2052  010050  104400                          SCOPE
 2053
```

```
2054                                    ;;THIS TEST VERIFYS THAT TXDONE CAUSES AN INTERRUPT
2055                                    ;;INTERRUPT VECTOR:  DUTIV
2056                                    ;;NOTE: TXDONE = 1          AFTER A MASTER RESET
2057                                    ;;
2058  010052  012767  000017  171046  TST15:  MOV     #15,TSTNO       ;SAVE THIS
2059  010060  012767  010236  171030         MOV     #TST16,NEXT             ;GO TO THIS TEST WHEN THRU
2060
2061  010066  052777  000400  007070         BIS     #MRESET,@TXCSR  ;MASTER RESET
2062  010074  012777  010144  007076         MOV     #2$,@DUTIV      ;SET UP TRAPCATCHER
2063  010102  016777  006406  007072         MOV     DUPRT,@DUTIS    ;
2064  010110  016767  006402  167660         MOV     LESS1,PS        ;ALLOW INTERPUTS
2065  010116  052777  000100  007040         BIS     #TXINTE,@TXCSR  ;ENABLE INTERRUPT
2066  010124  005000                         CLR     R0
2067  010126                          1$:
2068  010126  005200                         INC     R0              ;WAIT FOR INTERRUPT
2069  010130  001376                         BNE     1$              ;
2070  010132  042777  000100  007024         BIC     #TXINTE,@TXCSR  ;CLR INTERRUPT ENABLE
2071  010140  104000                         HLT                     ;INTERRUPT FAILED TO OCCUR
2072  010142  000423                         BR      4$      ;JUMP AROUND SVC ROUTINE
2073
2074                                    ;THE FOLLOWING IS THE INTERRUPT SERVICE ROUTINE
2075  010144  012767  000340  167624  2$:    MOV     #LEVEL7,PS      ;DON'T ALLOW ANYMORE INTERRUPTS
2076  010152  042777  000100  007004         BIC     #TXINTE,@TXCSR  ;CLR INTR ENABLE
2077  010160  105777  007000                 TSTB    @TXCSR          ;TXDONE?
2078  010164  100401                         BMI     3$
2079  010166  104000                         HLT                     ;FALSE INTERRUPT
2080  010170                          3$:
2081  010170  012716  010226                 MOV     #5$,(SP)        ;SET UP RETURN LOCATION
2082  010174  016777  007002  006776         MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
2083  010202  012777  000000  006772         MOV     #0,@DUTIS       ;
2084  010210  000002                         RTI
2085
2086  010212  016777  006764  006760  4$:    MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
2087  010220  012777  000000  006754         MOV     #0,@DUTIS       ;
2088
2089  010226  012767  000340  167542  5$:    MOV     #LEVEL7,PS      ;RESTORE NO INTERRUPT STATUS
2090  010234  104400                         SCOPE
2091
```

```
2092                                    ;;THIS TEST VERIFYS THAT TXDONE DOES NOT CAUSE AN INTERRUPT
2093                                    ;;WHEN PROCESSOR PRIORITY LEVEL IS TOO HIGH
2094                                    ;;INTERRUPT VECTOR: DUTIV
2095                                    ;;NOTE: TXDONE = 1 AFTER A MASTER RESET
2096                                    ;;
2097  010236  012767  000020  170662    TST16:  MOV     #16,TSTNO       ;SAVE THIS
2098  010244  012767  010420  170644            MOV     #TST17,NEXT             ;GO TO THIS TEST WHEN THRU
2099  010252  052777  000400  006704            BIS     #MRESET,@TXCSR  ;MASTER RESET
2100  010260  012777  010350  006712            MOV     #2$,@DUTIV      ;SET UP TRAPCATCHER
2101  010266  016777  006222  006706            MOV     DUPRT,@DUTIS    ;
2102  010274  016767  006214  167474            MOV     DUPRT,PS        ;SET PS LEVEL TOO HIGH
2103  010302  052777  000100  006654            BIS     #TXINTE,@TXCSR  ;ENABLE INTERRRUPT
2104  010310  005000                            CLR     R0      ;WAIT FOR INTERRUPT
2105  010312                                1$:
2106  010312  005200                            INC     R0      ;
2107  010314  001376                            BNE     1$      ;
2108  010316  042777  000100  006640            BIC     #TXINTE,@TXCSR  ;CLR INTR ENABLE
2109  010324  012767  000340  167444            MOV     #LEVEL7,PS      ;DON'T ALLOW INTERRUPTS
2110  010332  016777  006644  006640            MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
2111  010340  012777  000000  006634            MOV     #0,@DUTIS       ;
2112  010346  000423                            BR      4$      ;TEST IS OK....GET OUT OF TEST
2113                                  ;THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
2114  010350  012767  000340  167420    2$:     MOV     #LEVEL7,PS      ;DONT ALLOW ANYMORE INTERRUPTS
2115  010356  042777  000100  006600            BIC     #TXINTE,@TXCSR  ;CLR INTR ENABLE
2116  010364  012716  010406                    MOV     #3$,(SP)        ;SET UP RETURN LOCATION
2117                                                                    ;TO REPORT ERROR
2118  010370  016777  006606  006602            MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
2119  010376  012777  000000  006576            MOV     #0,@DUTIS       ;
2120  010404  000002                            RTI
2121                                  ;END OF INTERRUPT SVC ROUTINE
2122
2123
2124
2125                                  ;YOU SHOULD NOT GET INTO THIS FOLLOWING CODE UNLESS THERE
2126                                  ;WAS AN ERROR
2127  010406  012767  000340  167362    3$:     MOV     #LEVEL7,PS      ;DON'T ALLOW ANYMORE INTERRUPTS
2128  010414  104000                            HLT     ;INTERRUPT SHOULD NOT OF OCCURED,CHECK
2129                                                    ;THE INTERRUPT LEVEL SELECTED OR CHECK
2130                                                    ;INTERRUPT LOGIC OR BOTH
2131  010416  104400                    4$:     SCOPE
2132
2133
```

```
                                        ;;THIS TEST VERIFYS THAT TXDONE CAUSES ONLY ONE INTERRUPT
                                        ;;PROVIDING THAT TXCSR IS NOT READ
                                        ;;AND TXDBUF IS NOT LOADED (WRITTEN)
                                        ;;THIS TEST CHECKS THE ONCE ONLY FLIP/FLOP (V2)
                                        ;;OF THE INTERRUPT CONTROL LOGIC
                                        ;;INTERRUPT VECTOR: DUTIV
                                        ;;NOTE: TXDONE = 1    AFTER A MASTER RESET
                                        ;;
010420  012767  000021  170500  TST17:  MOV     #17,TSTNO       ;SAVE THIS
010426  012767  010614  170462          MOV     #TST18,NEXT             ;GO TO THIS TEST WHEN THRU
010434  052777  000400  006522          BIS     #MRESET,@TXCSR  ;MASTER RESET
010442  012777  010504  006530          MOV     #2$,@DUTIV      ;SET UP TRAPCATCHER
010450  016777  006040  006524          MOV     DUPRT,@DUTIS    ;
010456  016767  006034  167312          MOV     LESS1,PS        ;ALLOW INTERRUPTS
010464  052777  000100  006472          BIS     #TXINTE,@TXCSR  ;ENABLE INTR ENABLE
010472  005000                          CLR     R0
010474                          1$:
010474  005200                          INC     R0
010476  001376                          BNE     1$
010500  104000                          HLT     ;INTERRUPT FAILED TO OCCUR
010502  000427                          BR      7$
                                ;THE FOLLOWING IS THE INTR SVC ROUTINE
010504  012767  000340  167264  2$:      MOV     #LEVEL7,PS      ;DON'T ALLOW ANYMORE INTR
010512  012716  010554                  MOV     #5$,(SP)        ;SET UP RETURN LOCATION
010516  012777  010526  006454          MOV     #3$,@DUTIV      ;SET UP TRAPCATCHER TO
                                                                ;PROVE THAT THE INTERRUPT DOES NOT OCCUR
                                                                ;TWICE (AFTER RTI 'ING FROM THIS
                                                                ;SVC ROUTINE
010524  000002                          RTI
                                ;THE FOLLOWING INTERRUPT SVC ROUTINE WILL CATCH THE SECOND INTR
010526  012767  000340  167242  3$:      MOV     #LEVEL7,PS      ;DON'T ALLOW INTER
010534  012716  010562                  MOV     #7$,(SP)        ;SET UP RETURN LOCATION
010540  105777  006420                  TSTB    @TXCSR  ;TXDONE = 1?
010544  100401                          BMI     4$
010546  104000                          HLT     ;TXDONE SHOULD BE SET
010550                          4$:
010550  104000                          HLT     ;THE INTERRUPT WAS TAKEN TWICE.....
                                                ;CHECK OUT THE V2 FLIP/FLOP LOGIC
                                                ;IN THE INTERRUPT CONTROL LOGIC
010552  000002                          RTI
010554  005000                  5$:     CLR     R0      ;ALLOW TIME TO CATCH SECOND
010556                          6$:
010556  005200                          INC     R0              ;IF IT WERE TO OCCUR
010560  001376                          BNE     6$      ;
010562  016777  006414  006410  7$:      MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
010570  012777  000000  006404          MOV     #0,@DUTIS       ;
010576  042777  000100  006360          BIC     #TXINTE,@TXCSR  ;CLR INTERRUPT ENABLE
010604  012767  000340  167164          MOV     #LEVEL7,PS      ;RESTORE NO INTERRUPT STATUS
010612  104400                          SCOPE
```

```
                                        ;;THIS TEST VERIFYS THAT TWO INTERRUPTS THAT TRAP
                                        ;;TO THE SAME VECTOR ARE BOTH EXECUTED
                                        ;;INTERRUPT VECTOR:  DUTIV
                                        ;;MODE:  SYNC EXTERNAL

010614  012767  000022  170304  TST18:  MOV     #18,TSTNO       ;SAVE THIS
010622  012767  011204  170266           MOV     #TST19,NEXT             ;GO TO THIS TEST WHEN THRU

010630  052777  000400  006326           BIS     #MRESET,@TXCSR  ;MASTER RESET
010636  012777  020000  006314           MOV     #SYNEXT,@PARCSR ;SET THE MODE
010644  052777  000400  006312           BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINTENANCE MODE & SEND
                                                ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
010652  012777  004020  006304           MOV     #MINT!SEND,@TXCSR

                                        ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
010660  012777  026026  006272           MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
010666  112777  000025  006274           MOVB    #25,@TXDBUF     ;LOAD CHARACTER
010674  012767  000010  170236           MOV     #8.,SHIFT
                                        ;POKE CLK TO GET INTO SYNCRONIZATION
010702  052777  020000  006254           BIS     #CLK,@TXCSR     ;POKE CLK UP
010710  042777  020000  006246           BIC     #CLK,@TXCSR     ;POKE CLK DOWN

010716                                  1$:
010716  052777  020000  006240           BIS     #CLK,@TXCSR     ;POKE CLK UP
010724  042777  020000  006232           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010732  005367  170202                   DEC     SHIFT           ;LAST SHIFT?
010736  001367                           BNE     1$
010740  012777  011002  006232           MOV     #3$,@DUTIV      ;SET UP TRAPCATCHER
010746  016777  005542  006226           MOV     DUPRT,@DUTIS    ;
010754  016767  005536  167014           MOV     LESS1,PS        ;ALLOW INTERRUPTS
010762  052777  000140  006174           BIS     #TXINTE!DNAINTE,@TXCSR  ;ENABLE INTERRUPTS
010770  005000                           CLR     R0
010772                                  2$:
010772  005200                           INC     R0              ;WAIT FOR INTERRUPT
010774  001376                           BNE     2$              ;
010776  104000                           HLT                     ;INTERRUPT FAILED TO OCCUR
011000  000464                           BR      10$     ;JUMP AROUND SVC ROUTINES

                                        ;THE FOLLOWING IS THE 1ST INTERRUPT SVC ROUTINE
011002  012767  000340  166766  3$:      MOV     #LEVEL7,PS      ;DON'T ALLOW ANYMORE INTERRUPTS
011010  005777  006150                   TST     @TXCSR  ;DNA=0 ?
011014  100001                           BPL     4$
011016  104000                           HLT     ;DNA SHOULD NOT BE ASSERTED
011020                                  4$:
011020  105777  006140                   TSTB    @TXCSR  ;TXDONE = 1?
011024  100401                           BMI     5$
011026  104000                           HLT             ;FALSE INTERRUPT
011030                                  5$:
011030  012716  011174                   MOV     #11$,(SP)       ;SET UP RETURN LOCATION
011034  012777  011120  006136           MOV     #8$,@DUTIV      ;SET UP TRAPCATCHER
                                        ;NOW POKE CLK TO BRING UP DNA
011042  052777  020000  006114           BIS     #CLK,@TXCSR     ;POKE CLK
011050  112777  000025  006112           MOVB    #25,@TXDBUF     ;JUST LOAD ANY CHAR TO CLR
                                                ;TXDONE TO ALLOW SECOND INTERRUPT
```

```
2240  011056  016767  005434  166712          MOV     LESS1,PS        ;ALLOW INTERRUPTS
2241  011064  005000                          CLR     R0
2242  011066                          6$:
2243  011066  005200                          INC     R0              ;WAIT FOR INTERRUPT
2244  011070  001376                          BNE     6$
2245  011072  042777  000140  006064          BIC     #DNAINTE!TXINTE,@TXCSR  ;CLR INTR ENABLES
2246  011100  104000                          HLT                     ;2ND INTERRUPT FAILED TO OCCUR
2247
2248  011102  016777  006074  006070  7$:     MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
2249  011110  012777  000000  006064          MOV     #0,@DUTIS       ;
2250  011116  000002                          RTI
2251
2252                                  ;THE FOLLOWING IS THE 2ND INTERRUPT SVC ROUTINE
2253  011120  012767  000340  166650  8$:     MOV     #LEVEL7,PS
2254  011126  005777  006032                  TST     @TXCSR          ;DNA
2255  011132  100401                          BMI     9$
2256  011134  104000                          HLT                     ;FALSE INTERRUPT
2257  011136                          9$:
2258  011136  042777  000140  006020          BIC     #DNAINTE!TXINTE,@TXCSR  ;CLR BOTH INTR ENABLES
2259  011144  012716  011102                  MOV     #7$,(SP)                ;SETUP RETURN LOCATION
2260  011150  000002                          RTI
2261
2262  011152  016777  005024  006020  10$:    MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
2263  011160  012777  000000  006014          MOV     #0,@DUTIS
2264
2265  011166  042777  000140  005770          BIC     #DNAINTE!TXINTE,@TXCSR ;CLR BOTH INTERRUPT
2266                                                                  ;ENABLES
2267  011174  012767  000340  166574  11$:     MOV     #LEVEL7,PS      ;RESTORE NO INTERRUPT STATUS
2268
2269  011202  104400                          SCOPE
2270
```

```
2271                                    ;;THIS TEST VERIFYS CTP MODE (IE SYSTST MODE)
2272                                    ;;IT BASICALLY CHECKS THE EXISTANCE OF
2273                                    ;;THE FREE RUNNING OSCILLATOR
2274                                    ;;MODE:  ISOCRONOUS (ISYMOD)
2275                                    ;;LENGTH:  EIGHT
2276                                    ;;THIS TEST USES BOTH THE RECEIVER & TRANSMITTER LOGIC
2277
2278  011204  012767  000023  167714  TST19:  MOV     #19,TSTNO       ;SAVE THIS
2279  011212  012767  011512  167676           MOV     #TST20,NEXT             ;GO TO THIS TEST WHEN THRU
2280  011220  052777  000400  005736           BIS     #MRESET,@TXCSR  ;MASTER RESET
2281  011226  012777  000000  005724           MOV     #ISYMOD,@PARCSR ;LOAD THE MODE
2282  011234  052777  000400  005722           BIS     #MRESET,@TXCSR  ;MASTER RESET
2283  011242  012777  006026  005710           MOV     #ISYMOD!EIGHT!NOPAR!26,@PARCSR  ;LOAD THE MODE,
2284                                           ;# OF BITS PER CHAR,PARITY SENSE(NO PARITY),
2285                                           ;&SYNC CHARACTER (26)
2286  011250  112777  000025  005712           MOVB    #25,@TXDBUF     ;LOAD THE CHAR
2287  011256  012777  011356  005710           MOV     #3$,@DURIV      ;SET UP TRAPCATCHER
2288  011264  016777  005224  005704           MOV     DUPRT,@DURIS    ;
2289  011272  016767  005220  166476           MOV     LESS1,PS        ;ALLOW INTERRUPTS
2290  011300  016703  005650                   MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2291  011304  012700  000025                   MOV     #25,R0          ;EXPECTED
2292  011310  012777  014020  005646           MOV     #SYSTST!SEND,@TXCSR     ;OK  NOW LOAD SEND &
2293                                                                   ;MAINT. MODE
2294  011316  052777  000120  005624           BIS     #SYNSCH!RINTEN,@RXCSR   ;SET SEARCH SYNC &
2295                                                                   ;RECEIVER INTERRUPT
2296                                                                   ;ENABLE & WAIT FOR INTERRUPT
2297  011324  005067  167624                   CLR     TEMP5
2298  011330  005002                  1$:      CLR     R2
2299  011332                           2$:
2300  011332  005202                           INC     R2              ;WAIT FOR INTERRUPT
2301  011334  001376                           BNE     2$
2302  011336  005267  167612                   INC     TEMP5
2303  011342  022767  000003  167604           CMP     #3,TEMP5
2304  011350  002367                           BGE     1$
2305  011352  104000                           HLT                     ;INTERRUPT DID NOT OCCUR
2306  011354  000423                           BR      4$
2307
2308                                   ;THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
2309  011356  012767  000340  166412  3$:      MOV     #LEVEL7,PS      ;PREVENT INTERRUPTS
2310  011364  017704  005560                   MOV     @RXCSR,R4       ;SAVE
2311  011370  017701  005560                   MOV     @RXDBUF,R1      ;ACTUAL
2312  011374  016777  005576  005572           MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
2313  011402  012777  000000  005566           MOV     #0,@DURIS       ;
2314  011410  012716  011456                   MOV     #5$,(SP)        ;SET UP RETURN
2315  011414  042777  000100  005526           BIC     #RINTEN,@RXCSR  ;CLR INTERRUPT ENABLE
2316  011422  000002                           RTI
2317
2318  011424  042777  000100  005516  4$:      BIC     #RINTEN,@RXCSR  ;CLR INTERRUPT ENABLE
2319  011432  012767  000340  166336           MOV     #LEVEL7,PS      ;PREVENT INTERRUPTS
2320  011440  016777  005532  005526           MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
2321  011446  012777  00000C  005522           MOV     #0,@DURIS       ;
2322  011454  000415                           BR      7$
2323
2324  011456  020001                   5$:     CMP     R0,R1
2325  011460  001401                           BEQ     6$
2326  011462  104002                           HLT     2               ;CHARACTERS DID NOT MATCH
```

```
 2327  011464                       6$:
 2328  011464  016703  005460               MOV     RXCSR,R3        ;SETUP FOR ERROR MESSAGE
 2329  011470  012700  000200               MOV     #200,R0         ;EXPECTED
 2330  011474  010401                       MOV     R4,R1           ;ACTUAL
 2331  011476  042701  177577               BIC     #177577,R1      ;SAVE ONLY RXDONE
 2332  011502  020001                       CMP     R0,R1
 2333  011504  001401                       BEQ     .$
 2334  011506  104001                       HLT     1       ;FALSE INTERRUPT
 2335  011510                       7$:
 2336  011510  104400                       SCOPE
```

```
2337                                    ;;THIS TEST VERIFYS CTP MODE (IE SYSTST MODE)
2338                                    ;;IT BASICALLY CHECKS THE EXISTANCE OF
2339                                    ;;THE FREE RUNNING OSCILLATOR
2340                                    ;;MODE: SYNINT
2341                                    ;;LENGTH: EIGHT
2342                                    ;;THIS TEST USES BOTH THE RECEIVER & TRANSMITTER LOGIC
2343                                    ;;
2344  011512  012767  000024  167406    TST20:  MOV     #20,TSTNO       ;SAVE THIS
2345  011520  012767  012354  167370            MOV     #.EOP,NEXT              ;GO TO THIS TEST WHEN THRU
2346  011526  052777  000400  005430            BIS     #MRESET,@TXCSR  ;MASTER RESET
2347  011534  012777  030000  005416            MOV     #SYNINT,@PARCSR ;SET THE MODE
2348  011542  052777  000400  005414            BIS     #MRESET,@TXCSR  ;MASTER RESET
2349
2350                                    ;SET MAINTENANCE MODE & SEND
2351                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2352  011550  012777  014020  005406            MOV     #SYSTST!SEND,@TXCSR
2353
2354                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2355  011556  012777  036026  005374            MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
2356  011564  052777  000420  005356            BIS     #SYNSCH!STPSYN,@RXCSR   ;SET SEARCH SYNC &
2357                                            ;STRIP SYNC SO THAT RXDONE ASSERTS
2358                                            ;WHEN CHAR "25" ARRIVES AND NOT BEFORE...
2359                                            ; ...THEREFORE,SET STRIP SYNC
2360                                    ;.............WAIT FOR SYNSCH TO BE
2361                                    ;CLOCKED IN BY SYSTST CLK
2362  011572  005067  167356                    CLR     TEMP5
2363  011576                            1$:
2364  011576  005002                    2$:     CLR     R2
2365  011600                                    INC     R2      ;WAIT
2366  011600  005202                            BNE     2$
2367  011602  001376                            INC     TEMP5
2368  011604  005267  167344                    CMP     #3,TEMP5
2369  011610  022767  000003  167336            BGE     1$      ;GO BACK TO CLR R2 AND WAIT SOME MORE
2370  011616  002367                            MOV     #7$,@DURIV      ;SET UP TRAPCATCHER
2371  011620  012777  012040  005346            MOV     DUPRT,@DURIS    ;
2372  011626  016777  004662  005342            MOV     #8$,@DUTIV      ;
2373  011634  012777  012134  005336            MOV     DUPRT,@DUTIS    ;
2374  011642  016777  004646  005332            MOV     LESS1,PS        ;ALLOW INTERRUPTS
2375  011650  016767  004642  166120            MOV     RXDBUF,R3       ;SET UP FOR ERROR MSG
2376  011656  016703  005272                    MOV     #25,R0  ;EXPECTED CHAR
2377  011662  012700  000025                    MOV     #2,COUNT        ;# OF SYNC CHARS TO GET INTO
2378  011666  012767  000002  167246           ;SYNCRONIZATION
2379
2380  011674  105767  167276            TSTB    SYNCNO  ;TEST TO SEE HOW MANY SYNC CHARS NEEDED
2381  011700  100402                    BMI     3$
2382  011702  005367  167234            DEC     COUNT   ;MAKE IT ONE LESS
2383  011706  052777  000100  005234    3$:     BIS     #RINTEN,@RXCSR  ;SET INTERRUPT ENABLES
2384  011714  052777  000100  005242            BIS     #TXINTE,@TXCSR  ;
2385  011722  000167  000012                    JMP     5$      ;THE FIRST XMIT INTERRUPT SHOULD COME
2386                                            ;FROM TXDONE = 1 AFTER A MASTER RESET
2387  011726  112777  000026  005234    4$:     MOVB    #26,@TXDBUF     ;LOAD SYNC CHAR
2388  011734  005067  167214                    CLR     TEMP5
2389  011740  005002                    5$:     CLR     R2      ;WAIT FOR INTERRUPT
2390  011742                            6$:
2391  011742  005202                            INC     R2
2392  011744  001376                            BNE     6$      ;
```

```
2393  011746  005267  167202          INC     TEMP5
2394  011752  022767  000003  167174  CMP     #3,TEMP5
2395  011760  002367                  BGE     5$
2396  011762  012767  000340  166006  MOV     #LEVEL7,PS      ;PREVENT INTERRUPTS
2397  011770  042777  000100  005166  BIC     #TXINTE,@TXCSR  ;CLR INTR ENABLES
2398  011776  042777  000100  005144  BIC     #RINTEN,@RXCSR  ;
2399  012004  016777  005166  005162  MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
2400  012012  012777  000000  005156  MOV     #0,@DURIS       ;
2401  012020  016777  005156  005152  MOV     DUTIS,@DUTIV    ;
2402  012026  012777  000000  005146  MOV     #0,@DUTIS       ;
2403  012034  104000                  HLT     ;TXDONE INTERRUPT FAILED TO OCCUR
2404                                          ;WATCH OUT HERE::: THIS FAILURE MAY
2405                                          ;ALSO BE CAUSED BY TRANSMIT DATA NOT
2406                                          ;BEING CLOCKED OUT.  I.E.  TXDONE
2407                                          ;NOT RE-ASCERTING SO THAT THE 2ND
2408                                          ;SYNC CHARACTER CAN BE LOADED
2409
2410  012036  000542                  BR      17$     ;GET OUT OF THE TEST
2411
2412                          ;THE FOLLOWING IS THE RECEIVER INTERRUPT SVC ROUTINE
2413  012040  012767  000340  165730  7$:     MOV     #LEVEL7,PS      ;PREVENT INTERRUPTS
2414  012046  017704  005076          MOV     @RXCSR,R4       ;SAVE
2415  012052  017701  005076          MOV     @RXDBUF,R1      ;ACTUAL
2416  012056  016777  005114  005110  MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
2417  012064  012777  000000  005104  MOV     #0,@DURIS       ;
2418  012072  016777  005104  005100  MOV     DUTIS,@DUTIV    ;
2419  012100  012777  000000  005074  MOV     #0,@DUTIS       ;
2420  012106  012716  012270          MOV     #13$,(SP)       ;SET UP RETURN LOCATION
2421  012112  042777  000100  005030  BIC     #RINTEN,@RXCSR  ;CLR INTERRUPT ENABLES
2422  012120  042777  000100  005036  BIC     #TXINTE,@TXCSR  ;
2423  012126  016705  167010          MOV     COUNT,R5        ;SAVE COUNT
2424  012132  000002                  RTI
2425                          ;END OF RECEIVER INTERRUPT SVC ROUTINE
2426                          ;....THE FOLLOWING IS THE XMITTER INTERRUPT SVC ROUTINE
2427  012134  005367  167002  8$:     DEC     COUNT
2428  012140  100403                  BMI     9$
2429  012142  012716  011726          MOV     #4$,(SP)        ;SET UP RETURN LOCATION
2430                                                  ;(LOAD SYNC CHARACTER AGAIN)
2431  012146  000002                  RTI
2432  012150  012716  012156  9$:     MOV     #10$,(SP)       ;SET UP RETURN LOCATION
2433  012154  000002                  RTI
2434                          ;END OF XMITTER INTERRUPT SVC ROUTINE
2435  012156  112777  000025  005004  10$:    MOVB    #25,@TXDBUF     ;LOAD CHARACTER
2436  012164  042777  000100  004772          BIC     #TXINTE,@TXCSR  ;CLR INTR ENABLE
2437  012172  005067  166756          CLR     TEMP5
2438  012176  005002                  11$:    CLR     R2      ;WAIT FOR INTERRUPT(RECEIVER)
2439  012200                          12$:
2440  012200  005202                  INC     R2
2441  012202  001376                  BNE     12$     ;
2442  012204  005267  166744          INC     TEMP5
2443  012210  022767  000003  166736  CMP     #3,TEMP5
2444  012216  002367                  BGE     11$
2445  012220  012767  000340  165550  MOV     #LEVEL7,PS      ;PREVENT INTERRUPTS
2446  012226  042777  000100  004714  BIC     #RINTEN,@RXCSR  ;CLR INTR ENABLE
2447  012234  016777  004736  004732  MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
2448  012242  012777  000000  004725  MOV     #0,@DURIS       ;
```

E05


```
2449  012250  016777  004726  004722          MOV     DUTIS,@DUTIV    ;
2450  012256  012777  000000  004716          MOV     #0,@DUTIS       ;
2451  012264  104000                          HLT     ;RECEIVER INTR FAILED TO OCCUR
2452  012266  000426                          BR      17$     ;GET OUT OF TEST
2453  012270  020001                  13$:    CMP     R0,R1
2454  012272  001401                          BEQ     14$
2455  012274  104002                          HLT     2       ;CHARACTERS DID NOT MATCH
2456  012276                          14$:
2457  012276  016703  004646                  MOV     RXCSR,R3        ;SET UP FOR ERROR MSG
2458  012302  012700  000200                  MOV     #200,R0 ;EXPECTED RXDONE
2459  012306  010401                          MOV     R4,R1   ;ACTUAL
2460  012310  042701  177577                  BIC     #177577,R1      ;SAVE ONLY RXDONE
2461  012314  020001                          CMP     R0,R1
2462  012316  001401                          BEQ     15$
2463  012320  104001                          HLT     1       ;FALSE INTERRUPT
2464  012322                          15$:
2465  012322  020527  177777                  CMP     R5,#-1  ;WAS COUNT =-1   WHEN RECEIVER
2466                                                          ;INTERRUPTED ?
2467  012326  001401                          BEQ     16$
2468  012330  104000                          HLT     ;IF R5 IS GREATER THAN -1.........
2469                                          ;THEN EITHER THE # OF SYNC STRAP IS WRONG
2470                                          ;OR RXDONE IS OCCURING TOO SOON
2471  012332                          16$:
2472  012332  026727  166604  177777          CMP     COUNT,#-1
2473  012340  001401                          BEQ     17$
2474  012342  104000                          HLT     ;IF THIS TEST FAILS,BUT THE ABOVE TEST
2475                                          ;DOESN'T.....IT MAY BE THAT CLEARING
2476                                          ;TXINTE IN THE RECEIVER SVC ROUTINE
2477                                          ;IS NOT STOPPING TXDONE INTERRUPTS
2478  012344                          17$:
2479  012344  012767  000340  165424          MOV     #LEVEL7,PS      ;INHIBIT INTERRUPTS
2480  012352  104400                          SCOPE
```

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

012354  104402                .EOP:   TYPE                            ;TYPE NAME OF TEST
012356  015516                        MEPASS
012360  104410  012612                CONVRT  ,OUTCRY
012364  104402  015237                TYPE    ,DEVICE
012370  105767  166606                TSTB    MULTD                   ;ARE YOU RUNNING MULTIPLE DEVICES ?
012374  001511                        BEQ     CCC                     ;NO,JUMP AROUND
012376  005767  166614                TST     ACTREG                  ;ARE ANY DEVICES ACTIVE ?
012402  001007                        BNE     RUNIT                   ;YES
012404  104402  015251                TYPE    ,MCOW                   ;NO
012410  016700  166602                MOV     ACTREG,R0               ;DISPLAY ACTREG
012414  000000                        HALT                            ;SELECT SOMETHING TO RUN @ ACTREG:
                                        ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
012416  000167  166636                JMP     .START                  ;START OVER AGAIN.....YOU DESELECTED EVERYTHING
012422  062767  000010  166554  RUNIT:  ADD     #10,BASEADD             ;NEXT BLOCK (ADDRESSES)
012430  062767  000010  166554  ZERO:   ADD     #10,BASEIV              ;NEXT BLOCK (VECTORS)
012436  000241                        CLC
012440  006167  166554                ROL     ROTADD                  ;UP DATE ROTATING POINTER
012444  103410                        BCS     2$                      ;IS IT THE LAST DEVICE
                                                                        ;TO BE TESTED IN THIS PASS ?
012446  036767  166546  166542        BIT     ROTADD,ACTREG           ;TEST THIS DEVICE FOR ACTIVE STATUS
012454  001762                        BEQ     RUNIT                   ;IF NOT ACTIVE, TRY NEXT ADDRESS
012456  004767  000034                JSR     PC,REPLAY               ;CALCULATE NEW PARAMETERS
012462  000167  000174                JMP     RESTRT                  ;YES IT WAS ACTIVE,TEST THIS DEVICE
012466  012767  000001  166524  2$:   MOV     #1,ROTADD               ;OK!,NOW SET UP  ROTATING
                                                                        ;POINTER FOR NEXT MULTIPLE PASS
012474  016767  166506  166502        MOV     KEEPADD,BASEADD         ;RESTORE BASE ADDRESS
012502  016767  166506  166502        MOV     KEEPIV,BASEIV           ;RESTORE BASE INTERRUPT VECTORS
012510  004767  000002                JSR     PC,REPLAY               ;CALC NEW PARAMETERS
012514  000441                        BR      CCC                     ;JUMP AROUND REPLAY
012516  016767  166462  004126  REPLAY: MOV     BASEADD,DUBASE          ;SET UP FOR NEW ADDRESSES
012524  004767  003770                JSR     PC,DUADDR               ;CREATE NEW ADDRESSES
012530  016767  166456  004436        MOV     BASEIV,DURIV            ;CREATE DURIV
012536  062767  000002  166446        ADD     #2,BASEIV
012544  016767  166442  004424        MOV     BASEIV,DURIS            ;CREATE DURIS
012552  062767  000002  166432        ADD     #2,BASEIV
012560  016767  166426  004412        MOV     BASEIV,DUTIV            ;CREATE DUTIV
012566  062767  000002  166416        ADD     #2,BASEIV
012574  016767  166412  004400        MOV     BASEIV,DUTIS            ;CREATE DUTIS
012602  016767  004366  166402        MOV     DURIV,BASEIV            ;RESTORE
012610  000207                        RTS     PC

012612  000001                OUTCRY: 1
012614      006     002               .BYTE   6,2
012616  017150                        RXCSR

012620                          CCC:
012620  005067  166310                CLR     LSTERR                  ;CLEAR LAST ERROR PC
012624  005067  166374                CLR     ERRFLG                  ;CLEAR ERROR FLAG
012630  005267  166274                INC     PASCNT                  ;UPDATE PASS COUNT
```

```
 2537  012634  016777  166270  166240          MOV     PASCNT,@LIGHTS          ;DISPLAY PASS COUNT
 2538  012642  013701  000042                  MOV     @#42,R1                 ;CHECK FOR ACT-11 OR DDP
 2539  012646  001405                          BEQ     RESTRT                  ;IF NOT, CONTINUE TESTING
 2540  012650  000005                          RESET
 2541  012652  004711                  LOGICAL:        JSR     PC,(R1)
 2542  012654  000240                          NOP
 2543  012656  000240                          NOP
 2544  012660  000240                          NOP
 2545  012662  012767  000340  165106  RESTRT: MOV     #340,PS                 ;PREVENT INTERRUPTS (PRIO: 7)
 2546  012670  104413                          CKSWR                           ;CHECK FOR ↑G
 2547  012672  012767  002350  166214          MOV     #TST1,RTRN
 2548  012700  000167  167444                  JMP     TST1
 2549
 2550                                          ;SCOPE LOOP AND INTERATION HANDLER
 2551
 2552  012704                          .SCOPE:
 2553                                  ;**** START OF CODE FOR THE X OR TESTER ******
 2554  012704  000424                          BR      4$                      ;IF RUNNING ON THE X OR TESTER CHANGE
 2555                                                                          ;THIS INSTRUCTION TO A "NOP"(NOP=240)
 2556  012706  013746  000004                  MOV     @#4,-(SP)               ;SAVE CONTENTS OF ERROR VECTOR
 2557  012712  012737  012732  000004          MOV     #1$,@#4                 ;SET FOR TIME OUT
 2558  012720  005737  177060                  TST     @#177060                ;TIME OUT ON  X OR ?
 2559  012724  012637  000004                  MOV     (SP)+,@#4               ;RESTORE ERROR VECTOR
 2560  012730  000404                          BR      2$                      ;GO TO NEXT TEST
 2561  012732  022626                  1$:     CMP     (SP)+,(SP)+             ;CLEAR THE STACK AFTER A TIMEOUT
 2562  012734  012637  000004                  MOV     (SP)+,@#4               ;RESTORE ERROR VECTOR
 2563  012740  000403                          BR      3$                      ;LOOP ON PRESENT TEST
 2564  012742  016767  166150  166144  2$:     MOV     NEXT,RTRN               ;SET UP NEXT TEST IN RTRN
 2565  012750  016716  166140          3$:     MOV     RTRN,(SP)               ;SET UP STACK FOR RTI
 2566  012754  000002                          RTI
 2567  012756                          4$:     ;**** END OF CODE FOR THE  X OR TESTER ******
 2568  012756  104413                          CKSWR                           ;CHECK FOR ↑G
 2569  012760  032777  040000  166112          BIT     #SW14,@SWR              ;LOOP ON CURRENT TEST ?
 2570  012766  001407                  TTST:   BEQ     1$
 2571  012770  000432                          BR      3$
 2572  012772  105777  166106                  TSTB    @TKCSR                  ;TEST TTY FLAG
 2573  012776  100027                          BPL     3$
 2574  013000  017700  166102                  MOV     @TKDBR,R0               ;CLR DONE BIT
 2575  013004  000412                          BR      2$                      ;IF A TTY KEY IS STRUCK GO TO NEXT TST
 2576  013006  032777  004000  166064  1$:     BIT     #SW11,@SWR              ;INHIBIT ITERATIONS ?
 2577  013014  001006                          BNE     2$
 2578  013016  005267  166102                  INC     LPCNT
 2579  013022  026767  166076  166072          CMP     LPCNT,ICOUNT            ;CHECK FOR ITERATION CNT FINISH
 2580  013030  101412                          BLOS    3$
 2581  013032  105067  166166          2$:     CLRB    ERRFLG
 2582  013036  005067  166062                  CLR     LPCNT
 2583  013042  012767  000005  166052          MOV     #5,ICOUNT               ;SET UP ITERATION COUNT
 2584  013050  016767  166042  166036          MOV     NEXT,RTRN               ;SET UP NEXT TEST IN RTRN
 2585  013056  016716  166032          3$:     MOV     RTRN,(SP)               ;SET UP STACK FOR RTI
 2586  013062  000002                          RTI
 2587  013064  001407                  BRW:    1407                            ;RESTORE "BEQ 1$" INSTRUCTION
 2588  013066  000432                  BRX:    432                             ;RESTORE "BR 3$" INSTRUCTION
 2589
 2590                                          ;CHECK FOR FREEZE ON CURRENT DATA
 2591
 2592  013070  104413                  .SCOP1: CKSWR                           ;CHECK  FOR ↑G
```

```
 2593  013072  032777  001000  166000          BIT     #SW09,@SWR
 2594  013100  001402                          BEQ     1$
 2595  013102  016716  166012                  MOV     LOCK,(SP)
 2596  013106  000002                  1$:     RTI
 2597
 2598                                          ;TELETYPE OUTPUT ROUTINE
 2599
 2600  013110  010546                  .TYPE:  MOV     R5,-(SP)
 2601  013112  017605  000002                  MOV     @2(SP),R5
 2602  013116  062766  000002  000002          ADD     #2,2(SP)
 2603  013124  105715                  1$:     TSTB    (R5)            ;LOOK FOR "0"
 2604  013126  001406                          BEQ     3$
 2605  013130  105777  165754          2$:     TSTB    @TPCSR          ;TEST DONE BIT
 2606  013134  100375                          BPL     2$
 2607  013136  112577  165750                  MOVB    (R5)+,@TPDBR    ;TYPE CHAR
 2608  013142  000770                          BR      1$              ;DO IT AGAIN UNTIL "0" IS SEEN
 2609  013144  012605                  3$:     MOV     (SP)+,R5
 2610  013146  000002                          RTI
 2611
 2612                                          ;ASCII STRING INPUT ROUTINE
 2613
 2614  013150  010346                  .INSTR: MOV     R3,-(SP)
 2615  013152  010446                          MOV     R4,-(SP)
 2616  013154  017667  000004  000010          MOV     @4(SP),.MSG
 2617  013162  062766  000002  000004          ADD     #2,4(SP)
 2618  013170  104402                  .INST1: TYPE
 2619  013172  000000                  .MSG:   0
 2620  013174  012704  016304                  MOV     #INBUF,R4
 2621  013200  012703  000007                  MOV     #7,R3
 2622  013204  105777  165674          1$:     TSTB    @TKCSR
 2623  013210  100375                          BPL     1$
 2624  013212  117714  165670                  MOVB    @TKDBR,(R4)
 2625  013216  142714  000200                  BICB    #200,(R4)
 2626  013222  121427  000025                  CMPB    (R4),#25        ;IS IT <^U>
 2627  013226  001003                          BNE     200$
 2628  013230  104402  015426                  TYPE,MCRLF
 2629  013234  000755                          BR      .INST1
 2630  013236  122427  000015          200$:   CMPB    (R4)+,#15
 2631  013242  001423                          BEQ     INSTR2
 2632  013244  117777  165636  165640          MOVB    @TKDBR,@TPDBR
 2633  013252  105777  165632          2$:     TSTB    @TPCSR
 2634  013256  100375                          BPL     2$
 2635  013260  005303                          DEC     R3
 2636  013262  001350                          BNE     1$
 2637  013264  000402                          BR      .INSTG
 2638  013266  010346                  .INSTE: MOV     R3,-(SP)
 2639  013270  010446                          MOV     R4,-(SP)
 2640  013272  104402                  .INSTG: TYPE
 2641  013274  015422                          MQM
 2642  013276  005737  014564                  TST     @#RDSW
 2643  013302  001402                          BEQ     400$
 2644  013304  104402  015426                  TYPE,MCRLF
 2645  013310  000727                  400$:   BR      .INST1
 2646  013312  012604                  INSTR2: MOV     (SP)+,R4
 2647  013314  012603                          MOV     (SP)+,R3
 2648  013316  000002                          RTI
```

```
                                        ;CONVERT ASCII STRING TO OCTAL
                                .PARAM: MOV     R5,-(SP)
                                        MOV     R4,-(SP)
                                        MOV     4(SP),R5
                                        MOV     (R5)+,LOLIM
                                        MOV     (R5)+,HILIM
                                        MOV     (R5)+,DEVADR
                                        MOVB    (R5)+,LOBITS
                                        MOVB    (R5)+,ADRCNT
                                        MOV     R5,4(SP)
                                PARAM1: CLR     R5
                                        MOV     #INBUF,R4
                                        CMPB    #15,(R4)
                                        BEQ     PARERR
                                1$:     CMPB    (R4),#60
                                        BLT     PARERR
                                        CMPB    (R4),#67
                                        BGT     PARERR
                                        BICB    #60,(R4)
                                        BISB    (R4)+,R5
                                        CMPB    #15,(R4)
                                        BEQ     LIMITS
                                        ASL     R5
                                        ASL     R5
                                        ASL     R5
                                        BR      1$
                                PARERR: CMPB    #15,(R4)        ;IS FIRST CHARACTER A <CR>
                                        BNE     120$
                                        TST     @#RDSW          ;IS CKSWR ROUTINE BEING USED
                                        BNE     PARTI
                                120$:   INSTER
                                        BR      PARAM1

                                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS

                                LIMITS: CMP     R5,HILIM
                                        BHI     PARERR
                                        CMP     R5,LOLIM
                                        BLO     PARERR
                                        BITB    LOBITS,R5
                                        BNE     PARERR

                                        ;STORE NUMBER AT SPECIFIED ADDRESS

                                        MOV     DEVADR,R4
                                1$:     MOV     R5,(R4)+
                                        ADD     #2,R5
                                        DECB    ADRCNT
                                        BNE     1$
                                PARTI:  MOV     (SP)+,R4
                                        MOV     (SP)+,R5
                                        RTI
                                LOLIM:  0
                                HILIM:  0
```

```
2705  013530  000000                  DEVADR: 0
2706  013532  000000                  LOBITS: 0
2707          013533                  ADRCNT=LOBITS+1
2708
2709                                          ;SAVE PC OF TEST THAT FAILED AND R0-R5
2710
2711  013534  016667  000004  165432  .SAV05: MOV     4(SP),SAVPC
2712
2713                                          ;SAVE R0-R5
2714
2715  013542  010567  165422          SV05:   MOV     R5,SAVR5
2716  013546  010467  165414                  MOV     R4,SAVR4
2717  013552  010367  165406                  MOV     R3,SAVR3
2718  013556  010267  165400                  MOV     R2,SAVR2
2719  013562  010167  165372                  MOV     R1,SAVR1
2720  013566  010067  165364                  MOV     R0,SAVR0
2721  013572  000002                          RTI
2722
2723                                          ;RESTORE R0-R5
2724
2725  013574  016700  165356          .RES05: MOV     SAVR0,R0
2726  013600  016701  165354                  MOV     SAVR1,R1
2727  013604  016702  165352                  MOV     SAVR2,R2
2728  013610  016703  165350                  MOV     SAVR3,R3
2729  013614  016704  165346                  MOV     SAVR4,R4
2730  013620  016705  165344                  MOV     SAVR5,R5
2731  013624  000002                          RTI
2732
2733                                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2734
2735  013626  104402                  .CONVR: TYPE
2736  013630  015426                          MCRLF
2737  013632  010046                  .CNVRT: MOV     R0,-(SP)
2738  013634  010146                          MOV     R1,-(SP)
2739  013636  010346                          MOV     R3,-(SP)
2740  013640  010446                          MOV     R4,-(SP)
2741  013642  010546                          MOV     R5,-(SP)
2742  013644  017601  000012                  MOV     @12(SP),R1
2743  013650  016767  002470  165272          MOV     TEMP,TEMP3
2744  013656  062766  000002  000012          ADD     #2,12(SP)
2745  013664  012167  000154                  MOV     (R1)+,WRDCNT
2746  013670  112167  000152          1$:     MOVB    (R1)+,CHRCNT
2747  013674  112167  000147                  MOVB    (R1)+,SPACNT
2748  013700  013167  000144                  MOV     @(R1)+,BINWRD
2749  013704  016704  000140          2$:     MOV     BINWRD,R4
2750  013710  116705  000132                  MOVB    CHRCNT,R5
2751  013714  012700  016344                  MOV     #TEMP,R0
2752  013720  010403                  3$:     MOV     R4,R3
2753  013722  042703  177770                  BIC     #177770,R3
2754  013726  062703  000060                  ADD     #060,R3
2755  013732  110320                          MOVB    R3,(R0)+
2756  013734  006204                          ASR     R4              ;SHIFT FOR NEXT #
2757  013736  042704  100000                  BIC     #100000,R4      ;CLUGE TO STOP BIT 15 PROPAGATING.
2758  013742  006204                          ASR     R4              ;DITTO
2759  013744  006204                          ASR     R4              ;DITTO
2760  013746  005305                          DEC     R5
```

```
2761  013750  001363                        BNE     3$
2762  013752  012703  016404                MOV     #MDATA,R3
2763  013756  114023                4$:     MOVB    -(R0),(R3)+
2764  013760  105367  000062                DECB    CHRCNT
2765  013764  001374                        BNE     4$
2766  013766  105767  000055                TSTB    SPACNT
2767  013772  001405                        BEQ     6$
2768  013774  112723  000040          5$:   MOVB    #040,(R3)+
2769  014000  105367  000043                DECB    SPACNT
2770  014004  001373                        BNE     5$
2771  014006  105013                6$:     CLRB    (R3)
2772  014010  104402                        TYPE
2773  014012  016404                        MDATA
2774  014014  005367  000024                DEC     WRDCNT
2775  014020  001323                        BNE     1$
2776  014022  016767  165122  002314        MOV     TEMP3,TEMP
2777  014030  012605                        MOV     (SP)+,R5
2778  014032  012604                        MOV     (SP)+,R4
2779  014034  012603                        MOV     (SP)+,R3
2780  014036  012601                        MOV     (SP)+,R1
2781  014040  012600                        MOV     (SP)+,R0
2782  014042  000002                        RTI
2783  014044  000000              WRDCNT:   0
2784  014046  000000              CHRCNT:   0
2785          014047              SPACNT=CHRCNT+1
2786  014050  000000              BINWRD:   0
2787
2788                                        ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
2789                                        ;BUFFER TO THE CHARACTERS "N" AND "Y".
2790                                        ;IF THE CHARACTER IS "N" CLEAR THE FLAG
2791                                        ;IF THE CHARACTER IS "Y" SET THE FLAG
2792
2793  014052  017605  000000        .SETFLG:MOV     @(SP),R5
2794  014056  122767  000116  002220        CMPB    #'N,INBUF       ;IS IT "N" ?
2795  014064  001002                        BNE     1$
2796  014066  105015                        CLRB    (R5)    ;000
2797  014070  000406                        BR      2$
2798  014072  122767  000131  002204  1$:   CMPB    #'Y,INBUF               ;IS IT "Y" ?
2799  014100  001005                        BNE     3$
2800  014102  112715  177777              MOVB    #-1,(R5)                ;377
2801  014106  062716  000002        2$:   ADD     #2,(SP)
2802  014112  000002                        RTI
2803  014114  104404                3$:     INSTER                          ;RETRY
2804  014116  000755                        BR      .SETFLG
2805                                        ;TRAP DISPATCH SERVICE
2806                                        ;ARGUMENT OF TRAP IS EXTRACTED
2807                                        ;AND USED AS OFFSET TO OBTAIN POINTER
2808                                        ;TO SELECTED SUBROUTINE
2809
2810  014120  011646                .TRPSR: MOV     (SP),-(SP)              ;GET PC OF RETURN
2811  014122  162716  000002                SUB     #2,(SP)                 ;=PC OF TRAP
2812  014126  017616  000000                MOV     @(SP),(SP)              ;GET TRP
2813  014132  006316                TRPOK:  ASL     (SP)                    ;MULTIPLY TRAP ARG BY 2
2814  014134  042716  177001                BIC     #177001,(SP)            ;CLEAR UNWANTED BITS
2815  014140  062716  001226                ADD     #.TRPTAB,(SP)           ;POINTER TO SUBROUTINE ADDRESS
2816  014144  017616  000000                MOV     @(SP),(SP)              ;SUBROUTINE ADDRESS
```

```
2817  014150  000136                          JMP     @(SP)+          ;GO TO SUBROUTINE
2818
2819                                          ;ERROR HANDLER
2820
2821  014152  104413                  .HLT:   CKSWR                   ;CHECK FOR ↑G
2822  014154  032777  020000  164716          BIT     #SW13,@SWR      ;INHIBIT ERROR TYPE OUT ?
2823  014162  001061                          BNE     HALTS
2824  014164  021667  164744                  CMP     (SP),LSTERR
2825  014170  001404                          BEQ     1$
2826  014172  011667  164736                  MOV     (SP),LSTERR
2827  014176  105067  165022                  CLRB    ERRFLG
2828  014202  104406                  1$:     SAVOS
2829  014204  011605                          MOV     (SP),R5
2830  014206  162705  000002                  SUB     #2,R5
2831  014212  011504                          MOV     (R5),R4
2832  014214  006304                          ASL     R4
2833  014216  061504                          ADD     (R5),R4
2834  014220  006304                          ASL     R4
2835  014222  042704  177001                  BIC     #177001,R4
2836  014226  062704  017120                  ADD     #.ERRTAB,R4
2837  014232  012467  000040                  MOV     (R4)+,ERRMSG
2838  014236  012467  000046                  MOV     (R4)+,DATAHD
2839  014242  011467  000054                  MOV     (R4),DATABP
2840  014246  105767  164752                  TSTB    ERRFLG
2841  014252  001403                          BEQ     TYPMSG
2842  014254  005767  000042                  TST     DATABP
2843  014260  001014                          BNE     TYPDAT
2844  014262  104410                  TYPMSG: CONVRT
2845  014264  014414                          ERTABO
2846  014266  112767  177777  164730          MOVB    #-1,ERRFLG
2847  014274  104402                          TYPE
2848  014276  000000                  ERRMSG: 0
2849  014300  005767  000004                  TST     DATAHD
2850  014304  001402                          BEQ     TYPDAT
2851  014306  104402                          TYPE
2852  014310  000000                  DATAHD: 0
2853  014312  005767  000004          TYPDAT: TST     DATABP
2854  014316  001402                          BEQ     RESREG
2855  014320  104410                          CONVRT
2856  014322  000000                  DATABP: 0
2857  014324  104407                  RESREG: RESOS
2858  014326  005777  164546          HALTS:  TST     @SWR
2859  014332  100005                          BPL     EXITER
2860  014334  010046                          PUSHRO
2861  014336  016600  000002                  MOV     2(SP),RO
2862  014342  000000                          HALT
2863  014344  012600                          POPRO
2864  014346  104413                  EXITER: CKSWR                   ;CHECK FOR ↑G
2865  014350  005267  164556                  INC     ERRCNT
2866  014354  032777  000400  164516          BIT     #SW08,@SWR      ;LOOP ON ERROR ?
2867  014362  001007                          BNE     1$
2868  014364  032777  002000  164506          BIT     #SW10,@SWR      ;ESCAPE TO NEXT ON ERROR ?
2869  014372  001407                          BEQ     2$
2870  014374  016767  164516  164512          MOV     NEXT,RTRN       ;SET UP FOR NEXT TEST
2871  014402  012706  001100          1$:     MOV     #STACK,SP       ;REINITIALIZE SP
2872  014406  000177  164502                  JMP     @RTRN
```

```
014412 000002                   2$:     RTI
014414 000001                   ERTABO: 1
014416    006     002                   .BYTE   6,2
014420 001174                           SAVPC
                                        ;ENTER HERE ON POWER FAILURE


014422 010046                   .PFAIL: MOV     R0,-(SP)                ;SAVE R0-R5 ON PROCESSOR STACK
014424 010146                           MOV     R1,-(SP)
014426 010246                           MOV     R2,-(SP)
014430 010346                           MOV     R3,-(SP)
014432 010446                           MOV     R4,-(SP)
014434 010546                           MOV     R5,-(SP)
014436 016746 163362                    MOV     24,-(SP)
014442 010667 164524                    MOV     SP,SAVSP                ;SAVE STACK POINTER
014446 012767 014460 163350             MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
014454 000000                           HALT                            ;HALT ON POWER DOWN NORMAL
014456 000777                   1$:     BR      1$

                                        ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

014460 016706 164506            RESTAR: MOV     SAVSP,SP                ;RESTORE STACK POINTER
014464 012605                           MOV     (SP)+,R5                ;RESTORE R0-R5
014466 012604                           MOV     (SP)+,R4
014470 012603                           MOV     (SP)+,R3
014472 012602                           MOV     (SP)+,R2
014474 012601                           MOV     (SP)+,R1
014476 012600                           MOV     (SP)+,R0
014500 012767 014422 163316             MOV     #.PFAIL,24              ;SET UP FOR POWER FAILURE
014506 012767 000340 163262             MOV     #340,PS
014514 012706 001100                    MOV     #STACK,SP
014520 005067 001620                    CLR     TEMP
014524 005267 001614            1$:     INC     TEMP
014530 001375                           BNE     1$
014532 104410                           CONVRT
014534 014556                           PFTAB
014536 104402                           TYPE
014540 015431                           MPFAIL
014542 005067 164456                    CLR     ERRFLG
014546 005067 164362                    CLR     LSTERR
014552 000177 164336                    JMP     @RTRN
014556 000001                   PFTAB:  1
014560    006     002                   .BYTE   6,2
014562 001114                           RTRN


                                        ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ↑G TO ALLOW CHANGING
                                        ;OF LOC.176.
                                        ;LOCATIONS USED:
014564 000000                           RDSW:   .WORD   0


014566 005737 000042            .CKSWR: TST     @#42
014572 001042                           BNE     OUT
014574 022767 000176 164276             CMP     #SWREG,SWR              ;SOFTWARE SWITCH REGISTER PRESENT
014602 001036                           BNE     OUT                     ;NO, GET OUT
```

```
 2929  014604  105777  164274                    TSTB    @TKCSR          ;YES,WAIT FOR
 2930  014610  100033                            BPL     OUT             ;READY, GET CHARACTER
 2931  014612  017767  164270  176352            MOV     @TKDBR,.MSG     ;AND STRIP OFF
 2932  014620  042767  177600  176344            BIC     #177600,.MSG    ;THE GARBAGE
 2933  014626  122767  000007  176336            CMPB    #7,.MSG         ;IS IT A <^G>
 2934  014634  001021                            BNE     OUT
 2935  014636  104402  014714                    TYPE,$CNTG
 2936  014642  005137  014564          .CNTLU:   COM     @#RDSW
 2937  014646  104402  014721                    TYPE,$MSWR
 2938  014652  104411  014706                    CNVRT,SWREGC
 2939  014656  104403  014731                    INSTR,$MNEW
 2940  014662  104405                            PARAM
 2941  014664  000000                            0
 2942  014666  177777                            177777
 2943  014670  000176                            SWREG
 2944  014672     000     001          .BYTE     0,1
 2945  014674  104402  015426                    TYPE,MCRLF
 2946  014700  005037  014564          OUT:      CLR     @#RDSW
 2947  014704  000002                            RTI
 2948  014706  000001                  SWREGC:   1
 2949  014710     006     002          .BYTE     6,2
 2950  014712  000176                            SWREG
 2951  014714  005015  043536     000  $CNTG:    .ASCIZ  <15><12>/^G/
 2952  014721     015  051412  051127  $MSWR:    .ASCIZ  <15><12>/SWR= /
 2953  014726  020075     000
 2954  014731     040  047040  053505  $MNEW:    .ASCIZ  /  NEW= /
 2955  014736  020075     000
 2956          014742                    .EVEN
 2957  014742  005015  042012  030525  MTITLE:   .ASCIZ  <15><12><12>/DU11 DZDUE-C TAPE E /<15><12>
 2958  014750  020061  055104  052504
 2959  014756  026505  020103  040524
 2960  014764  042520  042440  006440
 2961  014772  000012
 2962  014774  005015  042526  052103  MVECTO:   .ASCIZ  <15><12>/VECTOR ADDRESS-/
 2963  015002  051117  040440  042104
 2964  015010  042522  051523  000055
 2965  015016  005015  051461  020124  MREGAD:   .ASCIZ  <15><12>/1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-/
 2966  015024  042504  044526  042503
 2967  015032  020072  042522  042503
 2968  015040  053111  051105  041440
 2969  015046  047117  051124  046117
 2970  015054  051040  043505  051511
 2971  015062  042524  020122  042101
 2972  015070  051104  051505  026523
 2973  015076     000
 2974  015077     015  040412  042522  MMULT:    .ASCIZ  <15><12>/ARE YOU RUNNING MULTIPLE DEVICES ? (Y OR N)-/
 2975  015104  054440  052517  051040
 2976  015112  047125  044516  043516
 2977  015120  046440  046125  044524
 2978  015126  046120  020105  042504
 2979  015134  044526  042503  020123
 2980  015142  020077  054450  047440
 2981  015150  020122  024516  000055
 2982  015156  005015  040514  052123  MLASTD:   .ASCIZ  <15><12>/LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-/
 2983  015164  042040  053105  041511
 2984  015172  035105  042522  042503
```

```
 2985  015200  053111  051105  041440
 2986  015206  047117  051124  046117
 2987  015214  051040  043505  051511
 2988  015222  042524  020122  042101
 2989  015230  051104  051505  026523
 2990  015236     000
 2991  015237     075  042504  044526  DEVICE: .ASCIZ  /=DEVICE  /
 2992  015244  042503  020040     000
 2993  015251     015  044012  053517  MCOW:   .ASCIZ  <15><12>/HOW NOW BROWN COW? ...SELECT SOMETHING TO RUN @ACTREG/
 2994  015256  047040  053517  041040
 2995  015264  047522  047127  041440
 2996  015272  053517  020077  027056
 2997  015300  051456  046105  041505
 2998  015306  020124  047523  042515
 2999  015314  044124  047111  020107
 3000  015322  047524  051040  047125
 3001  015330  040040  041501  051124
 3002  015336  043505     000
 3003  015341     015  047412  052125  MRANGE: .ASCIZ  <15><12>/OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-/
 3004  015346  047440  020106  040522
 3005  015354  043516  035105  042522
 3006  015362  054524  042520  046040
 3007  015370  051501  020124  042504
 3008  015376  044526  042503  051040
 3009  015404  041530  051123  040440
 3010  015412  042104  042522  051523
 3011  015420  000055
 3012  015422  020040  000077          MQM:    .ASCIZ  /  ?/
 3013  015426  005015     000          MCRLF:  .ASCIZ  <15><12>
 3014  015431     040  050040  053517  MPFAIL: .ASCIZ  /  POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
 3015  015436  051105  043040  044501
 3016  015444  052514  042522  020054
 3017  015452  051120  043517  040522
 3018  015460  020115  042522  052123
 3019  015466  051101  020124  052101
 3020  015474  052040  051505  020124
 3021  015502  047111  050040  047522
 3022  015510  051107  051505  000123
 3023  015516  005015  047105  020104  MEPASS: .ASCIZ  <15><12>/END OF PASS TAPE E/
 3024  015524  043117  050040  051501
 3025  015532  020123  040524  042520
 3026  015540  042440     000
 3027  015543     015  051012     000  MR:     .ASCIZ  <15><12>/R/
 3028  015547     015  052012  051505  MTSTPC: .ASCIZ  <15><12>/TEST PC-/
 3029  015554  020124  041520  000055
 3030  015562  005015  047514  045503  MLOCK:  .ASCIZ  <15><12>/LOCK ON SELECTED TEST? (Y OR N)-/
 3031  015570  047440  020116  042523
 3032  015576  042514  052103  042105
 3033  015604  052040  051505  037524
 3034  015612  024040  020131  051117
 3035  015620  047040  026451     000
 3036  015625     015  042012  020125  MLEVEL: .ASCIZ  <15><12>/DU PRIORITY LEVEL-/
 3037  015632  051120  047511  044522
 3038  015640  054524  046040  053105
 3039  015646  046105  000055
 3040  015652  005015  020043  043117  MSYNC:  .ASCIZ  <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/
```

```
3041  015660  051440  047131  020103
3042  015666  044103  051101  020123
3043  015674  042523  042514  052103
3044  015702  042105  024040  030440
3045  015710  047440  020122  024462
3046  015716  000055
3047  015720  005015  051511  051440  MWIRE6: .ASCIZ  <15><12>/IS SEC XMIT JUMPER #6 IN? (Y OR N)-/
3048  015726  041505  054040  044515
3049  015734  020124  052512  050115
3050  015742  051105  021440  020066
3051  015750  047111  020077  054450
3052  015756  047440  020122  024516
3053  015764  000055
3054  015766  005015  051511  051440  MWIRE5: .ASCIZ  <15><12>/IS SEC REC JUMPER #5 IN? (Y OR N)-/
3055  015774  041505  051040  041505
3056  016002  045040  046525  042520
3057  016010  020122  032443  044440
3058  016016  037516  024040  020131
3059  016024  051117  047040  026451
3060  016032     000
3061  016033     015  044412  020123  MWIRE4: .ASCIZ  <15><12>/IS OPT CLR ENABLE JUMPER #4 IN? (Y OR N)-/
3062  016040  050117  020124  046103
3063  016046  020122  047105  041101
3064  016054  042514  045040  046525
3065  016062  042520  020122  032043
3066  016070  044440  037516  024040
3067  016076  020131  051117  047040
3068  016104  026451     000
3069  016107     015  040412  042522  MEXTJ:  .ASCII  <15><12>/ARE YOU RUNNING IN MAINT MODE EXTERNAL?/
3070  016114  054440  052517  051040
3071  016122  047125  044516  043516
3072  016130  044440  020116  040515
3073  016136  047111  020124  047515
3074  016144  042504  042440  052130
3075  016152  051105  040516  037514
3076  016160  005015  040401  042116          .ASCII  <15><12><1>/AND ..... DO YOU HAVE THE EXTERNAL MODEM BYPASS/
3077  016166  027040  027056  027056
3078  016174  042040  020117  047531
3079  016202  020125  040510  042526
3080  016210  052040  042510  042440
3081  016216  052130  051105  040516
3082  016224  020114  047515  042504
3083  016232  020115  054502  040520
3084  016240  051523
3085  016242  005015  045001  046525          .ASCIZ  <15><12><1>/JUMPER CONNECTOR ON ?(Y OR N)-/
3086  016250  042520  020122  047503
3087  016256  047116  041505  047524
3088  016264  020122  047117  037440
3089  016272  054450  047440  020122
3090  016300  024516  000055
3091                                  .EVEN
3092
3093                                  ;BUFFERS FOR INPUT-OUTPUT
3094
3095  016304  000040                  INBUF:  .BLKB   40
3096  016344  000040                  TEMP:   .BLKB   40
```

```
3097  016404  000040                    MDATA:  .BLKB   40
3098                                    ;*****************************************************************
3099                                            ;UTILITIES
3100                                    ;*****************************************************************
3101
3102                                    ;THIS UTILITY CALCULATES PRIORITY LEVEL
3103  016444  006367  000044            DULEV:  ASL     DUPRT   ;SHIFT LEFT
3104  016450  006367  000040                    ASL     DUPRT   ;
3105  016454  006367  000034                    ASL     DUPRT   ;
3106  016460  006367  000030                    ASL     DUPRT   ;
3107  016464  006367  000024                    ASL     DUPRT   ;
3108  016470  016767  000020  000020           MOV     DUPRT,LESS1     ;MOVE THIS TO LESS1
3109  016476  162767  000001  000012           SUB     #1,LESS1        ;CREATE LESS1
3110  016504  042767  000037  000004           BIC     #37,LESS1       ;CLEAR TNZVC
3111  016512  000207                    RTS     PC
3112  016514  000240                    DUPRT:  LEVEL5
3113  016516  000200                    LESS1:  LEVEL4  ;LEVEL TO ALLOW INTERRUPTS
3114
3115                                            ;NEW DU ADDRESSES
3116  016520  016767  000126  000422    DUADDR: MOV     DUBASE,RXCSR    ;XXX0
3117  016526  005267  000120                    INC     DUBASE
3118  016532  016767  000114  000412           MOV     DUBASE,HRXCSR   ;XXX1
3119  016540  005267  000106                    INC     DUBASE
3120  016544  016767  000102  000402           MOV     DUBASE,RXDBUF   ;XXX2
3121  016552  016767  000074  000400           MOV     DUBASE,PARCSR   ;XXX2
3122  016560  005267  000066                    INC     DUBASE
3123  016564  016767  000062  000364           MOV     DUBASE,HRXDBUF  ;XXX3
3124  016572  016767  000054  000362           MOV     DUBASE,HPARCSR  ;XXX3
3125  016600  005267  000046                    INC     DUBASE
3126  016604  016767  000042  000352           MOV     DUBASE,TXCSR    ;XXX4
3127  016612  005267  000034                    INC     DUBASE
3128  016616  016767  000030  000342           MOV     DUBASE,HTXCSR   ;XXX5
3129  016624  005267  000022                    INC     DUBASE
3130  016630  016767  000016  000332           MOV     DUBASE,TXDBUF   ;XXX6
3131  016636  005267  000010                    INC     DUBASE
3132  016642  016767  000004  000322           MOV     DUBASE,HTXDBUF  ;XXX7
3133  016650  000207                    RTS     PC
3134  016652  000000                    DUBASE: 0
3135
3136                                    ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3137                                    ;INFORMATION CONTAINED IN TEMP1 AND IT IS
3138                                    ;SHIFTED IN BY THE CONTENTS OF SHIFT
3139  016654  042777  040000  000302    RPOKE:  BIC     #MTDATA,@TXCSR
3140  016662  005067  162260                    CLR     TEMP2
3141  016666  006067  162252                    ROR     TEMP1   ;FORCE CARRY
3142  016672  006067  162250                    ROR     TEMP2   ;PICK UP CARRY IN BIT 15
3143  016676  006267  162244                    ASR     TEMP2   ;SHIFT INTO BIT 14
3144  016702  042767  100000  162236           BIC     #BIT15,TEMP2    ;CLR BIT 15
3145  016710  056777  162232  000246           BIS     TEMP2,@TXCSR    ;POKE MAINT DATA
3146  016716  042777  020000  000240           BIC     #CLK,@TXCSR     ;POKE CLK
3147  016724  052777  020000  000232           BIS     #CLK,@TXCSR     ;
3148  016732  005367  162202                    DEC     SHIFT
3149  016736  001346                    BNE     RPOKE
3150  016740  000207                    RTS     PC
3151
3152                                    ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
```

```
3153  016742  016767  162176  162176  ODD8:   MOV     TEMP1,TEMP2     ;SAVE TEMP1
3154  016750  005067  162174                  CLR     TEMP2
3155  016754  012727  000010                  MOV     #8.,(PC)+
3156  016760  000000                  1$:     0
3157  016762  006067  162160          2$:     ROR     TEMP2
3158  016766  005567  162156                  ADC     TEMP3
3159  016772  005367  177762                  DEC     1$
3160  016776  001371                          BNE     2$
3161  017000  006067  162144                  ROR     TEMP3
3162  017004  103404                          BCS     3$
3163  017006  052767  000400  162130          BIS     #BIT8,TEMP1     ;SET ODD PARITY
3164  017014  000403                          BR      4$
3165  017016  042767  000400  162120  3$:     BIC     #BIT8,TEMP1     ;CLR EVEN PARITY
3166                                          ;TEMP1 NOW HAS ODD PARITY CHARACTER
3167  017024  000207                  4$:     RTS     PC
3168
3169                                  ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3170  017026  016767  162112  162112  EVEN8:  MOV     TEMP1,TEMP2     ;SAVE TEMP1
3171  017034  005067  162110                  CLR     TEMP3
3172  017040  012727  000010                  MOV     #8.,(PC)+
3173  017044  000000                  1$:     0
3174  017046  006067  162074          2$:     ROR     TEMP2
3175  017052  005567  162072                  ADC     TEMP3
3176  017056  005367  177762                  DEC     1$
3177  017062  001371                          BNE     2$
3178  017064  006067  162060                  ROR     TEMP3
3179  017070  103004                          BCC     3$
3180  017072  052767  000400  162044          BIS     #BIT8,TEMP1     ;SET EVEN PARITY
3181  017100  000403                          BR      4$
3182  017102  042767  000400  162034  3$:     BIC     #BIT8,TEMP1     ;CLR ODD PARITY
3183                                          ;TEMP1 NOW HAS EVEN PARITY CHARACTER
3184  017110  000207                  4$:     RTS     PC
3185  017112  062716  000002          TRPREG: ADD     #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3186                                          ;IN MAIN PART OF THE PROGRAM
3187  017116  000002                          RTI
3188                                          ;ERROR HLT TABLE
3189  017120  017204                  .ERRTAB:        EM0     ;HLT 0  BIT ERROR (GENERAL)
3190  017122  000000                          0
3191  017124  000000                          0
3192  017126  017220                          EM1     ;HLT 1  REGISTER ERROR
3193  017130  017371                          DH1
3194  017132  017412                          DT1
3195  017134  017262                          EM2     ;HLT 2  RECEIVER ERROR
3196  017136  017371                          DH1
3197  017140  017412                          DT1
3198  017142  017324                          EM3     ;HLT 3  TRANSMITTER ERROR
3199  017144  017371                          DH1
3200  017146  017412                          DT1
3201                                  ;DEFAULT DU ADDRESSES
3202  017150  160040          RXCSR:  160040
3203  017152  160041          HRXCSR: 160041
3204  017154  160042          RXDBUF: 160042
3205  017156  160043          HRXDBUF:        160043
3206  017160  160042          PARCSR: 160042
3207  017162  160043          HPARCSR:        160043
3208  017164  160044          TXCSR:  160044
```

```
3209  017166  160045                    HTXCSR: 160045
3210  017170  160046                    TXDBUF: 160046
3211  017172  160047                    HTXDBUF:        160047
3212                                     ;DEFAULT DU VECTORS
3213  017174  000770                    DURIV:  770     ;REC INTR VECTOR
3214  017176  000772                    DURIS:  772     ;REC INTR STATUS
3215  017200  000774                    DUTIV:  774     ;XMIT INTR VECTOR
3216  017202  000776                    DUTIS:  776     ;XMIT INTR STATUS
3217                                     ;ERROR MESSAGES
3218  017204  036440  042440  051122    EM0:    .ASCIZ  / # ERROR PC/
3219  017212  051117  050040  000103
3220  017220  036440  051040  043505    EM1:    .ASCIZ  / = REGISTER ERROR PC/<15><12><1>/REGISTER  /
3221  017226  051511  042524  020122
3222  017234  051105  047522  020122
3223  017242  041520  005015  051001
3224  017250  043505  051511  042524
3225  017256  020122  000040
3226  017262  036440  051040  041505    EM2:    .ASCIZ  / = RECEIVER ERROR PC/<15><12><1>/REGISTER  /
3227  017270  044505  042526  020122
3228  017276  051105  047522  020122
3229  017304  041520  005015  051001
3230  017312  043505  051511  042524
3231  017320  020122  000040
3232  017324  036440  052040  040522    EM3:    .ASCIZ  / = TRANSMITTER ERROR PC/<15><12><1>/REGISTER  /
3233  017332  051516  044515  052124
3234  017340  051105  042440  051122
3235  017346  051117  050040  006503
3236  017354  000412  042522  044507
3237  017362  052123  051105  020040
3238  017370     000
3239                                     ;DATA HEADERS FOR ERROR MESSAGES
3240  017371     105  050130  041505    DH1:    .ASCIZ  /EXPECTED  ACTUAL/
3241  017376  042524  020104  040440
3242  017404  052103  040525  000114
3243                                     .EVEN
3244                                     ;DATA TABLES FOR ERROR MESSAGES
3245  017412  000003                    DT1:    3
3246  017414     006     004                     .BYTE   6,4
3247  017416  001164                             SAVR3   ;REGISTER
3248  017420     006     004                     .BYTE   6,4
3249  017422  001156                             SAVR0   ;EXPECTED DATA
3250  017424     006     002                     .BYTE   6,2
3251  017426  001160                             SAVR1   ;ACTUAL DATA
3252          000001                     .END
```

```
AAA     002046  954#
ACTREG  001216  772#  902*  916*  917*  924*  2494  2497  2507
ADRCNT= 013533  2659*  2698*  2707#
BASEAD  001204  767#  884*  921*  922  928*  930*  2501*  2513*  2517
BASEIV  001212  770#  894*  2502*  2514*  2519  2520*  2521  2522*  2523  2524*  2525  2526*
BBB     001664  901  905#
BINWRD  014050  2748*  2749  2786#
BITW  = 002000  682#  1227  1360  1408
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  3144
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  3163  3165  3180  3182
BIT9  = 001000  621#  649  665
BREAK = 000001  689#  1048  1108  1142  1440  1510  1772  1833  1913
BRW     013064  994  1008  2587#
BRX     013066  995  1009  2588#
CARDET= 010000  646#
CCC     012620  2493  2516  2533#
CHRCNT  014046  2746*  2750  2764*  2784#  2785
CKSWR = 104413  814#  2546  2568  2592  2821  2864
CLK   = 020000  681#  1048  1054  1055  1057  1058  1072  1073  1075  1076  1108  1113  1114
                1116  1117  1126  1127  1142  1147  1148  1160  1161  1195  1196  1201  1202
                1207  1210  1224  1225  1276  1277  1280  1281  1307  1308  1354  1355  1363
                1364  1402  1403  1411  1412  1440  1448  1449  1451  1452  1510  1517  1518
                1520  1521  1645  1652  1661  1668  1694  1701  1749  1750  1755  1756  1833
                1839  1840  1869  1870  1913  1919  1920  1971  1972  2015  2016  2019  2020
                2028  2205  2206  2209  2210  2237  3146  3147
CNTLU = 104414  816#  861
CNVRT = 104411  810#  2938
CONVRT= 104410  808#  2490  2844  2855  2907
COUNT   001142  743#  1077*  1085*  1275*  1288*  1298*  1311*  1454*  1458*  1462*  1522*  1527*  1579*
                1583*  1587*  1746*  1774*  2378*  2382*  2423  2427*  2472
CTS   = 020000  645#
DATABP  014322  2839*  2842  2853  2856#
DATAHD  014310  2838*  2849  2852#
DEVADR  013530  2657*  2695  2705#
DEVICE  015237  2491  2991#
DH1     017371  3193  3196  3199  3240#
DISPRE  000174  710#  854
DLIGHT= 177570  600#  845
DNA   = 100000  679#
DNAINT= 000040  686#  2026  2041  2050  2216  2245  2258  2265
DSC   = 100000  643#
DSINTE= 000040  653#  1800  1806  1815  1943  1956  1960  1985
DSR   = 001000  649#
```

```
DSWR  = 177570          599#   844
DTR   = 000002          657#  1807  1935
DTI     017412         3194  3197  3200  3245#
DUADDR  016520          883  2518  3116#
DUBASE  016652          879   882  2517*  3116  3117*  3118  3119*  3120  3121  3122*  3123  3124  3125*
                       3126  3127*  3128  3129*  3130  3131*  3132  3134#
DULEV   016444          949  3103#
DUPRT   016514          946  1791  1842  1922  2024  2063  2101  2102  2146  2214  2288  2372  2374
                       3103*  3104*  3105*  3106*  3107*  3108  3112#
DURIS   017176         1791*  1802  1803*  1812  1813*  1842*  1853  1854*  1878  1879*  1922*  1946  1947*
                       1980  1981*  2288*  2312  2313*  2320  2321*  2372*  2399  2400*  2416  2417*  2447
                       2448*  2521*  3214#
DURIV   017174          890   893   894  1790*  1802*  1812*  1841*  1853*  1878*  1921*  1933*  1946*  1980*
                       2287*  2312*  2320*  2371*  2399*  2416*  2447*  2519*  2526  3213#
DUTIS   017202         2024*  2043  2044*  2047  2048*  2063*  2082  2083*  2086  2087*  2101*  2110  2111*
                       2118  2119*  2146*  2178  2179*  2214*  2248  2249*  2262  2263*  2374*  2401  2402*
                       2418  2419*  2449  2450*  2525*  3216#
DUTIV   017200         2023*  2043*  2047*  2062*  2082*  2086*  2100*  2110*  2118*  2145*  2158*  2178*  2213*
                       2235*  2248*  2262*  2373*  2401*  2418*  2449*  2523*  3215#
EIGHT = 006000          674#  1051  1111  1145  1183  1271  1347  1396  1443  1513  1641  1743  1836
                       1916  2011  2201  2283  2355
EM0     017204         3189  3218#
EM1     017220         3192  3220#
EM2     017262         3195  3226#
EM3     017324         3198  3232#
ERRCNT  001132          736#   835*  2865*
ERRFLG  001224          782#   834*  2535*  2581*  2827*  2840  2846*  2911*
ERRMSG  014276         2837*  2848#
ERTABO  014414         2845   2874#
EVEN8   017026         3170#
EVEPAR= 001400          677#
EVPAR = 000400          666#
EXITER  014346         2859   2864#
FIVE  = 000000          671#
FRMERR= 020000          662#
HALTS   014326         2823   2858#
HOXEN = 000010          688#  1691
HILIM   013526         2656*  2686  2704#
HOLD    001136          741#  1647  1654  1663  1670  1696  1703
HPARCS  017162         3124*  3207#
HRXCSR  017152         3118*  3203#
HRXDBU  017156         3123*  3205#
HTXCSR  017166         3128*  3209#
HTXDBU  017172         3132*  3211#
ICOUNT  001122          732#  2579  2583*
INBUF   016304          957   961  2620  2662  2794  2798  3095#
INIFLG  001222          780#   840   843*
INSTER= 104404          800#   965  2681  2803
INSTR = 104403          798#   874   885   895   906   931   941   955   968   972   976   980   999
                       1012  2939
INSTR2  013312         2631  2646#
ISYMOD= 000000          670#  2281  2283
JMPBY   001203          763#   983  1629  1783  1900
KEEPAD  001206          768#   882*  884   928   930  2513
KEEPIV  001214          771#   893*  2514
LASTAD  001210          769#   911   922   936
```

```
LESS1   016516          1792  1843  1923  1938  2025  2064  2147  2215  2240  2289  2375  3108* 3109*
                        3110* 3113#
LEVEL0= 000000           640#
LEVEL1= 000040           639#
LEVEL2= 000100           638#
LEVEL3= 000140           637#
LEVEL4= 000200           636#  3113
LEVEL5= 000240           635#  3112
LEVEL6= 000300           634#
LEVEL7= 000340           633#  1795  1817  1846  1890  1927  1951  1992  2036  2051  2075  2089  2109
                        2114  2127  2156  2164  2181  2225  2253  2267  2309  2319  2396  2413  2445
                        2479
LIGHTS  001102           721#   845*  854*  2537*
LIMITS  013454          2672  2686#
LOBITS  013532          2658* 2690  2706# 2707
LOCK    001120           731#  1182* 2595
LOGICA  012652          2541#
LOKFLG  001225           783#  1002  1003
LOLIM   013524          2655* 2688  2703#
LPCNT   001124           733#   831*  2578# 2579  2582*
LSTERR  001134           737#   836*  2534# 2824  2826*  2912*
MCOW    015251          2496  2993#
MCRLF   015426          2628  2644  2736  2945  3013#
MDATA   016404          2762  2773  3097#
MEPASS  015516          2489  3023#
MEXT  = 010000           693#  1638
MEXTJ   016107           981   3069#
MINT  = 004000           692#  1048  1108  1142  1187  1268  1344  1393  1440  1510  1740  1833  1913
                        2008  2198
MLASTD  015156           907   2982#
MLEVEL  015625           942   3036#
MLOCK   015562          1000   3030#
MMULT   015077           896   2974#
MPFAIL  015431          2910   3014#
MQM     015422          2641   3012#
MR      015543          1022   3027#
MRANGE  015341           932   3003#
MREGAD  015016           875   2965#
MRESET= 000400           683#  1043  1045  1103  1105  1137  1139  1184  1186  1262  1264  1338  1340
                        1387  1389  1435  1437  1505  1507  1632  1634  1734  1736  1785  1828  1830
                        1908  1910  2002  2004  2061  2099  2144  2192  2194  2280  2282  2346  2348
MSYNC   015652           956   3040#
MTDATA= 040000           680#  1048  1108  1121  1125  1142  1154  1158  1440  1510  1833  1864  1868
                        1913  1966  1970  3139
MTITLE  014742           842   2957#
MTSTPC  015547          1013   3028#
MULTD   001202           762#   898   899   991   2492
MVECTO  014774           886   2962#
MWIRE4  016033           977   3061#
MWIRE5  015766           973   3054#
MWIRE6  015720           969   3047#
NEXT    001116           730#  1040* 1102* 1181* 1260* 1336* 1385* 1434* 1499* 1628* 1733* 1782* 1826*
                        1899* 2000* 2059* 2098* 2143* 2190* 2279* 2345* 2564  2584  2870
NOPAR = 000000           675#  1051  1111  1145  1183  1271  1347  1396  1443  1513  1641  1743  1836
                        1916  2011  2201  2283  2355
ODDPAR= 001000           676#
```

```
ODD8     016742          3153#
ONCE     001362           841     844#
OPTCLR   001201           761#    979     1786    1904
OUT      014700          2926    2928     2930    2934    2946#
OUTCRY   012612          2490    2529#
OUTMUL   002024           904     929      940#
OVRRUN=  040000           661#   1317
PARAM =  104405           802#    876      887     908     933     943     1014    2940
PARAM1   013360          2661#   2682
PARCSR   017160          1044#   1051#    1104#   1111#   1138#   1145#   1185#   1188#   1263#   1271#   1339#   1347#   1388#
                         1396#   1436#    1443#   1506#   1513#   1633#   1641#   1735#   1743#   1829#   1836#   1909#   1916#
                         2003#   2011#    2193#   2201#   2281#   2283#   2347#   2355#   3121#   3206#
PAREN =  001000           665#
PARER =  010000           663#
PARERR   013434          2664    2666     2668    2677#   2687    2689    2691
PARTI    013516          2680    2700#
PASCNT   001130           735#    833#    2536#   2537
PFTAB    014556          2908    2914#
POPRO =  012600           609#    2863
POP1SP=  005726           607#
POP2SP=  022626           611#
PS       177776           601#    828#     987#   1792#   1795#   1817#   1843#   1846#   1890#   1923#   1927#   1938#   1951#
                         1992#   2025#    2036#   2051#   2064#   2075#   2089#   2102#   2109#   2114#   2127#   2147#   2156#
                         2164#   2181#    2215#   2225#   2240#   2253#   2267#   2289#   2309#   2319#   2375#   2396#   2413#
                         2445#   2479#    2545#   2902#
PUSHRO=  010046           608#    2860
PUSH1S=  005746           606#
PUSH2S=  024646           610#
ROSW     014564          2642    2679     2922#   2936#   2946#
RECACT=  004000           647#    1062     1078    1087    1464    1533    1540    1545    1555    1565    1589
REPLAY   012516          2509    2515     2517#
RESREG   014324          2854    2857#
RESTAR   014460          2888    2894#
RESTRT   012662          2510    2539     2545#
RESOS =  104407           806#    2857
RING  =  040000           644#
RINTEN=  000100           652#    1847     1858    1883    1943    1956    1960    1985    2294    2315    2318    2383    2398
                         2421    2446
ROTADD   001220           775#    903#     915#    917     919#    924     927#    2504#   2507    2511#
RPOKE    016654          1061    1068     1084    1461    1474    1526    1539    1553    1564    1586    1599    3139#   3149
RTRN     001114           729#    838#    1017    1021#   1023    2547#   2564#   2565    2584#   2585    2870#   2872    2913
                         2916
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    012422          2495    2501#    2508
RXCSR    017150          1052#   1062     1078    1087    1273#   1284    1313    1445#   1464    1468    1475    1515#   1533
                         1540    1544#    1545    1554#   1555    1559    1565    1569    1589    1593    1600    1642#   1681
                         1710    1744#    1763    1788#   1789    1796    1800#   1806#   1807#   1815#   1837#   1847#   1848
                         1858#   1883#    1906#   1917#   1928    1935#   1943#   1952    1956#   1960#   1985#   2294#   2310
                         2315#   2318#    2328    2356#   2383#   2398#   2414    2421#   2446#   2457    2531    3116#   3202#
RXDBUF   017154          1120    1129     1163    1274    1291    1318    1453    1480    1523    1574    1605    1631    1686
```

```
                    1715    1745    1767    1855    1880    1882    1936    1982    1984    2290    2311    2376    2415
                    3120*   3204#
RXDONE= 000200       651#
RXERR = 100000       660#   1317
SAVPC   001174       756#   2711*   2876
SAVR0   001156       749#   2720*   2725    3249
SAVR1   001160       750#   2719*   2726    3251
SAVR2   001162       751#   2718*   2727
SAVR3   001164       752#   2717*   2728    3247
SAVR4   001166       753#   2716*   2729
SAVR5   001170       754#   2715*   2730
SAVSP   001172       755#   2887*   2894
SAVOS = 104406       804#   2828
SCOPE = 104400       792#   1091    1172    1241    1328    1375    1427    1485    1619    1723    1776    1818    1891
                    1993    2052    2090    2131    2182    2269    2336    2480
SCOP1 = 104401       794#   1237
SEND  = 000020       687#   1187    1268    1344    1349    1352    1393    1423    1638    1740    2008    2198    2292
                    2352
SEREC   001200       760#    975
SETFLG= 104412       812#    897     970     974     978     982    1001
SEVEN = 004000       673#
SEXMIT  001177       759#    971
SHIFT   001140       742#   1059*   1066*   1082*   1118*   1135*   1149*   1170*   1207*   1211*   1221*   1231*   1278*
                    1282*   1304*   1309*   1356*   1372*   1400*   1420*   1421    1424    1459*   1472*   1524*   1537*
                    1552*   1563*   1584*   1597*   1659*   1675*   1676    1679    1692*   1708*   1752*   1757*   1758
                    1761    1859*   1871*   1961*   1973*   2013*   2021*   2203*   2211*   3148*
SIX   = 002000       672#
SPACNT= 014047      2747*   2766    2769*   2785#
SRD   = 002000       648#
STACK = 001100       602#    829     988    2871    2903
STD   = 000010       655#
STFLG   001223       781#    832*
STPSYN= 000400       650#   1273    1445    2356
SVOS    013542      2715#
SWR     001100       720#    844*    849     853*    859     862     997    1010    2569    2576    2593    2822    2858
                    2866    2868    2927
SWREG   000176       711#    853     859    2927    2943    2950
SWREGC  014706      2938    2948#
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#   2866
SW09  = 001000       574#   2593
SW10  = 002000       573#   2868
SW11  = 004000       572#   2576
SW12  = 010000       571#
SW13  = 020000       570#   2822
SW14  = 040000       569#   2569
SW15  = 100000       568#
SYNCNO  001176       758#    959*    963*   1041    1456    1501    1530    1581    2380
SYNEXT= 020000       669#   1138    1145    1263    1271    1339    1347    1388    1396    1633    1641    1735    1743
                    1829    1836    1909    1916    2003    2011    2193    2201
```

```
SYNINT= 030000          668#  1044  1051  1104  1111  1183  1185  1436  1443  1506  1513  2347  2355
SYNSCH= 000020          654#  1052  1273  1445  1515  1544  1554  1642  1744  1837  1917  2294  2356
SYSTST= 014000          694#  2292  2352
TEMP    016344         2743   2751  2776* 2904* 2905* 3096#
TEMP1   001144          744#  1060* 1067* 1083* 1119* 1123* 1150* 1156* 1350* 1358* 1399* 1406* 1460*
                       1473*  1525* 1538* 1585* 1598* 1860* 1866* 1962* 1968* 3141* 3153  3163* 3165*
                       3170   3180* 3182*
TEMP2   001146          745#  1151* 1152* 1159* 1162  3140* 3142* 3143* 3144* 3145  3153* 3157* 3170*
                       3174*
TEMP3   001150          746#  2743* 2776  3154* 3158* 3161* 3171* 3175* 3178*
TEMP4   001152          747#
TEMP5   001154          748#  2297* 2302* 2303  2362* 2368* 2369  2388* 2393* 2394  2437* 2442* 2443
TKCSR   001104          722#  2572  2622  2929
TKDBR   001106          723#  2574  2624  2632  2931
TLAST = 011512         1016   2483#
TPCSR   001110          724#  2609  2633
TPDBR   001112          725#  2607* 2632*
TRPOK   014132         2813#
TRPREG  017112         3185#
TSTNO   001126          734#   837* 1039* 1101* 1180* 1259* 1335* 1384* 1433* 1498* 1627* 1732* 1781*
                       1825*  1898* 1999* 2058* 2097* 2142* 2189* 2278* 2344*
TST1    002350         1015   1021  1039# 2547  2548
TST10   006054         1628   1732#
TST11   006312         1733   1781#
TST12   006532         1782   1825#
TST13   007100         1826   1898#
TST14   007556         1899   1999#
TST15   010052         2000   2058#
TST16   010236         2059   2097#
TST17   010420         2098   2142#
TST18   010614         2143   2189#
TST19   011204         2190   2278#
TST2    002652         1040   1101#
TST20   011512         2279   2344#  2482
TST21 = ****** U       2345
TST3    003250         1102   1180#
TST4    003546         1181   1259#
TST5    004050         1260   1335#
TST6    004256         1336   1384#
TST7    004472         1385   1433#
TST8    004746         1434   1498#
TST9    005476         1499   1627#
TTST    012766          994*   995* 1005* 1006* 1008* 1009* 2570#
TXCSR   017164         1043*  1045* 1048* 1054* 1055* 1057* 1058* 1072* 1073* 1075* 1076* 1103* 1105*
                       1108*  1113* 1114* 1116* 1117* 1121* 1125* 1126* 1127* 1137* 1139* 1142* 1147*
                       1148*  1154* 1158* 1160* 1161* 1184* 1186* 1187* 1189  1195* 1196* 1197  1201*
                       1202*  1203  1209* 1210* 1213  1224* 1225* 1227  1262* 1264* 1268* 1276* 1277*
                       1280*  1281* 1307* 1308* 1338* 1340* 1344* 1348  1349* 1352* 1354* 1355* 1363*
                       1364*  1365  1387* 1389* 1393* 1397  1402* 1403* 1411* 1412* 1416* 1423* 1435*
                       1437*  1440* 1448* 1449* 1451* 1452* 1505* 1507* 1510* 1517* 1518* 1520* 1521*
                       1632*  1634* 1638* 1645* 1652* 1661* 1668* 1691* 1694* 1701* 1734* 1736* 1740*
                       1749*  1750* 1755* 1756* 1772* 1785* 1828* 1830* 1833* 1839* 1840* 1864* 1868*
                       1869*  1870* 1908* 1910* 1913* 1919* 1920* 1966* 1970* 1971* 1972* 2002* 2004*
                       2008*  2015* 2016* 2019* 2020* 2026* 2028* 2037  2041* 2050* 2061* 2065* 2070*
                       2076*  2077  2099* 2103* 2108* 2115* 2144* 2148* 2166  2180* 2192* 2194* 2198*
                       2205*  2206* 2209* 2210* 2216* 2226  2230  2237* 2245* 2254  2258* 2265* 2280*
```

```
                    2282*  2292*  2346*  2348*  2352*  2384*  2397*  2422*  2436*  3126*  3139*  3145*  3146*
                    3147*  3208#
TXDBUF  017170      1193*  1272*  1299*  1351*  1398*  1643*  1678*  1747*  1760*  2012*  2202*  2238*  2286*
                    2387*  2435*  3130*  3210#
TXDONE= 000200       684#
TXINTE= 000100       685#  2065   2070   2076   2103   2108   2115   2148   2180   2216   2245   2258   2265
                    2384   2397   2422   2436
TYPDAT  014312      2843   2850   2853#
TYPE  = 104402       796#   842   1022   2488   2491   2496   2618   2628   2640   2644   2735   2772   2847
                    2851   2909   2935   2937   2945
TYPMSG  014262      2841   2844#
USER  = 000000       691#
VOID  = 000001       658#
WRDCNT  014044      2745*  2774*  2783#
ZERO    012430      2502#
$CNTG   014714      2935   2951#
$E    = 000026       552#  1040   1041#  1102   1103#  1181   1183#  1260   1261#  1336   1337#  1385   1386#
                    1434   1435#  1499   1500#  1628   1629#  1733   1734#  1782   1783#  1826   1827#  1899
                    1900#  2000   2001#  2059   2060#  2098   2099#  2143   2144#  2190   2191#  2279   2280#
                    2345   2346#
$MNEW   014731      2939   2954#
$MSWR   014721      2937   2952#
$N    = 000024       552#  1039   1041#  1101   1103#  1180   1183#  1259   1261#  1335   1337#  1384   1386#
                    1433   1435#  1498   1500#  1627   1629#  1732   1734#  1781   1783#  1825   1827#  1898
                    1900#  1999   2001#  2058   2060#  2097   2099#  2142   2144#  2189   2191#  2278   2280#
                    2344   2346#  2482#
$Y    = 000015       784#   792    794#   796#   798#   800#   802#   804#   806#   808#   810#   812#   814#
                    816#   818#
.     = 017430       696#   699#   709#   716#  2956#  3095#  3096#  3097#
.BEGIN  002156       864    987#
.CKSWR  014566       815   2925#
.CNTLU  014642       817   2936#
.CNVRT  013632       811   2737#
.CONVR  013626       809   2735#
.EOP    012354      2345   2488#
.ERRTA  017120      2836   3189#
.HLT    014152       702   2821#
.INSTE  013266       801   2638#
.INSTG  013272      2637   2640#
.INSTR  013150       799   2614#
.INST1  013170      2618#  2629   2645
.MSG    013172      2616*  2619#  2931*  2932*  2933
.PARAM  013320       803   2652#
.PFAIL  014422       700    830   2880#  2901
.RESOS  013574       807   2725#
.SAVOS  013534       805   2711#
.SCOPE  012704       793   2552#
.SCOP1  013070       795   2592#
.SETFL  014052       813   2793#  2804
.START  001260       712    828#   838   2500
.TRPSR  014120       704   2810#
.TRPTA  001226       789#  2815
.TYPE   013110       797   2600#
```

```
HLT      612#  1064  1080  1089  1132  1166  1191  1199  1205  1235  1286  1294  1315  1321  1369
        1417   1466  1470  1477  1483  1535  1542  1547  1557  1561  1567  1571  1577  1591  1595
        1602   1608  1683  1689  1712  1718  1765  1770  1798  1816  1850  1886  1888  1930  1944
        1954   1988  1990  2033  2039  2071  2079  2128  2153  2168  2170  2221  2228  2232  2246
        2256   2305  2326  2334  2403  2451  2455  2463  2468  2474
PRGEND   552#  2481
PRGFRT   552#   553
PUSSYF   552#  1645  1660  1693
RSETUP   552#  1043  1103  1137  1435  1505  1828  1908
TSETUP   552#  1262  1338  1387  1632  1734  2002  2192  2346
$BEGIN   552#   984
$BINAR   552#
$BUFFE   552#  3092
$CABLE   552#
$CATCH   552#   695
$CLRVE   552#   844
$CONVR   552#  2732
$DNA     552#
$EOP     552#  2481
$GETFL   552#   895   968   972   976   980   999
$GETPA   552#   874   885   905   931   940  1012
$GETSY   552#   950
$HEADE   552#   553
$HLT     552#  2818
$INSTR   552#  2611
$ISOB    552#
$MATCH   552#
$MRR     552#
$MRRW    552#
$MRW     552#
$MSG     552#  2957
$PARAM   552#  2649
$PFAIL   552#  2877
$POKE    552#  1195  1201  1208  1223  1276  1279  1306  1354  1362  1402  1410  1749  1754  2015
        2018   2205  2208
$POKER   552#  1056  1071  1074  1112  1115  1146  1450  1519  1838  1918
$RCNET   552#
$RECAC   552#
$REG     552#  2708
$RESET   552#  1043  1045  1103  1105  1137  1139  1262  1264  1338  1340  1387  1389  1435  1437
        1505   1507  1632  1634  1734  1736  1785  1828  1830  1908  1910  2002  2004  2061  2099
        2144   2192  2194  2280  2282  2346  2348
$RXACT   552#
$SCOPE   552#  2549
$SCOPI   552#  2589
$SETFL   552#  2787
$SETVE   552#   696
$START   552#   820
$STRIP   552#
$SYMBO   552#   565
$SYNCR   552#  1052  1515
$TRAPS   552#   784
$TRPAR   552#
$TRPDE   552#   792   794   796   798   800   802   804   806   808   810   812   814   816
$TRPSR   552#  2805
$TSTNO   552#  1039  1101  1180  1259  1335  1384  1433  1498  1627  1732  1781  1825  1898  1999
```

```
DZDUE-C MACY11 27(1006)  01-OCT-76  09:42  PAGE 84
DZDUEC.P11    05-AUG-76 00:00          CROSS REFERENCE TABLE -- MACRO NAMES                    SEQ 0079

          2058    2097    2142    2189    2278    2344
$TYPE      552#  2597
$UNIBU     552#
$VARIA     552#   715
$WORDF     552#
$WORDO     552#
$WORDP     552#


. ABS.  017430     000


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZDUEC,DZDUEC/CRF/SOL=HELLO.P11,PARA.P11,KEET.P11,DZDUEC.P11
 RUN-TIME: 19 30 3 SECONDS
 RUN-TIME RATIO: 172/53=3.2
 CORE USED:  18K  (35 PAGES)
```

C07

Spooler runtime 10 Seconds, 45 KCS, 290 disk reads, 0 disk writes, 79 pages