DUV/LSI-11

DUV11 OFF-LINE LOGIC

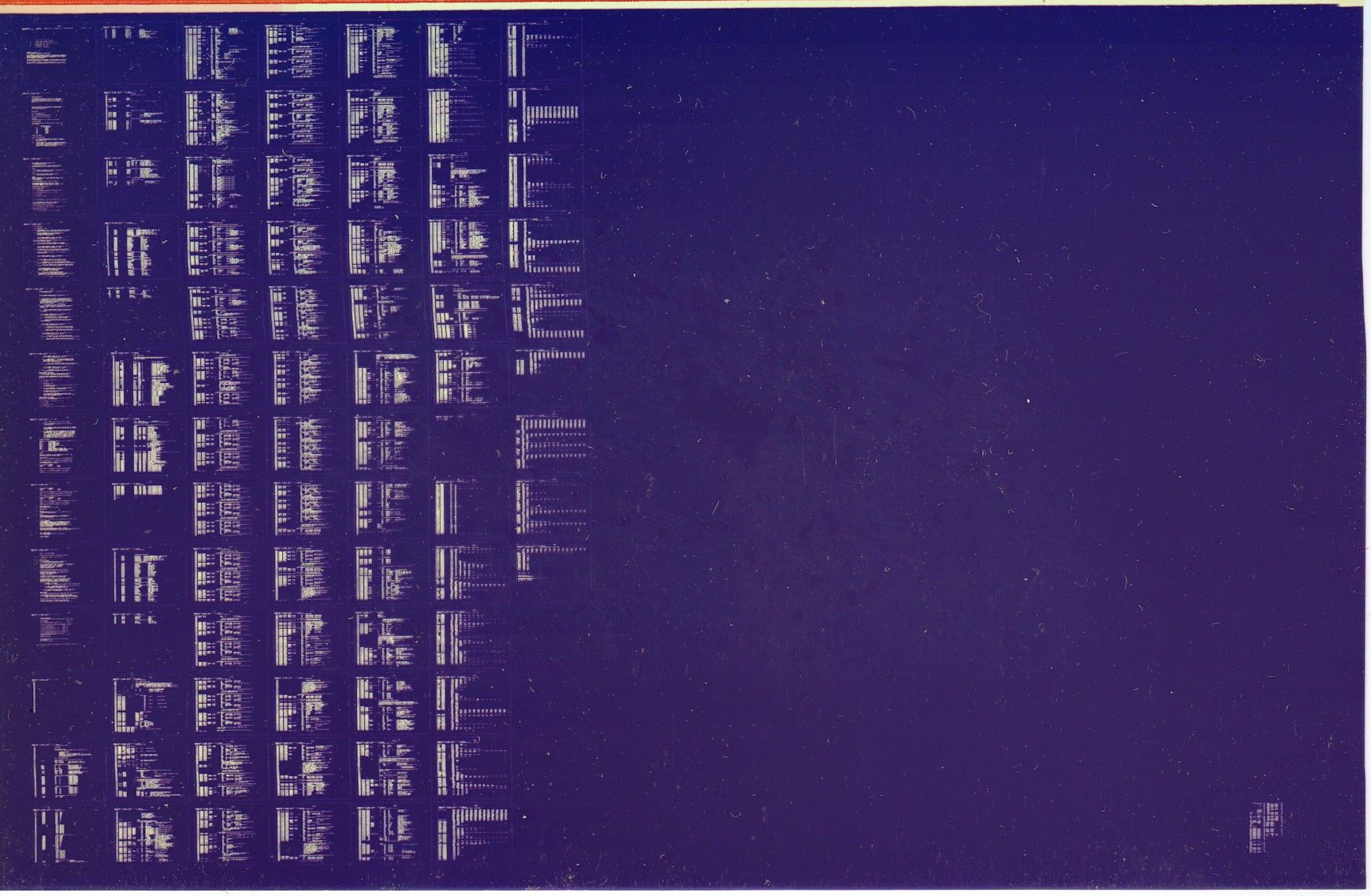
MD-11-DZDUQ-A

EP-DZDUQ-A-DL-A

FICHE 1 OF 1

APR 1977

MADE IN USA



.REM *

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

PRODUCT CODE: MAINDEC-11-DZDUQ-A-D

PRODUCT NAME: DUV11 OFFLINE LOGIC TESTS

RELEASE DATE: FEB. 1977

MAINTAINER : DIAGNOSTICS

*
.REM *

*

.REM *

GENERAL DESCRIPTION

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

*
.REM *

1. THE DUV11 OFFLINE LOGIC TESTS VERIFY THAT ALL REGISTERS EXIST
,AND ALL RESPECTIVE BITS CAN BE MASTER CLEARED, READ,
WRITTEN AND/OR READ/WRITTEN

*
.REM *

2. REQUIREMENTS

PDP-11/03 COMPUTER (LSI)

DUV11 SYNCRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

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

3. LOADING PROCEDURE

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

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

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

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

NOTE: RUNNING UNDER APT-11, THERE IS A USER SWITCH REGISTER
CALLED "SUSWR". IN ORDER TO BE FLEXIBLE ON THE AVAILIBILITY OF THE
H315 CONNECTOR, ONE BIT PASSES STATUS TO APT-11.
BIT 0 IN $USWR REFLECTS THIS STATUS, A 0 = CONNECTOR
PRESENT, A 1 = CONNECTOR NOT AVAILIBLE.

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 SET SWITCH REGISTER (LOC. 176) TO ZERO.

4.3.1.3 TYPE 200G.

4.3.1.4 PROGRAM WILL START.

* .REM *

4.3.1.5 THE PROGRAM WILL TYPE "DUV11 DZDUQ-A TAPE A" (ONCE ONLY)

* .REM *

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

4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN

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

4.3.3 PROGRAM RESTART WITH SW00=1

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

4.3.3.2 TYPE 200G.

4.3.3.3 PROGRAM WILL START.

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

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

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

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

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

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

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

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

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

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

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

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

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

4.3.3.11.1 IF AN "OUT OF RANGE" ADDRESS IS TYPED IE. MORE THAN 16 (10) DEVICES AWAY (UPWARDS).....THE PROGRAM WILL TYPE "OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.11.2 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL

REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED
BY A <CARRIAGE RETURN>

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.3.4.2 TYPE 200G.

4.3.4.3 PROGRAM WILL START.

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

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

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

NOTE: CARE MUST BE TAKEN WHEN THIS FEATURE IS USED
,SINCE THERE IS NO PROTECTION AGAINST SELECTING AN ADDRESS
†HAT 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 SET SW02=1 IN SWITCH REG. (LOC. 176)

4.3.5.2 TYPE 200G.

4.3.5.3 PROGRAM WILL START.

4.3.5.4 THE PROGRAM WILL TYPE "LOCK ON SELECTED TEST ? (Y OR N)-"

AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.5.5 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A
‹CARRIAGE RETURN›

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

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

5. OPERATING PROCEDURE

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

SW15 =1 HALT ON ERROR
SW14 =1 LOOP ON CURRENT TEST
SW13 =1 INHIBIT ERROR TYPEOUT
SW11 =1 INHIBIT ITERATIONS
SW10 =1 ESCAPE TO NEXT TEST ON ERROR
SW09 =1 LOOP ON ERROR
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 (UNDER LSI ALL HALT ERRORS RETURN CONTROL TO O.D.T.)
THERE ARE FOUR DISTINCT ERROR TYPEOUTS

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

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

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

6.1.2 PC +2 = REGISTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.3 PC +2 = RECEIVER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

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

WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

6.1.4 PC +2 = TRANSMITTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

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

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.5 ERROR DESCRIPTIONS
SEE LISTINGS FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

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

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

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

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

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

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

END OF PASS TAPE Y
16XXXX = DEVICE

WHERE Y IS THE TAPE LOADED

WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF

7. RESTRICTIONS

7.1 MULTIPLE DEVICES

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

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

7.2 DISQUALIFYING DEVICES WHEN RUNNING MULTIPLE DEVICES

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

7.2.1 IF DEVICE 0 IS TO BE DISQUALIFIED ,SIMPLY RESTART

PROGRAM WITH SW00 =1 AND OMIT THE FIRST DEVICE.

7.2.2 IF HOWEVER, DEVICES 1 THRU 15 OR ANY COMBINATION THEREOF

ARE TO BE DISQUALIFIED....LOAD THE LOCATION OF ACTREG:
OBSERVE THE ACTIVE BITS (ACTIVE =1, NONACTIVE = 0)
AND DEPOSIT 0 WHERE THOSE DEVICES ARE TO BE DISQUALIFIED

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

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

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

7.3 CABLE DELAYS

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

7.3.1 TO PROVIDE SUFFICIENT DELAY FOR CLOCK SIGNAL OVER THE CABLE,

LOCATION "HOLD:" MUST BE MODIFIED TO ACCOMODATE FOR FASTER MACHINES.
PRESENTLY "HOLD:" =20 IS SUFFICIENT TIME ON AN 11/03 MACHINE.

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

7.4 TO USE THE "XOR" TESTER ,THE BRANCH AROUND THE "XOR"

CODE MUST BE PATCHED TO A "NOP". (SEE LISTINGS FOR DETAILS)

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

VECTOR ADDRESS- DURIV: 770

ARE YOU RUNNING MULTIPLE DEVICES ?- NO MULTD: 0

LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- LASTADD: 0

OF SYNC CHARS SELECTED - 2 SYNCNO: 377

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

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

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

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

9. PROGRAM DESCRIPTION

*
.REM *

9.1 THIS PROGRAM PERFORMS THE OFFLINE LOGIC BIT BANGING
OF THE DEVICE
SEE LISTING FOR DETAILS

*
.REM *
.REM *

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

11. LISTINGS

*

```
                        000001                  STN=1
```

```
                                .ENABLE ABS

                                ;DUV11 DZDUQ-A TAPE A
                                ;COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

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

                                .SBTTL  BASIC DEFINITIONS

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

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

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

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

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

BASIC DEFINITIONS

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

                        .#DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
000001                  BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

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

```
DZDUA-A MACY11 27(1006)  03-FEB-77  07:44  PAGE 16
DZDUAA.M11    27-JAN-77 09:46          BASIC DEFINITIONS

    668          000014                TBITVEC=14          ;;"T" BIT
    669          000014                TRTVEC= 14          ;;TRACE TRAP
    670          000014                BPTVEC= 14          ;;BREAKPOINT TRAP (BPT)
    671          000020                IOTVEC= 20          ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
    672          000024                PWRVEC= 24          ;;POWER FAIL
    673          000030                EMTVEC= 30          ;;EMULATOR TRAP (EMT) **ERROR**
    674          000034                TRAPVEC=34          ;;"TRAP" TRAP
    675          000060                TKVEC=  60          ;;TTY KEYBOARD VECTOR
    676          000064                TPVEC=  64          ;;TTY PRINTER VECTOR
    677          000240                PIRQVEC=240         ;;PROGRAM INTERRUPT REQUEST VECTOR
```

```
                                        ;STANDARD INTERRUPT VECTORS

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


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


                                        ;PROGRAM CONTROL PARAMETERS

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

                                        ;PROGRAM VARIABLES

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

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

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

                                        ;PROGRAM CONTROL FLAGS

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

```
                                ;INSTRUCTION DEFINITIONS

005746                          PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
005726                          POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
010046                          PUSHR0=10046    ;SAVE R0 ON STACK =MOV R0,-(SP)
012600                          POPR0=12600     ;RESTORE R0 FROM STACK =MOV (SP)+,R0
024646                          PUSH2SP=24646   ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
022626                          POP2SP=22626    ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
                                ;REGISTER DEFINITIONS
                                        ;RXCSR BIT DEFINITIONS
100000                          DSC=BIT15                       ;DATA SET CHANGE
040000                          RING=BIT14                      ;RING
020000                          CTS=BIT13                       ;CLR TO SEND
010000                          CARDET=BIT12                    ;CARRIER DETECT
004000                          RECACT=BIT11                    ;REC ACTIVE
002000                          SRO=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
```

```
804          000040                   DNAINTE=BIT5                ;DNA INTR ENAB
805          000020                   SEND=BIT4                   ;SEND
806          000010                   HDXEN=BIT3                  ;HDX/FDX
807          000001                   BREAK=BIT0                  ;BREAK
808                                            ;TXCSR WRD DEFINITIONS
809          000000                   USER=0                      ;USER MODE
810          004000                   MINT=4000                   ;MAINT INT MODE
811          010000                   MEXT=10000                  ;MAINT EXT MODE
812          014000                   SYSTST=14000                ;SYSTEM TEST MODE
```

```
                                        COMMON TAGS

                                .SBTTL  COMMON TAGS

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

                        001400          .=.
001400                          $CMTAG:                         ;;START OF COMMON TAGS
001400  000000                          .WORD   0
001402  000                     $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001403  000                     $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001404  000000                  $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001406  000000                  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001410  000000                  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001412  000000                  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001414  000                     $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001415  001                     $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001416  000000                  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001420  000000                  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001422  000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001424  000000                  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001426  000000                  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001430  000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
001432  000000                          .WORD   0
001434  000                     $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001435  000                     $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001436  000000                          .WORD   0
001440  177570                  $SWR:   .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001442  177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001444  177560                  $TKS:   177560                  ;;TTY KBD STATUS
001446  177562                  $TKB:   177562                  ;;TTY KBD BUFFER
001450  177564                  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001452  177566                  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001454  000                     $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001455  002                     $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001456  012                     $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001457  000                     $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001460  000000                  $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                                ;;WHICH  ($REG0) WAS OBTAINED
001462  000000                  $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001464  000000                  $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001466  000000                  $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001470  000000                  $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001472  000000                  $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001474  000000                  $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001476  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
001500  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
001502  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
001504  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
001506  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
001510  000000                  $TMP5:  .WORD   0               ;;USER DEFINED
001512  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001514  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001516  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001522  077                     $QUES:  .ASCII  /?/             ;;QUESTION MARK
001523  015                     $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
```

```
001524  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001526                          $MAIL:                          ;;APT MAILBOX
001526  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001530  000000                  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001532  000000                  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001534  000000                  $PASS:  .WORD   APASS           ;;PASS COUNT
001536  000000                  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001540  000000                  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001542  000000                  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
001544  000000                  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
001546                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
001546     000                  $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
001547     000                  $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
001550  000000                  $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
001552  000000                  $USWR:  .WORD   AUSWR           ;;USER SWITCHES
001554  000000                  $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
                                ;*                              BITS 15-11=CPU TYPE
                                ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                                      11/70=06,PDQ=07,Q=10
                                ;*                              BIT 10=REAL TIME CLOCK
                                ;*                              BIT  9=FLOATING POINT PROCESSOR
                                ;*                              BIT  8=MEMORY MANAGEMENT
001556     000                  $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
001557     000                  $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
                                ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                                      900 NSEC CORE=001
                                ;*                                      300 NSEC BIPOLAR=002
                                ;*                                      500 NSEC MOS=003
001560  000000                  $MADR1: .WORD   AMADR1          ;;HIGH ADDRESS,BLK#1
                                ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001562     000                  $MAMS2: .BYTE   AMAMS2          ;;HIGH ADDRESS,M.S. BYTE
001563     000                  $MTYP2: .BYTE   AMTYP2          ;;MEM.TYPE,BLK#2
001564  000000                  $MADR2: .WORD   AMADR2          ;;MEM.LAST ADDRESS,BLK#2
001566     000                  $MAMS3: .BYTE   AMAMS3          ;;HIGH ADDRESS,M.S.BYTE
001567     000                  $MTYP3: .BYTE   AMTYP3          ;;MEM.TYPE,BLK#3
001570  000000                  $MADR3: .WORD   AMADR3          ;;MEM.LAST ADDRESS,BLK#3
001572     000                  $MAMS4: .BYTE   AMAMS4          ;;HIGH ADDRESS,M.S.BYTE
001573     000                  $MTYP4: .BYTE   AMTYP4          ;;MEM.TYPE,BLK#4
001574  000000                  $MADR4: .WORD   AMADR4          ;;MEM.LAST ADDRESS,BLK#4
001576  000000                  $VECT1: .WORD   AVECT1          ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001600  000000                  $VECT2: .WORD   AVECT2          ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001602  000000                  $BASE:  .WORD   ABASE           ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001604  000000                  $DEVM:  .WORD   ADEVM           ;;DEVICE MAP
001606  000000                  $CDW1:  .WORD   ACDW1           ;;CONTROLLER DESCRIPTION WORD#1
001610  000000                  $CDW2:  .WORD   ACDW2           ;;CONTROLLER DESCRIPTION WORD#2
001612  000000                  $DDW0:  .WORD   ADDW0           ;;DEVICE DESCRIPTOR WORD#0
001614  000000                  $DDW1:  .WORD   ADDW1           ;;DEVICE DESCRIPTOR WORD#1
001616  000000                  $DDW2:  .WORD   ADDW2           ;;DEVICE DESCRIPTOR WORD#2
001620  000000                  $DDW3:  .WORD   ADDW3           ;;DEVICE DESCRIPTOR WORD#3
001622  000000                  $DDW4:  .WORD   ADDW4           ;;DEVICE DESCRIPTOR WORD#4
001624  000000                  $DDW5:  .WORD   ADDW5           ;;DEVICE DESCRIPTOR WORD#5
```

```
DZDUQ-A MACY11 27(1006)  03-FEB-77  07:44  PAGE 23
DZDUQA.M11    27-JAN-77 09:46          APT MAILBOX-ETABLE

   925  001626  000000                 $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
   926  001630  000000                 $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
   927  001632  000000                 $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
   928  001634  000000                 $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
   929  001636  000000                 $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
   930  001640  000000                 $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
   931  001642  000000                 $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
   932  001644  000000                 $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
   933  001646  000000                 $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
   934  001650  000000                 $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
   935
   936
   937  001652                         $ETEND:
   938
   939
   940
```

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

```
  997          000040                  DNAINTE=BIT5                ;DNA INTR ENAB
  998          000020                  SEND=BIT4                   ;SEND
  999          000010                  HDXEN=BIT3                  ;HDX/FDX
 1000          000001                  BREAK=BIT0                  ;BREAK
 1001                                          ;TXCSR WRD DEFINITIONS
 1002          000000                  USER=0                      ;USER MODE
 1003          004000                  MINT=4000                   ;MAINT INT MODE
 1004          010000                  MEXT=10000                  ;MAINT EXT MODE
 1005          014000                  SYSTST=14000                ;SYSTEM TEST MODE
```

```
                                        ERROR POINTER TABLE

  1006                                  .SBTTL  ERROR POINTER TABLE
  1007
  1008                                  ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
  1009                                  ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
  1010                                  ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
  1011                                  ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
  1012                                  ;*NOTE2:      EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
  1013
  1014                                  ;*      EM              ;;POINTS TO THE ERROR MESSAGE
  1015                                  ;*      DH              ;;POINTS TO THE DATA HEADER
  1016                                  ;*      DT              ;;POINTS TO THE DATA
  1017                                  ;*      DF              ;;POINTS TO THE DATA FORMAT
  1018
  1019
  1020  001652                          SERRTB:
  1021                                          ;ERROR TABLE
  1022  001652  001762                                  EM1     ;ERROR 1        REGISTER ERROR
  1023  001654  002067                                  DH1
  1024  001656  002116                                  DT1
  1025  001660  002132                                  DF1
  1026  001662  002022                                  EM2     ;ERROR 2        RECEIVER ERROR
  1027  001664  002067                                  DH1
  1028  001666  002116                                  DT1
  1029  001670  002132                                  DF1
  1030  001672  002043                                  EM3     ;ERROR 3        TRANSMITTER ERROR
  1031  001674  002067                                  DH1
  1032  001676  002116                                  DT1
  1033  001700  002132                                  DF1
  1034  001702  001746                                  EM4     ;ERROR 4        BIT ERROR (GENERAL)
  1035  001704  000000                                  0
  1036  001706  002126                                  DT4
  1037  001710  002132                                  DF1
  1038
  1039                                          ;DEFAULT DU ADDRESSES
  1040  001712  160010                  RXCSR:  160010
  1041  001714  160011                  HRXCSR: 160011
  1042  001716  160012                  RXDBUF: 160012
  1043  001720  160013                  HRXDBUF:        160013
  1044  001722  160012                  PRXCSR: 160012
  1045  001724  160013                  HPRXCSR:        160013
  1046  001726  160014                  TXCSR:  160014
  1047  001730  160015                  HTXCSR: 160015
  1048  001732  160016                  TXDBUF: 160016
  1049  001734  160017                  HTXDBUF:        160017
  1050                                          ;DEFAULT DU VECTORS
  1051  001736  000770                  DURIV:  770     ;REC INTR VECTOR
  1052  001740  000772                  DURIS:  772     ;REC INTR STATUS
  1053  001742  000774                  DUTIV:  774     ;XMIT INTR VECTOR
  1054  001744  000776                  DUTIS:  776     ;XMIT INTR STATUS
  1055                                          ;ERROR MESSAGES
  1056  001746  020040  051105  047522  EM4:     .ASCIZ  /  ERROR PC /
  1057  001754  020122  041520  000040
  1058  001762  020040  047503  050115  EM1:     .ASCIZ /  COMPARISON ERROR ON REGISTERS/
  1059  001770  051101  051511  047117
  1060  001776  042440  051122  051117
  1061  002004  047440  020116  042522
```

```
                                                ERROR POINTER TABLE

1062   002012  044507  052123  051105
1063   002020  000123
1064   002022  020040  042522  042503  EM2:    .ASCIZ  /  RECEIVER ERROR/
1065   002030  053111  051105  042440
1066   002036  051122  051117     000
1067   002043     040  052040  040522  EM3:    .ASCIZ  /  TRANSMITTER ERROR/
1068   002050  051516  044515  052124
1069   002056  051105  042440  051122
1070   002064  051117     000
1071                                            ;DATA HEADERS FOR ERROR MESSAGES
1072   002067     105  051122  041520  DH1:    .ASCIZ  /ERRPC  WANTED  ACTUAL/
1073   002074  020040  040527  052116
1074   002102  042105  020040  041501
1075   002110  052524  046101     000
1076           002116                   .EVEN
1077                                            ;DATA TABLES FOR ERROR MESSAGES
1078   002116  001416  001130  001132  DT1:    .WORD   $ERRPC,HLD0,HLD1,0
1079   002124  000000
1080
1081   002126  001416  000000          DT4:    .WORD   $ERRPC,0
1082
1083   002132     000     000     000  DF1:    .BYTE   0,0,0,0
1084   002135     000
1085                                    .EVEN
1086                                    .SBTTL  ACT11 HOOKS
1087
1088                                    ;;******************************************************************
1089                                    ;HOOKS REQUIRED BY ACT11
1090           002136                           $SVPC=.                 ;SAVE PC
1091           000046                           .=46
1092   000046  012632                           $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1093           000052                           .=52
1094   000052  000000                           .WORD   0               ;;2)SET LOC.52 TO ZERO
1095           002136                           .=$SVPC                 ;; RESTORE PC
1096                                    .SBTTL  APT PARAMETER BLOCK
1097
1098                                    ;;******************************************************************
1099                                    ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1100                                    ;;******************************************************************
1101           002136                           .$X=.   ;;SAVE CURRENT LOCATION
1102           000024                           .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1103   000024  000200                           200     ;;FOR APT START UP
1104           000044                           .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1105   000044  002136                           $APTHDR ;;POINT TO APT HEADER BLOCK
1106           002136                           .=.$X   ;;RESET LOCATION COUNTER
1107                                    ;;******************************************************************
1108                                    ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1109                                    ;INTERFACE SPEC.
1110
1111   002136                           $APTHD:
1112   002136  000000                   $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1113   002140  001526                   $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1114   002142  000010                   $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
1115   002144  000010                   $PASTM: .WORD   10      ;;RUN TIME IN SECS, OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1116   002146  000000                   $UNITM: .WORD           ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1117   002150  000052                           .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
1118
1119
1120                                        ;PROGRAM INITIALIZATION
1121                                        ;LOCK OUT INTERRUPTS
1122                                        ;SET UP PROCESSOR STACK
1123                                        ;SET UP POWER FAIL VECTOR
1124                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1125                                        ;TYPE TITLE MESSAGE
1126
1127  002152                          .START:
1128                                  .SBTTL  INITIALIZE THE COMMON TAGS
1129                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1130  002152  012706  001400                MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1131  002156  005026                        CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1132  002160  022706  001440                CMP     #$WR,R6 ;;DONE?
1133  002164  001374                        BNE     .-6             ;;LOOP BACK IF NO
1134  002166  012706  001100                MOV     #$STACK,SP              ;;SETUP THE STACK POINTER
1135                                  ;;INITIALIZE A FEW VECTORS
1136  002172  012737  016262  000020        MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1137  002200  012737  000340  000022        MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1138  002206  012737  014152  000030        MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1139  002214  012737  000340  000032        MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1140  002222  012737  016600  000034        MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1141  002230  012737  000340  000036        MOV     #340,@#TRAPVEC+2;LEVEL 7
1142  002236  012737  014754  000024        MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1143  002244  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1144  002252  005067  177234               CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1145  002256  005067  177232               CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1146  002262  112767  000001  177125       MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1147  002270  012767  002270  177110       MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1148  002276  012767  002276  177104       MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1149                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1150                                  ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
1151  002304  013746  000004               MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1152  002310  012737  002344  000004        MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1153  002316  012767  177570  177114       MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1154  002324  012767  177570  177110       MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1155  002332  022777  177777  177100       CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1156  002340  001012                        BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1157                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
1158  002342  000403                        BR      65$             ;;BRANCH IF NO TIMEOUT
1159  002344  012716  002352        64$:   MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1160  002350  000002                        RTI
1161  002352  012767  000176  177060 65$:   MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1162  002360  012767  000174  177054        MOV     #DISPREG,DISPLAY
1163  002366  012637  000004        66$:   MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1164
1165  002372  005067  177136               CLR     $PASS           ;;CLEAR PASS COUNT
1166  002376  132767  000200  177143       BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1167  002404  001403                        BEQ     67$             ;;YES,USE NON-APT SWITCH
1168  002406  012767  001550  177024       MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1169  002414                         67$:
1170  002414  012706  001100               MOV     #STACK,SP               ;SET STACK
1171  002420  106427  000340               MTPS    #340                    ;LOCK INTERRUPTS
1172  002424  012737  014754  000024        MOV     #.PFAIL,@#24            ;SET UP POWER FAIL VECTOR
1173  002432  105067  176535               CLRB    STFLG                   ;CLEAR START FLAG
```

```
1174  002436  005067  176450                  CLR     PASCNT                  ;CLEAR PASS COUNT
1175  002442  105067  176735                  CLRB    SERFLG                  ;CLEAR ERROR FLAG
1176  002446  005067  176740                  CLR     SERTTL                  ;CLEAR ERROR COUNT
1177  002452  005067  176740                  CLR     SERRPC                  ;CLEAR LAST ERROR POINTER
1178  002456  012767  000001  176716          MOV     #1,$TSTNM               ;SET UP FOR TEST 1
1179  002464  012767  002152  176412          MOV     #.START,RETURN          ;SET UP FOR POWER FAIL BEFORE
1180                                                                          ;TESTING STARTS
1181  002472  013746  000006                  MOV     @#6,-(SP)
1182  002476  013746  000004                  MOV     @#4,-(SP)
1183  002502  012737  002516  000004          MOV     #1$,@#4
1184  002510  005777  176724                  TST     @SWR
1185  002514  000407                          BR      2$
1186  002516  012767  000176  176714  1$:     MOV     #SWREG,SWR
1187  002524  012767  000174  176710          MOV     #DISPREG,DISPLAY
1188  002532  022626                          CMP     (SP)+,(SP)+
1189  002534  012637  000004          2$:     MOV     (SP)+,@#4
1190  002540  012637  000006                  MOV     (SP)+,@#6
1191  002544  022767  000176  176666          CMP     #SWREG,SWR
1192  002552  001007                          BNE     3$
1193  002554  005737  000042                  TST     @#42                    ;CHECK FOR CHAIN
1194  002560  001402                          BEQ     33$
1195  002562  000167  000522                  JMP     .BEGIN
1196  002566  004767  010146          33$:    JSR     PC,CNTLU
1197  002572  105767  176374          3$:     TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
1198  002576  001004                          BNE     ONCE
1199  002600  104401  015114                  TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
1200  002604  105167  176362                  COMB    INIFLG                  ;IF NOT SET FLAG AND DO
1201  002610  105767  176732          ONCE:   TSTB    $ENV                    ;APT CONTROL?
1202  002614  001410                          BEQ     11$                     ;BR IF NO
1203  002616  032767  000001  176726          BIT     #1,SUSWR                ;EXTENAL JUMPER ON?
1204  002624  001002                          BNE     12$                     ;NO
1205  002626  105067  176321                  CLRB    JMRBY                   ;CLEAR FLAG
1206  002632  000167  000452          12$:    JMP     .BEGIN                  ;GO DO IT
1207  002636  032777  000001  176574  11$:    BIT     #SW00,@SWR              ;RESELECT VECTOR & CONTROL REG?
1208  002644  001002                          BNE     1$
1209  002646  000167  000436                  JMP     .BEGIN
1210  002652  012700  000300          1$:     MOV     #300,R0 ;RESTORE VECTOR AREA TO TRAPCATCHER
1211  002656  012701  000302                  MOV     #302,R1 ;START AT LOCATION 300
1212  002662  012702  000004                  MOV     #4,R2
1213  002666  010110          2$:     MOV     R1,(R0)
1214  002670  005011                          CLR     (R1)
1215  002672  060200                          ADD     R2,R0
1216  002674  060201                          ADD     R2,R1
1217  002676  022701  001000                  CMP     #1000,R1                ;END AT LOCATION 776
1218  002702  002771                          BLT     2$
1219  002704  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1220  002706  015162                          MREGAD  ;MESSAGE
1221  002710  104410                          PARAM   ;CONVERT STRING
1222  002712  160000                          160000  ;LOW LIMIT
1223  002714  167776                          167776  ;HIGH LIMIT
1224  002716  017074                          DUBASE  ;STORE AT THIS LOCATION
1225  002720     001          .BYTE   1       ;MASK
1226  002721     001          .BYTE   1       ;HOW MANY TIMES + 2
1227  002722  016767  014146  176226          MOV     DUBASE,KEEPADD  ;SAVE
1228  002730  004767  014006                  JSR     PC,DUADDR
1229  002734  016767  176216  176212          MOV     KEEPADD,BASEADD ;RESTORE FOR ROTATION
```

```
1230  002742  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1231  002744  015147                          MVECTO  ;MESSAGE
1232  002746  104410                          PARAM   ;CONVERT STRING
1233  002750  000300                          300     ;LOW LIMIT
1234  002752  000776                          776     ;HIGH LIMIT
1235  002754  001736                          DURIV   ;STORE AT THIS LOCATION
1236  002756     001                  .BYTE   1       ;MASK
1237  002757     004                  .BYTE   4       ;HOW MANY TIMES + 2
1238  002760  016767  176752  176176          MOV     DURIV,KEEPIV    ;SAVE
1239  002766  016767  176744  176166          MOV     DURIV,BASEIV    ;SET UP FOR ROTATION
1240  002774  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1241  002776  015212                          MMULT   ;MESSAGE
1242  003000  104414                          SETFLG  ;SET FLAG BASED UPON INPUT STRING
1243  003002  001152                          MULTD   ;THIS FLAG
1244  003004  105767  176142                  TSTB    MULTD   ;ARE THERE MULTIPLE DEVICES
1245                                                          ;ON THE SYSTEM ?
1246  003010  100406                          BMI     BBB     ;YES,ASK NEXT QUESTION
1247  003012  005067  176150                  CLR     ACTREG
1248  003016  005067  176146                  CLR     ROTADD
1249  003022  000167  000140                  JMP     OUTMUL  ;JUMP AROUND NEXT QUESTION
1250  003026                          BBB:
1251  003026  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1252  003030  015241                          MLASTD  ;MESSAGE
1253  003032  104410                          PARAM   ;CONVERT STRING
1254  003034  160000                          160000  ;LOW LIMIT
1255  003036  167776                          167776  ;HIGH LIMIT
1256  003040  001160                          LASTADD ;STORE AT THIS LOCATION
1257  003042     001                  .BYTE   1       ;MASK
1258  003043     001                  .BYTE   1       ;HOW MANY TIMES + 2
1259                                  ;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1260  003044  012767  000001  176116  1$:     MOV     #1,ROTADD       ;SET UP POINTER
1261  003052  005067  176110                  CLR     ACTREG  ;CLR ACTIVE REGISTER
1262  003056  056767  176106  176102  2$:     BIS     ROTADD,ACTREG   ;MAKE THIS DEVICE ACTIVE
1263  003064  000241                          CLC
1264  003066  006167  176076                  ROL     ROTADD  ;SET UP POINTER
1265  003072  103421                          BCS     3$      ;ARE YOU OUT OF RANGE ?
1266  003074  062767  000010  176052          ADD     #10,BASEADD     ;SET UP BASE ADDRESS
1267  003102  026767  176052  176044          CMP     LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
1268  003110  101362                          BHI     2$      ;NO DO IT AGAIN
1269  003112  056767  176052  176046          BIS     ROTADD,ACTREG   ;THIS ASSUMES THAT THERE ARE AT
1270                                                          ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1271                                                          ;MULTIPLE DEVICE QUESTION
1272  003120  012767  000001  176042  4$:     MOV     #1,ROTADD       ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1273  003126  016767  176024  176020          MOV     KEEPADD,BASEADD ;DITTO
1274  003134  000414                          BR      OUTMUL  ;CONTINUE QUESTIONS
1275  003136  016767  176014  176010  3$:     MOV     KEEPADD,BASEADD ;RESTORE
1276  003144  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1277  003146  015335                          MRANGE  ;MESSAGE
1278  003150  104410                          PARAM   ;CONVERT STRING
1279  003152  160000                          160000  ;LOW LIMIT
1280  003154  167776                          167776  ;HIGH LIMIT
1281  003156  001160                          LASTADD ;STORE AT THIS LOCATION
1282  003160     001                  .BYTE   1       ;MASK
1283  003161     001                  .BYTE   1       ;HOW MANY TIMES + 2
1284  003162  000167  177656                  JMP     1$      ;DO IT AGAIN
1285  003166  012767  000340  013542  OUTMUL: MOV     #340,DUPRT
```

```
1286  003174  004767  013466                      JSR     PC,DULEV
1287                                              ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1288                                              ;BUFFER TO THE CHARACTERS "1" AND "2".
1289                                              ;IF THE CHARACTER IS "1" CLEAR THE FLAG
1290                                              ;IF THE CHARACTER IS "2" SET THE FLAG
1291  003200                            AAA:
1292  003200  104406                             INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1293  003202  015553                             MSYNC   ;MESSAGE
1294  003204  122767  000061  012702   3$:      CMPB    #'1,INBUF       ;IS IT "1" ?
1295  003212  001003                             BNE     1$
1296  003214  105067  175726                    CLRB    SYNCNO  ;000
1297  003220  000412                             BR      4$
1298  003222  122767  000062  012664   1$:      CMPB    #'2,INBUF       ;IS IT "2" ?
1299  003230  001004                             BNE     2$
1300  003232  112767  177777  175706             MOVB    #-1,SYNCNO      ;377
1301  003240  000402                             BR      4$
1302  003242  104407                   2$:       INSTER  ;RETRY
1303  003244  000757                             BR      3$
1304  003246  000240                   4$:       NOP
1305  003250  104406                             INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1306  003252  015621                             MWIRE6  ;MESSAGE
1307  003254  104414                             SETFLG  ;SET FLAG BASED UPON INPUT STRING
1308  003256  001147                             SEXMIT  ;THIS FLAG
1309  003260  104406                             INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1310  003262  015672                             MWIRE5  ;MESSAGE
1311  003264  104414                             SETFLG  ;SET FLAG BASED UPON INPUT STRING
1312  003266  001150                             SEREC   ;THIS FLAG
1313  003270  104406                             INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1314  003272  015742                             MWIRE4  ;MESSAGE
1315  003274  104414                             SETFLG  ;SET FLAG BASED UPON INPUT STRING
1316  003276  001151                             OPTCLR  ;THIS FLAG
1317  003300  104406                             INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1318  003302  016021                             MEXTJ   ;MESSAGE
1319  003304  104414                             SETFLG  ;SET FLAG BASED UPON INPUT STRING
1320  003306  001153                             JMRBY   ;THIS FLAG
1321
1322                                              ;TEST START AND RESTART
1323
1324  003310  012706  001100           .BEGIN:   MOV     #STACK,SP               ;SET UP STACK
1325  003314  106427  000340                     MTPS    #340                    ;LOCK OUT INTERRUPTS
1326  003320  032777  000002  176112             BIT     #SW01,@SWR              ;IF SW01=1, GET STARTING PC
1327  003326  001406                             BEQ     3$
1328  003330  104406                             INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1329  003332  015505                             MTSTPC  ;MESSAGE
1330  003334  104410                             PARAM   ;CONVERT STRING
1331  003336  003362                             TST1    ;LOW LIMIT
1332                                                    ;HIGH LIMIT
1333                                                    ;STORE AT THIS LOCATION
1334  003340     001                  .BYTE   1       ;MASK
1335  003341     001                  .BYTE   1       ;HOW MANY TIMES + 2
1336  003342  000403                             BR      4$
1337  003344  012767  003362  175532   3$:      MOV     #TST1,RETURN            ;START AT TEST 1
1338  003352  104401  015501           4$:      TYPE    ,MR                     ;TYPE R
1339  003356  000177  175522                     JMP     @RETURN                 ;START TESTING
1340
1341
```

```
1342                                                    ;;THIS TEST PROVES EXISTANCE OF DEVICE REGISTERS
1343                                                    ;;
1344                                            ;;**************************************************************************
1345  003362  000004                           TST1:   SCOPE
1346  003364  012737  017334  000004                    MOV     #TRPREG,@#4     ;SETUP TRAPCATCHER
1347  003372  016737  174742  000006                    MOV     PR7,@#6 ;
1348  003400  105277  176306                           INCB    @RXCSR          ;TEST THIS REG
1349  003404  000401                                   BR      .+4             ;IF OK JMP AROUND ERROR
1350  003406  104004                                   ERROR   4                       ;CHECK DEVICE REG ADDRESSES
1351  003410  105277  176300                           INCB    @HRXCSR ;TEST UPPER BYTE THIS REGISTER
1352  003414  000401                                   BR      .+4             ;IF OK JMP AROUND ERROR
1353  003416  104004                                   ERROR   4                       ;CHECK DEVICE REG ADDRESSES
1354  003420  012737  000006  000004                    MOV     #6,@#4          ;RESTORE TRAPCATCHER
1355  003426  012737  000000  000006                    MOV     #0,@#6          ;
1356
1357  003434  012767  003777  175456                    MOV     #3777,HOLD      ;SET LONGER DELAY FOR TEST.
1358                                                    ;;THIS TEST PROVES EXISTANCE OF DEVICE REGISTERS
1359                                                    ;;
1360                                            ;;**************************************************************************
1361  003442  000004                           TST2:   SCOPE
1362  003444  012737  017334  000004                    MOV     #TRPREG,@#4     ;SETUP TRAPCATCHER
1363  003452  016737  174662  000006                    MOV     PR7,@#6 ;
1364  003460  105277  176232                           INCB    @RXDBUF         ;TEST THIS REG
1365  003464  000401                                   BR      .+4             ;IF OK JMP AROUND ERROR
1366  003466  104004                                   ERROR   4                       ;CHECK DEVICE REG ADDRESSES
1367  003470  105277  176224                           INCB    @HRXDBUF        ;TEST UPPER BYTE THIS REGISTER
1368  003474  000401                                   BR      .+4             ;IF OK JMP AROUND ERROR
1369  003476  104004                                   ERROR   4                       ;CHECK DEVICE REG ADDRESSES
1370  003500  012737  000006  000004                    MOV     #6,@#4          ;RESTORE TRAPCATCHER
1371  003506  012737  000000  000006                    MOV     #0,@#6          ;
1372
1373                                                    ;;THIS TEST PROVES EXISTANCE OF DEVICE REGISTERS
1374                                                    ;;
1375                                            ;;**************************************************************************
1376  003514  000004                           TST3:   SCOPE
1377  003516  012737  017334  000004                    MOV     #TRPREG,@#4     ;SETUP TRAPCATCHER
1378  003524  016737  174610  000006                    MOV     PR7,@#6 ;
1379  003532  105277  176164                           INCB    @PARCSR         ;TEST THIS REG
1380  003536  000401                                   BR      .+4             ;IF OK JMP AROUND ERROR
1381  003540  104004                                   ERROR   4                       ;CHECK DEVICE REG ADDRESSES
1382  003542  105277  176156                           INCB    @HPARCSR        ;TEST UPPER BYTE THIS REGISTER
1383  003546  000401                                   BR      .+4             ;IF OK JMP AROUND ERROR
1384  003550  104004                                   ERROR   4                       ;CHECK DEVICE REG ADDRESSES
1385  003552  012737  000006  000004                    MOV     #6,@#4          ;RESTORE TRAPCATCHER
1386  003560  012737  000000  000006                    MOV     #0,@#6          ;
1387
1388                                                    ;;THIS TEST PROVES EXISTANCE OF DEVICE REGISTERS
1389                                                    ;;
1390                                            ;;**************************************************************************
1391  003566  000004                           TST4:   SCOPE
1392  003570  012737  017334  000004                    MOV     #TRPREG,@#4     ;SETUP TRAPCATCHER
1393  003576  016737  174536  000006                    MOV     PR7,@#6 ;
1394  003604  105277  176116                           INCB    @TXCSR          ;TEST THIS REG
1395  003610  000401                                   BR      .+4             ;IF OK JMP AROUND ERROR
1396  003612  104004                                   ERROR   4                       ;CHECK DEVICE REG ADDRESSES
1397  003614  105277  176110                           INCB    @HTXCSR ;TEST UPPER BYTE THIS REGISTER
```

INITIALIZE THE COMMON TAGS

```
1398  003620  000401                        BR      .+4             ;IF OK JMP AROUND ERROR
1399  003622  104004                        ERROR   4                       ;CHECK DEVICE REG ADDRESSES
1400  003624  012737  000006  000004        MOV     #6,@#4          ;RESTORE TRAPCATCHER
1401  003632  012737  000000  000006        MOV     #0,@#6          ;
1402
1403                                        ;;THIS TEST PROVES EXISTANCE OF DEVICE REGISTERS
1404                                        ;
1405                                    ;;**************************************************************
1406  003640  000004                    TST5:   SCOPE
1407  003642  012737  017334  000004        MOV     #TRPREG,@#4     ;SETUP TRAPCATCHER
1408  003650  016737  174464  000006        MOV     PR7,@#6 ;
1409  003656  105277  176050                INCB    @TXDBUF         ;TEST THIS REG
1410  003662  000401                        BR      .+4             ;IF OK JMP AROUND ERROR
1411  003664  104004                        ERROR   4                       ;CHECK DEVICE REG ADDRESSES
1412  003666  105277  176042                INCB    @#TXDBUF        ;TEST UPPER BYTE THIS REGISTER
1413  003672  000401                        BR      .+4             ;IF OK JMP AROUND ERROR
1414  003674  104004                        ERROR   4                       ;CHECK DEVICE REG ADDRESSES
1415  003676  012737  000006  000004        MOV     #6,@#4          ;RESTORE TRAPCATCHER
1416  003704  012737  000000  000006        MOV     #0,@#6          ;
1417
1418                                        ;;BUS DRIVER TEST
1419                                        ;
1420                                    ;;**************************************************************
1421  003712  000004                    TST6:   SCOPE
1422  003714  022777  000000  176010        CMP     #0,@TXDBUF
1423  003722  001401                        BEQ     .+4
1424  003724  104004                        ERROR   4       ;READING TXDBUF SHOULD BE ALL ZERO'S
1425                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
1426                                        ;;TESTING OF READ/WRITE BIT DTR
1427                                        ;
1428                                    ;;**************************************************************
1429  003726  000004                    TST7:   SCOPE
1430  003730  052777  000002  175754        BIS     #DTR,@RXCSR     ;SET THIS BIT
1431  003736  032777  000002  175746        BIT     #DTR,@RXCSR     ;TEST THIS BIT
1432  003744  001001                        BNE     .+4     ;BR IF "1"
1433  003746  104004                        ERROR   4                       ;THIS BIT SHOULD BE SET
1434  003750  042777  000002  175734        BIC     #DTR,@RXCSR     ;CLR THIS BIT
1435  003756  032777  000002  175726        BIT     #DTR,@RXCSR     ;TEST THIS BIT
1436  003764  001401                        BEQ     .+4             ;BR IF "0"
1437  003766  104004                        ERROR   4                       ;THIS BIT SHOULD BE CLR
1438                                        ;NOW SET THIS BIT
1439  003770  052777  000002  175714        BIS     #DTR,@RXCSR
1440  003776  052777  000400  175722        BIS     #MRESET,@TXCSR  ;MASTER RESET
1441                                        ;;CHECK EXISTANCE OF OPTIONAL CLEAR JUMPER
1442                                        ;
1443  004004  105767  175141                TSTB    OPTCLR          ;TEST FLAG
1444  004010  100006                        BPL     1$              ;OPTIONAL CLR JUMPER IS NOT IN
1445  004012  032777  000002  175672        BIT     #DTR,@RXCSR     ;TEST THIS BIT
1446  004020  001401                        BEQ     .+4             ;BR IF "0"
1447  004022  104004                        ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1448  004024  000405                        BR      2$              ;JMP AROUND
1449  004026  032777  000002  175656  1$:   BIT     #DTR,@RXCSR     ;TEST THIS BIT
1450  004034  001001                        BNE     .+4     ;BR IF "1"
1451  004036  104004                        ERROR   4                       ;CHECK OUT OPTIONAL CLR JUMPER
1452  004040  000240                2$:     NOP
1453
```

```
1454                                            ;;THIS TEST PERFORMS MASTER RESET TESTING &
1455                                            ;;TESTING OF READ/WRITE BIT RTS
1456                                            ;;
1457                                    ;;**************************************************************************
1458  004042  000004                   TST10:   SCOPE
1459  004044  052777  000004  175640            BIS     #RTS,@RXCSR     ;SET THIS BIT
1460  004052  032777  000004  175632            BIT     #RTS,@RXCSR     ;TEST THIS BIT
1461  004060  001001                            BNE     .+4     ;BR IF "1"
1462  004062  104004                            ERROR   4                       ;THIS BIT SHOULD BE SET
1463  004064  042777  000004  175620            BIC     #RTS,@RXCSR     ;CLR THIS BIT
1464  004072  032777  000004  175612            BIT     #RTS,@RXCSR     ;TEST THIS BIT
1465  004100  001401                            BEQ     .+4             ;BR IF "0"
1466  004102  104004                            ERROR   4                       ;THIS BIT SHOULD BE CLR
1467                                            ;NOW SET THIS BIT
1468  004104  052777  000004  175600            BIS     #RTS,@RXCSR
1469  004112  052777  000400  175606            BIS     #MRESET,@TXCSR  ;MASTER RESET
1470                                            ;;CHECK EXISTANCE OF OPTIONAL CLEAR JUMPER
1471                                            ;;
1472  004120  105767  175025                   TSTB    OPTCLR          ;TEST FLAG
1473  004124  100006                            BPL     1$              ;OPTIONAL CLR JUMPER IS NOT IN
1474  004126  032777  000004  175556            BIT     #RTS,@RXCSR     ;TEST THIS BIT
1475  004134  001401                            BEQ     .+4             ;BR IF "0"
1476  004136  104004                            ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1477  004140  000405                            BR      2$              ;JMP AROUND
1478  004142  032777  000004  175542  1$:       BIT     #RTS,@RXCSR     ;TEST THIS BIT
1479  004150  001001                            BNE     .+4     ;BR IF "1"
1480  004152  104004                            ERROR   4                       ;CHECK OUT OPTIONAL CLR JUMPER
1481  004154  000240                   2$:      NOP
1482
1483                                            ;WAIT FOR CABLE DELAYS
1484                                            ;######################################
1485                                            ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
1486                                            ;######################################
1487  004156  016702  174736                   MOV     HOLD,R2 ;SET DELAY TIME
1488  004162  005302                            DEC     R2
1489  004164  001376                            BNE     .-2     ;WAIT THIS TIME
1490                                            ;OK NOW FALL THRU AND CONTINUE TESTING.....
1491                                            ;EXIT STAGE LEFT....CHINNNG!
1492                                            ;;THIS TEST PERFORMS MASTER RESET TESTING &
1493                                            ;;TESTING OF READ/WRITE BIT STD
1494                                            ;;
1495                                    ;;**************************************************************************
1496  004166  000004                   TST11:   SCOPE
1497  004170  052777  000010  175514            BIS     #STD,@RXCSR     ;SET THIS BIT
1498  004176  032777  000010  175506            BIT     #STD,@RXCSR     ;TEST THIS BIT
1499  004204  001001                            BNE     .+4     ;BR IF "1"
1500  004206  104004                            ERROR   4                       ;THIS BIT SHOULD BE SET
1501  004210  042777  000010  175474            BIC     #STD,@RXCSR     ;CLR THIS BIT
1502  004216  032777  000010  175466            BIT     #STD,@RXCSR     ;TEST THIS BIT
1503  004224  001401                            BEQ     .+4             ;BR IF "0"
1504  004226  104004                            ERROR   4                       ;THIS BIT SHOULD BE CLR
1505                                            ;NOW SET THIS BIT
1506  004230  052777  000010  175454            BIS     #STD,@RXCSR
1507  004236  052777  000400  175462            BIS     #MRESET,@TXCSR  ;MASTER RESET
1508                                            ;;CHECK EXISTANCE OF OPTIONAL CLEAR JUMPER
1509                                            ;;
```

INITIALIZE THE COMMON TAGS

```
1510  004244  105767  174701                  TSTB    OPTCLR          ;TEST FLAG
1511  004250  100006                          BPL     1$              ;OPTIONAL CLR JUMPER IS NOT IN
1512  004252  032777  000010  175432          BIT     #STD,@RXCSR     ;TEST THIS BIT
1513  004260  001401                          BEQ     .+4             ;BR IF "0"
1514  004262  104004                          ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1515  004264  000405                          BR      2$              ;JMP AROUND
1516  004266  032777  000010  175416  1$:     BIT     #STD,@RXCSR     ;TEST THIS BIT
1517  004274  001001                          BNE     .+4     ;BR IF "1"
1518  004276  104004                          ERROR   4                       ;CHECK OUT OPTIONAL CLR JUMPER
1519  004300  000240                  2$:     NOP
1520
1521                                          ;;THIS TEST PERFORMS MASTER RESET TESTING &
1522                                          ;;TESTING OF READ/WRITE BIT SYNSCH
1523                                          ;;
1524                                  ;;**************************************************************************
1525  004302  000004                  TST12:  SCOPE
1526  004304  052777  000020  175400          BIS     #SYNSCH,@RXCSR  ;SET THIS BIT
1527  004312  032777  000020  175372          BIT     #SYNSCH,@RXCSR  ;TEST THIS BIT
1528  004320  001001                          BNE     .+4     ;BR IF "1"
1529  004322  104004                          ERROR   4                       ;THIS BIT SHOULD BE SET
1530  004324  042777  000020  175360          BIC     #SYNSCH,@RXCSR  ;CLR THIS BIT
1531  004332  032777  000020  175352          BIT     #SYNSCH,@RXCSR  ;TEST THIS BIT
1532  004340  001401                          BEQ     .+4             ;BR IF "0"
1533  004342  104004                          ERROR   4                       ;THIS BIT SHOULD BE CLR
1534                                          ;NOW SET THIS BIT
1535  004344  052777  000020  175340          BIS     #SYNSCH,@RXCSR
1536  004352  052777  000400  175346          BIS     #MRESET,@TXCSR  ;MASTER RESET
1537  004360  032777  000020  175324          BIT     #SYNSCH,@RXCSR  ;TEST THIS BIT
1538  004366  001401                          BEQ     .+4             ;BR IF "0"
1539  004370  104004                          ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1540
1541                                          ;;THIS TEST PERFORMS MASTER RESET TESTING &
1542                                          ;;TESTING OF READ/WRITE BIT DSINTE
1543                                          ;;
1544                                  ;;**************************************************************************
1545  004372  000004                  TST13:  SCOPE
1546  004374  052777  000040  175310          BIS     #DSINTE,@RXCSR  ;SET THIS BIT
1547  004402  032777  000040  175302          BIT     #DSINTE,@RXCSR  ;TEST THIS BIT
1548  004410  001001                          BNE     .+4     ;BR IF "1"
1549  004412  104004                          ERROR   4                       ;THIS BIT SHOULD BE SET
1550  004414  042777  000040  175270          BIC     #DSINTE,@RXCSR  ;CLR THIS BIT
1551  004422  032777  000040  175262          BIT     #DSINTE,@RXCSR  ;TEST THIS BIT
1552  004430  001401                          BEQ     .+4             ;BR IF "0"
1553  004432  104004                          ERROR   4                       ;THIS BIT SHOULD BE CLR
1554                                          ;NOW SET THIS BIT
1555  004434  052777  000040  175250          BIS     #DSINTE,@RXCSR
1556  004442  052777  000400  175256          BIS     #MRESET,@TXCSR  ;MASTER RESET
1557  004450  032777  000040  175234          BIT     #DSINTE,@RXCSR  ;TEST THIS BIT
1558  004456  001401                          BEQ     .+4             ;BR IF "0"
1559  004460  104004                          ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1560
1561                                          ;;THIS TEST PERFORMS MASTER RESET TESTING &
1562                                          ;;TESTING OF READ/WRITE BIT RINTEN
1563                                          ;;
1564                                  ;;**************************************************************************
1565  004462  000004                  TST14:  SCOPE
```

INITIALIZE THE COMMON TAGS

```
1566  004464  052777  000100  175220          BIS     #RINTEN,@RXCSR  ;SET THIS BIT
1567  004472  032777  000100  175212          BIT     #RINTEN,@RXCSR  ;TEST THIS BIT
1568  004500  001001                          BNE     .+4     ;BR IF -1"
1569  004502  104004                          ERROR   4                       ;THIS BIT SHOULD BE SET
1570  004504  042777  000100  175200          BIC     #RINTEN,@RXCSR  ;CLR THIS BIT
1571  004512  032777  000100  175172          BIT     #RINTEN,@RXCSR  ;TEST THIS BIT
1572  004520  001401                          BEQ     .+4                     ;BR IF "0"
1573  004522  104004                          ERROR   4                       ;THIS BIT SHOULD BE CLR
1574                                          ;NOW SET THIS BIT
1575  004524  052777  000100  175160          BIS     #RINTEN,@RXCSR
1576  004532  052777  000400  175166          BIS     #MRESET,@TXCSR  ;MASTER RESET
1577  004540  032777  000100  175144          BIT     #RINTEN,@RXCSR  ;TEST THIS BIT
1578  004546  001401                          BEQ     .+4                     ;BR IF "0"
1579  004550  104004                          ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1580
1581                                          ;;THIS TEST PERFORMS MASTER RESET TESTING &
1582                                          ;;TESTING OF READ/WRITE BIT STPSYN
1583                                          ;;
1584                                  ;;**************************************************************************
1585  004552  000004                  TST15:  SCOPE
1586  004554  052777  000400  175130          BIS     #STPSYN,@RXCSR  ;SET THIS BIT
1587  004562  032777  000400  175122          BIT     #STPSYN,@RXCSR  ;TEST THIS BIT
1588  004570  001001                          BNE     .+4     ;BR IF -1"
1589  004572  104004                          ERROR   4                       ;THIS BIT SHOULD BE SET
1590  004574  042777  000400  175110          BIC     #STPSYN,@RXCSR  ;CLR THIS BIT
1591  004602  032777  000400  175102          BIT     #STPSYN,@RXCSR  ;TEST THIS BIT
1592  004610  001401                          BEQ     .+4                     ;BR IF "0"
1593  004612  104004                          ERROR   4                       ;THIS BIT SHOULD BE CLR
1594                                          ;NOW SET THIS BIT
1595  004614  052777  000400  175070          BIS     #STPSYN,@RXCSR
1596  004622  052777  000400  175076          BIS     #MRESET,@TXCSR  ;MASTER RESET
1597  004630  032777  000400  175054          BIT     #STPSYN,@RXCSR  ;TEST THIS BIT
1598  004636  001401                          BEQ     .+4                     ;BR IF "0"
1599  004640  104004                          ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1600
1601                                          ;;THIS TEST PERFORMS MASTER RESET TESTING &
1602                                          ;;TESTING OF READ/WRITE BIT BREAK
1603                                          ;;
1604                                  ;;**************************************************************************
1605  004642  000004                  TST16:  SCOPE
1606  004644  052777  000001  175054          BIS     #BREAK,@TXCSR   ;SET THIS BIT
1607  004652  032777  000001  175046          BIT     #BREAK,@TXCSR   ;TEST THIS BIT
1608  004660  001001                          BNE     .+4     ;BR IF -1"
1609  004662  104004                          ERROR   4                       ;THIS BIT SHOULD BE SET
1610  004664  042777  000001  175034          BIC     #BREAK,@TXCSR   ;CLR THIS BIT
1611  004672  032777  000001  175026          BIT     #BREAK,@TXCSR   ;TEST THIS BIT
1612  004700  001401                          BEQ     .+4                     ;BR IF "0"
1613  004702  104004                          ERROR   4                       ;THIS BIT SHOULD BE CLR
1614                                          ;NOW SET THIS BIT
1615  004704  052777  000001  175014          BIS     #BREAK,@TXCSR
1616  004712  052777  000400  175006          BIS     #MRESET,@TXCSR  ;MASTER RESET
1617  004720  032777  000001  175000          BIT     #BREAK,@TXCSR   ;TEST THIS BIT
1618  004726  001401                          BEQ     .+4                     ;BR IF "0"
1619  004730  104004                          ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1620
1621                                          ;;THIS TEST PERFORMS MASTER RESET TESTING &
```

```
                                        ;;TESTING OF READ/WRITE BIT HDXEN
                                        ;;
                                ;;*****************************************************************
004732  000004                 TST17:   SCOPE
004734  052777  000010  174764          BIS     #HDXEN,@TXCSR   ;SET THIS BIT
004742  032777  000010  174756          BIT     #HDXEN,@TXCSR   ;TEST THIS BIT
004750  001001                          BNE     .+4     ;BR IF "1"
004752  104004                          ERROR   4                       ;THIS BIT SHOULD BE SET
004754  042777  000010  174744          BIC     #HDXEN,@TXCSR   ;CLR THIS BIT
004762  032777  000010  174736          BIT     #HDXEN,@TXCSR   ;TEST THIS BIT
004770  001401                          BEQ     .+4                     ;BR IF "0"
004772  104004                          ERROR   4                       ;THIS BIT SHOULD BE CLR
                                        ;NOW SET THIS BIT
004774  052777  000010  174724          BIS     #HDXEN,@TXCSR
005002  052777  000400  174716          BIS     #MRESET,@TXCSR  ;MASTER RESET
005010  032777  000010  174710          BIT     #HDXEN,@TXCSR   ;TEST THIS BIT
005016  001401                          BEQ     .+4                     ;BR IF "0"
005020  104004                          ERROR   4                       ;CHECK OUT MASTER RESET LOGIC

                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
                                        ;;TESTING OF READ/WRITE BIT SEND
                                        ;;
                                ;;*****************************************************************
005022  000004                 TST20:   SCOPE
005024  052777  000020  174674          BIS     #SEND,@TXCSR    ;SET THIS BIT
005032  032777  000020  174666          BIT     #SEND,@TXCSR    ;TEST THIS BIT
005040  001001                          BNE     .+4     ;BR IF "1"
005042  104004                          ERROR   4                       ;THIS BIT SHOULD BE SET
005044  042777  000020  174654          BIC     #SEND,@TXCSR    ;CLR THIS BIT
005052  032777  000020  174646          BIT     #SEND,@TXCSR    ;TEST THIS BIT
005060  001401                          BEQ     .+4                     ;BR IF "0"
005062  104004                          ERROR   4                       ;THIS BIT SHOULD BE CLR
                                        ;NOW SET THIS BIT
005064  052777  000020  174634          BIS     #SEND,@TXCSR
005072  052777  000400  174626          BIS     #MRESET,@TXCSR  ;MASTER RESET
005100  032777  000020  174620          BIT     #SEND,@TXCSR    ;TEST THIS BIT
005106  001401                          BEQ     .+4                     ;BR IF "0"
005110  104004                          ERROR   4                       ;CHECK OUT MASTER RESET LOGIC

                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
                                        ;;TESTING OF READ/WRITE BIT DNAINTE
                                        ;;
                                ;;*****************************************************************
005112  000004                 TST21:   SCOPE
005114  052777  000040  174604          BIS     #DNAINTE,@TXCSR ;SET THIS BIT
005122  032777  000040  174576          BIT     #DNAINTE,@TXCSR ;TEST THIS BIT
005130  001001                          BNE     .+4     ;BR IF "1"
005132  104004                          ERROR   4                       ;THIS BIT SHOULD BE SET
005134  042777  000040  174564          BIC     #DNAINTE,@TXCSR ;CLR THIS BIT
005142  032777  000040  174556          BIT     #DNAINTE,@TXCSR ;TEST THIS BIT
005150  001401                          BEQ     .+4                     ;BR IF "0"
005152  104004                          ERROR   4                       ;THIS BIT SHOULD BE CLR
                                        ;NOW SET THIS BIT
005154  052777  000040  174544          BIS     #DNAINTE,@TXCSR
005162  052777  000400  174536          BIS     #MRESET,@TXCSR  ;MASTER RESET
005170  032777  000040  174530          BIT     #DNAINTE,@TXCSR ;TEST THIS BIT
```

```
1678  005176  001401                        BEQ     .+4             ;BR IF "0"
1679  005200  104004                        ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1680
1681                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
1682                                        ;;TESTING OF READ/WRITE BIT TXINTE
1683                                        ;;
1684                                ;;**************************************************************
1685  005202  000004                TST22:  SCOPE
1686  005204  052777  000100  174514        BIS     #TXINTE,@TXCSR  ;SET THIS BIT
1687  005212  032777  000100  174506        BIT     #TXINTE,@TXCSR  ;TEST THIS BIT
1688  005220  001001                        BNE     .+4     ;BR IF "1"
1689  005222  104004                        ERROR   4                       ;THIS BIT SHOULD BE SET
1690  005224  042777  000100  174474        BIC     #TXINTE,@TXCSR  ;CLR THIS BIT
1691  005232  032777  000100  174466        BIT     #TXINTE,@TXCSR  ;TEST THIS BIT
1692  005240  001401                        BEQ     .+4             ;BR IF "0"
1693  005242  104004                        ERROR   4                       ;THIS BIT SHOULD BE CLR
1694                                        ;NOW SET THIS BIT
1695  005244  052777  000100  174454        BIS     #TXINTE,@TXCSR
1696  005252  052777  000400  174446        BIS     #MRESET,@TXCSR  ;MASTER RESET
1697  005260  032777  000100  174440        BIT     #TXINTE,@TXCSR  ;TEST THIS BIT
1698  005266  001401                        BEQ     .+4             ;BR IF "0"
1699  005270  104004                        ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1700
1701                                        ;;TEST MAINT MODE BIT 0
1702                                        ;;
1703                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
1704                                        ;;TESTING OF READ/WRITE BIT BIT11
1705                                        ;;
1706                                ;;**************************************************************
1707  005272  000004                TST23:  SCOPE
1708  005274  052777  004000  174424        BIS     #BIT11,@TXCSR   ;SET THIS BIT
1709  005302  032777  004000  174416        BIT     #BIT11,@TXCSR   ;TEST THIS BIT
1710  005310  001001                        BNE     .+4     ;BR IF "1"
1711  005312  104004                        ERROR   4                       ;THIS BIT SHOULD BE SET
1712  005314  042777  004000  174404        BIC     #BIT11,@TXCSR   ;CLR THIS BIT
1713  005322  032777  004000  174376        BIT     #BIT11,@TXCSR   ;TEST THIS BIT
1714  005330  001401                        BEQ     .+4             ;BR IF "0"
1715  005332  104004                        ERROR   4                       ;THIS BIT SHOULD BE CLR
1716                                        ;NOW SET THIS BIT
1717  005334  052777  004000  174364        BIS     #BIT11,@TXCSR
1718  005342  052777  000400  174356        BIS     #MRESET,@TXCSR  ;MASTER RESET
1719  005350  032777  004000  174350        BIT     #BIT11,@TXCSR   ;TEST THIS BIT
1720  005356  001401                        BEQ     .+4             ;BR IF "0"
1721  005360  104004                        ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1722
1723                                        ;;TEST MAINT MODE BIT 1
1724                                        ;;
1725                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
1726                                        ;;TESTING OF READ/WRITE BIT BIT12
1727                                        ;;
1728                                ;;**************************************************************
1729  005362  000004                TST24:  SCOPE
1730  005364  052777  010000  174334        BIS     #BIT12,@TXCSR   ;SET THIS BIT
1731  005372  032777  010000  174326        BIT     #BIT12,@TXCSR   ;TEST THIS BIT
1732  005400  001001                        BNE     .+4     ;BR IF "1"
1733  005402  104004                        ERROR   4                       ;THIS BIT SHOULD BE SET
```

```
1734  005404  042777  010000  174314          BIC     #BIT12,@TXCSR   ;CLR THIS BIT
1735  005412  032777  010000  174306          BIT     #BIT12,@TXCSR   ;TEST THIS BIT
1736  005420  001401                          BEQ     .+4             ;BR IF "0"
1737  005422  104004                          ERROR   4                       ;THIS BIT SHOULD BE CLR
1738                                          ;NOW SET THIS BIT
1739  005424  052777  010000  174274          BIS     #BIT12,@TXCSR
1740  005432  052777  000400  174266          BIS     #MRESET,@TXCSR  ;MASTER RESET
1741  005440  032777  010000  174260          BIT     #BIT12,@TXCSR   ;TEST THIS BIT
1742  005446  001401                          BEQ     .+4             ;BR IF "0"
1743  005450  104004                          ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1744
1745                                          ;;THIS TEST PERFORMS MASTER RESET TESTING &
1746                                          ;;TESTING OF READ/WRITE BIT CLK
1747                                          ;;
1748                                  ;;**************************************************************************
1749  005452  000004                  TST25:  SCOPE
1750  005454  052777  020000  174244          BIS     #CLK,@TXCSR     ;SET THIS BIT
1751  005462  032777  020000  174236          BIT     #CLK,@TXCSR     ;TEST THIS BIT
1752  005470  001001                          BNE     .+4     ;BR IF "1"
1753  005472  104004                          ERROR   4                       ;THIS BIT SHOULD BE SET
1754  005474  042777  020000  174224          BIC     #CLK,@TXCSR     ;CLR THIS BIT
1755  005502  032777  020000  174216          BIT     #CLK,@TXCSR     ;TEST THIS BIT
1756  005510  001401                          BEQ     .+4             ;BR IF "0"
1757  005512  104004                          ERROR   4                       ;THIS BIT SHOULD BE CLR
1758                                          ;NOW SET THIS BIT
1759  005514  052777  020000  174204          BIS     #CLK,@TXCSR
1760  005522  052777  000400  174176          BIS     #MRESET,@TXCSR  ;MASTER RESET
1761  005530  032777  020000  174170          BIT     #CLK,@TXCSR     ;TEST THIS BIT
1762  005536  001401                          BEQ     .+4             ;BR IF "0"
1763  005540  104004                          ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1764
1765                                          ;;THIS TEST PERFORMS MASTER RESET TESTING &
1766                                          ;;TESTING OF READ/WRITE BIT MTDATA
1767                                          ;;
1768                                  ;;**************************************************************************
1769  005542  000004                  TST26:  SCOPE
1770  005544  052777  040000  174154          BIS     #MTDATA,@TXCSR  ;SET THIS BIT
1771  005552  032777  040000  174146          BIT     #MTDATA,@TXCSR  ;TEST THIS BIT
1772  005560  001001                          BNE     .+4     ;BR IF "1"
1773  005562  104004                          ERROR   4                       ;THIS BIT SHOULD BE SET
1774  005564  042777  040000  174134          BIC     #MTDATA,@TXCSR  ;CLR THIS BIT
1775  005572  032777  040000  174126          BIT     #MTDATA,@TXCSR  ;TEST THIS BIT
1776  005600  001401                          BEQ     .+4             ;BR IF "0"
1777  005602  104004                          ERROR   4                       ;THIS BIT SHOULD BE CLR
1778                                          ;NOW SET THIS BIT
1779  005604  052777  040000  174114          BIS     #MTDATA,@TXCSR
1780  005612  052777  000400  174106          BIS     #MRESET,@TXCSR  ;MASTER RESET
1781  005620  032777  040000  174100          BIT     #MTDATA,@TXCSR  ;TEST THIS BIT
1782  005626  001401                          BEQ     .+4             ;BR IF "0"
1783  005630  104004                          ERROR   4                       ;CHECK OUT MASTER RESET LOGIC
1784
1785                                  ;;THIS TEST VERIFYS THAT INIT (RESET) CLEARS BITS IN THE
1786                                  ;;RXCSR & TXCSR
1787                                  ;;
1788                                  ;;**************************************************************************
1789  005632  000004                  TST27:  SCOPE
```

```
DZDUQ-A MACY11 27(1006)  03-FEB-77  07:44  PAGE 40
DZDUQA.M11    27-JAN-77 09:46          INITIALIZE THE COMMON TAGS

 1790  005634  012777  177777  174050          MOV     #177777,@RXCSR  ;SET ALL POSSIBLE BITS
 1791  005642  012777  177777  174056          MOV     #177777,@TXCSR  ;DITTO
 1792  005650  000005                          RESET
 1793  005652  106427  000340                  MTPS    #340    ;RESTORE NON INTERRUPT STATUS
 1794  005656  017701  174030                  MOV     @RXCSR,R1       ;SAVE
 1795  005662  017702  174040                  MOV     @TXCSR,R2       ;SAVE
 1796  005666  105767  173257                  TSTB    OPTCLR  ;IS THE OPTIONAL CLR JUMPER ON ?
 1797  005672  100402                          BMI     1$      ;YES
 1798  005674  042701  000016                  BIC     #16,R1  ;CLR THE NON RESETABLE BITS
 1799  005700  042701  073000          1$:     BIC     #073000,R1      ;CLR ALL NON-CLEARABLE BITS
 1800  005704  005701                          TST     R1      ;ARE THEY ALL 0 ?
 1801  005706  001401                          BEQ     .+4
 1802  005710  104004                          ERROR   4       ;ALL SPECIFIED BITS SHOULD BE CLEAR
 1803  005712  042702  002200                  BIC     #002200,R2      ;CLEAR ALL NON-CLEARABLE BITS
 1804  005716  005702                          TST     R2      ;ARE THEY ALL 0 ?
 1805  005720  001401                          BEQ     .+4
 1806  005722  104004                          ERROR   4       ;ALL SPECIFIED BITS SHOULD BE CLEAR
 1807                                          ;WAIT FOR CABLE DELAYS
 1808                                          ;*************************************
 1809                                          ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
 1810                                          ;*************************************
 1811  005724  016702  173170                  MOV     HOLD,R2 ;SET DELAY TIME
 1812  005730  005302                          DEC     R2
 1813  005732  001376                          BNE     .-2     ;WAIT THIS TIME
 1814                                          ;OK NOW FALL THRU AND CONTINUE TESTING.....
 1815                                          ;EXIT STAGE LEFT....CHINNNG!
 1816
 1817                                          ;;THIS TEST PERFORMS MASTER RESET TESTING &
 1818                                          ;;TESTING OF WRITE ONLY BIT MRESET
 1819                                          ;;
 1820                                  ;;*****************************************************************
 1821  005734  000004                  TST30:  SCOPE
 1822  005736  052777  000400  173762          BIS     #MRESET,@TXCSR  ;TRY TO SET THIS BIT
 1823  005744  032777  000400  173754          BIT     #MRESET,@TXCSR  ;TEST THIS BIT
 1824  005752  001401                          BEQ     .+4             ;BR IF "0"
 1825  005754  104004                          ERROR   4                       ;THIS BIT SHOULD NOT BE SET
 1826  005756  052777  000400  173742          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1827  005764  032777  000400  173734          BIT     #MRESET,@TXCSR  ;TEST THIS BIT
 1828  005772  001401                          BEQ     .+4             ;BR IF "0"
 1829  005774  104004                          ERROR   4                       ;THIS BIT SHOULD NOT BE SET
 1830                                                                  ;CHECK MASTER RESET LOGIC
 1831
 1832                                  ;;THIS TEST VERIFYS THAT THE RXCSR & TXCSR CAN BE BYTE ADDRESSED (DATOB)
 1833                                  ;;
 1834                                  ;;*****************************************************************
 1835  005776  000004                  TST31:  SCOPE
 1836  006000  052777  000400  173720          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1837  006006  105767  173137                  TSTB    OPTCLR  ;IS THE OPTIONAL CLR JUMPER ON ?
 1838  006012  100405                          BMI     1$      ;YES
 1839  006014  012777  000000  173670          MOV     #0,@RXCSR       ;CLR OUT NON RESETABLE BITS
 1840  006022  005777  173664                  TST     @RXCSR  ;CLR OUT DSC BY READING RXCSR
 1841  006026  152777  000001  173660  1$:     BISB    #BIT0,@RXCSR    ;SET STRIP SYNC  UPPER BYTE
 1842  006034  017701  173652                  MOV     @RXCSR,R1       ;SAVE RXCSR
 1843  006040  022701  000400                  CMP     #400,R1 ;TEST RXCSR
 1844  006044  001401                          BEQ     .+4
 1845  006046  104004                          ERROR   4       ;ONLY STRIP SYNC SHOULD BE SET
```

```
006050  105077  173636                  CLRB    @RXCSR  ;CLR LOWER BYTE
006054  017701  173632                  MOV     @RXCSR,R1       ;SAVE RXCSR
006060  022701  000400                  CMP     #400,R1 ;TEST RXCSR
006064  001401                          BEQ     .+4
006066  104004                          ERROR   4       ;ONLY STRIP SYNC SHOULD BE SET
006070  052777  000400  173630          BIS     #MRESET,@TXCSR  ;MASTER RESET
006076  152777  000040  173624          BISB    #BIT5,@HTXCSR   ;SET MAINT CLK  UPPER BYTE
006104  017701  173616                  MOV     @TXCSR,R1       ;SAVE TXCSR
006110  042701  002000                  BIC     #BITW,R1        ;CLR BIT WINDOW (DEPENDENT
                                                        ;ON H315 CONNECTOR EXISTANCE)
006114  022701  020200                  CMP     #20200,R1       ;TEST TXCSR
006120  001401                          BEQ     .+4
006122  104004                          ERROR   4       ;ONLY MAINT CLK BIT & TXDONE SHOULD BE SET
006124  105077  173576                  CLRB    @TXCSR  ;CLR LOWER BYTE
006130  017701  173572                  MOV     @TXCSR,R1       ;SAVE TXCSR
006134  042701  002000                  BIC     #BITW,R1        ;CLR BIT WINDOW (DITTO)
006140  022701  020200                  CMP     #20200,R1       ;TEST TXCSR
006144  001401                          BEQ     .+4
006146  104004                          ERROR   4       ;ONLY MAINT CLK BIT & TXDONE SHOULD BE SET
                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
                                        ;;TESTING OF READ ONLY BIT BITW
                                        ;;MAINT INTERNAL
                                        ;;
                                ;;*************************************************************
006150  000004                  TST32:  SCOPE
006152  012777  044001  173546          MOV     #MINT!MTDATA!BREAK,@TXCSR       ;SET MAINT INT.,BREAK,
                                                                                ;&MTDATA
006160  032777  002000  173540          BIT     #BITW,@TXCSR    ;TEST BITW
006166  001001                          BNE     .+4
006170  104004                          ERROR   4               ;BIT WINDOW SHOULD BE SET
006172  042777  040000  173526          BIC     #MTDATA,@TXCSR
006200  013702  001132                  MOV     @#1132,R2
006204  005302                  1$:     DEC     R2
006206  001376                          BNE     1$
006210  032777  002000  173510          BIT     #BITW,@TXCSR
006216  001401                          BEQ     .+4
006220  104004                          ERROR   4               ;BIT SHOULD BE CLR
                                        ;NOW SET THE MTDATA
006222  052777  040000  173476          BIS     #MTDATA,@TXCSR
006230  052777  000400  173470          BIS     #MRESET,@TXCSR  ;MASTER RESET
006236  052777  004001  173462          BIS     #MINT!BREAK,@TXCSR
006244  013702  001132                  MOV     @#1132,R2
006250  005302                  2$:     DEC     R2
006252  001376                          BNE     2$
006254  032777  002000  173444          BIT     #BITW,@TXCSR
006262  001401                          BEQ     .+4
006264  104004                          ERROR   4       ;BITW SHOULD BE CLR BY MASTER RESET
                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
                                        ;;TESTING OF READ ONLY BIT BITW
                                        ;;MAINT EXTERNAL
                                        ;;
                                ;;*************************************************************
006266  000004                  TST33:  SCOPE
                                        ;TEST TO SEE IF EXTERNAL MODEM BYPASS CONNECTOR
                                        ;IS ON (H315)....IF "NO" JUMP AROUND TEST
006270  105767  172657                  TSTB    JMRBY
```

```
                                        INITIALIZE THE COMMON TAGS

1902   006274  100036                           BPL     1$      ;IT IS NOT ON
1903   006276  012777  050001  173422           MOV     #MEXT!MTDATA!BREAK,@TXCSR       ;SET MAINT EXT.,BREAK,
1904                                                                                    ;&MTDATA
1905   006304  032777  002000  173414           BIT     #BITW,@TXCSR    ;TEST BITW
1906   006312  001001                           BNE     .+4
1907   006314  104004                           ERROR   4               ;BIT WINDOW SHOULD BE SET
1908   006316  042777  040000  173402           BIC     #MTDATA,@TXCSR
1909   006324  032777  002000  173374           BIT     #BITW,@TXCSR
1910   006332  001401                           BEQ     .+4
1911   006334  104004                           ERROR   4               ;BIT SHOULD BE CLR
1912                                            ;NOW SET THE MTDATA
1913   006336  052777  040000  173362           BIS     #MTDATA,@TXCSR
1914   006344  052777  000400  173354           BIS     #MRESET,@TXCSR  ;MASTER RESET
1915   006352  052777  010001  173346           BIS     #MEXT!BREAK,@TXCSR
1916   006360  032777  002000  173340           BIT     #BITW,@TXCSR
1917   006366  001401                           BEQ     .+4
1918   006370  104004                           ERROR   4       ;BITW SHOULD BE CLR BY MASTER RESET
1919   006372                          1$:
1920
1921                                            ;;THIS TEST PERFORMS MASTER RESET TESTING &
1922                                            ;;TESTING OF READ ONLY BIT RXDONE
1923                                            ;;
1924                                   ;;*****************************************************************
1925   006372  000004                  TST34:   SCOPE
1926   006374  052777  000400  173324           BIS     #MRESET,@TXCSR  ;MASTER RESET
1927   006402  032777  000200  173302           BIT     #RXDONE,@RXCSR  ;TEST THIS BIT
1928   006410  001401                           BEQ     .+4             ;BR IF "0"
1929   006412  104004                           ERROR   4                       ;CHECK MASTER RESET LOGIC
1930                                                            ;OR SHORT ON THIS BIT
1931
1932                                            ;;THIS TEST PERFORMS MASTER RESET TESTING &
1933                                            ;;TESTING OF READ ONLY BIT RECACT
1934                                            ;;
1935                                   ;;*****************************************************************
1936   006414  000004                  TST35:   SCOPE
1937   006416  052777  000400  173302           BIS     #MRESET,@TXCSR  ;MASTER RESET
1938   006424  032777  004000  173260           BIT     #RECACT,@RXCSR  ;TEST THIS BIT
1939   006432  001401                           BEQ     .+4             ;BR IF "0"
1940   006434  104004                           ERROR   4                       ;CHECK MASTER RESET LOGIC
1941                                                            ;OR SHORT ON THIS BIT
1942
1943                                            ;;THIS TEST PERFORMS MASTER RESET TESTING &
1944                                            ;;TESTING OF READ ONLY BIT DSC
1945                                            ;;
1946                                   ;;*****************************************************************
1947   006436  000004                  TST36:   SCOPE
1948   006440  052777  000400  173260           BIS     #MRESET,@TXCSR  ;MASTER RESET
1949   006446  032777  100000  173236           BIT     #DSC,@RXCSR     ;TEST THIS BIT
1950   006454  001401                           BEQ     .+4             ;BR IF "0"
1951   006456  104004                           ERROR   4                       ;CHECK MASTER RESET LOGIC
1952                                                            ;OR SHORT ON THIS BIT
1953
1954                                            ;;THIS TEST PERFORMS MASTER RESET TESTING &
1955                                            ;;TESTING OF READ ONLY BIT TXDONE
1956                                            ;;
1957
```

INITIALIZE THE COMMON TAGS

```
                                ;;*****************************************************************
006460  000004                 TST37:  SCOPE
006462  052777  000400  173236         BIS     #MRESET,@TXCSR  ;MASTER RESET
006470  032777  000200  173230         BIT     #TXDONE,@TXCSR  ;TEST THIS BIT
006476  001001                         BNE     .+4                     ;BR IF "1"
006500  104004                         ERROR   4               ;CHECK MASTER RESET LOGIC
                                                        ;OR SHORT ON THIS BIT
                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
                                        ;;TESTING OF READ ONLY BIT DNA
                                        ;;
                                ;;*****************************************************************
006502  000004                 TST40:  SCOPE
006504  052777  000400  173214         BIS     #MRESET,@TXCSR  ;MASTER RESET
006512  032777  100000  173206         BIT     #DNA,@TXCSR     ;TEST THIS BIT
006520  001401                         BEQ     .+4                     ;BR IF "0"
006522  104004                         ERROR   4                       ;CHECK MASTER RESET LOGIC
                                                        ;OR SHORT ON THIS BIT

                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
                                        ;;TESTING OF READ  ONLY WORD RECEIVE DATA
                                        ;;
                                ;;*****************************************************************
006524  000004                 TST41:  SCOPE
006526  052777  000400  173172         BIS     #MRESET,@TXCSR  ;MASTER RESET
006534  016703  173156                 MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
006540  012700  000377                 MOV     #377,R0         ;EXPECTED
006544  017701  173146                 MOV     @RXDBUF,R1      ;ACTUAL
006550  120001                         CMPB    R0,R1
006552  001401                         BEQ     .+4                     ;BR IF "U"
006554  104002                         ERROR   2               ;REC DATA SHOULD BE ALL 1'S
                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
                                        ;;TESTING OF READ ONLY BIT PARER
                                        ;;
                                ;;*****************************************************************
006556  000004                 TST42:  SCOPE
006560  052777  000400  173140         BIS     #MRESET,@TXCSR  ;MASTER RESET
006566  032777  010000  173122         BIT     #PARER,@RXDBUF  ;TEST THIS BIT
006574  001401                         BEQ     .+4                     ;BR IF "0"
006576  104004                         ERROR   4                       ;CHECK MASTER RESET LOGIC
                                                        ;OR SHORT ON THIS BIT

                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
                                        ;;TESTING OF READ ONLY BIT FRMERR
                                        ;;
                                ;;*****************************************************************
006600  000004                 TST43:  SCOPE
006602  052777  000400  173116         BIS     #MRESET,@TXCSR  ;MASTER RESET
006610  032777  020000  173100         BIT     #FRMERR,@RXDBUF ;TEST THIS BIT
006616  001401                         BEQ     .+4                     ;BR IF "0"
006620  104004                         ERROR   4                       ;CHECK MASTER RESET LOGIC
                                                        ;OR SHORT ON THIS BIT

                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
                                        ;;TESTING OF READ ONLY BIT OVRRUN
                                        ;;
                                ;;*****************************************************************
```

INITIALIZE THE COMMON TAGS

```
                                TST44:  SCOPE
006622  000004
006624  052777  000400  173074          BIS     #MRESET,@TXCSR  ;MASTER RESET
006632  032777  040000  173056          BIT     #OVRRUN,@RXDBUF ;TEST THIS BIT
006640  001401                          BEQ     .+4             ;BR IF "0"
006642  104004                          ERROR   4                       ;CHECK MASTER RESET LOGIC
                                                        ;OR SHORT ON THIS BIT

                                        ;;THIS TEST PERFORMS MASTER RESET TESTING &
                                        ;;TESTING OF READ ONLY BIT RXERR
                                        ;;
                                ;;**************************************************************
006644  000004                  TST45:  SCOPE
006646  052777  000400  173052          BIS     #MRESET,@TXCSR  ;MASTER RESET
006654  032777  100000  173034          BIT     #RXERR,@RXDBUF  ;TEST THIS BIT
006662  001401                          BEQ     .+4             ;BR IF "0"
006664  104004                          ERROR   4                       ;CHECK MASTER RESET LOGIC
                                                        ;OR SHORT ON THIS BIT

                                        ;;THIS TEST VERIFYS THAT THE DEVICE REGISTER RXCSR
                                        ;;IS CLEARED BY MASTER RESET
                                ;;**************************************************************
006666  000004                  TST46:  SCOPE
006670  012777  177777  173014          MOV     #177777,@RXCSR  ;SET ALL POSSIBLE BITS
006676  052777  000400  173022          BIS     #MRESET,@TXCSR  ;MASTER RESET
006704  016703  173002                  MOV     RXCSR,R3        ;FOR ERROR MESSAGE
006710  017701  172776                  MOV     @RXCSR,R1       ;SAVE ACTUAL
006714  105767  172231                  TSTB    OPTCLR          ;TEST THE OPT CLR JUMPER FLAG
006720  100010                          BPL     1$              ;NO ,ITS NOT IN
006722  042701  173000                  BIC     #173000,R1      ;CLR NON-MASTER RESETTABLE
                                                ;BITS(SINCE THESE ARE DEPENDENT ON H315 CONNECTORS EXISTANCE)
006726  012700  000000                  MOV     #0,R0   ;EXPECTED
006732  020001                          CMP     R0,R1   ;EXPECTED VS ACTUAL
006734  001401                          BEQ     .+4
006736  104001                          ERROR   1       ;ALL MASTER RESETABLE BITS SHOULD BE CLR
006740  000407                          BR      2$              ;JUMP AROUND
006742  042701  173000          1$:     BIC     #173000,R1      ;CLR NON-MASTER RESETABLE
                                                ;BITS(SINCE THESE ARE DEPENDENT ON H315 CONNECTORS EXISTANCE)
006746  012700  000016                  MOV     #16,R0          ;EXPECTED
006752  020001                          CMP     R0,R1   ;EXPECTED VS ACTUAL
006754  001401                          BEQ     .+4
006756  104001                          ERROR   1       ;ONLY STD,RTS,DTR BITS SHOULD BE SET
                                                ;NOTE THAT STD IS READ =1 INDEPENDENT OF
                                                ;SEC XMIT #6 STRAP
006760                          2$:
                                ;WAIT FOR CABLE DELAYS
                                ;**************************************
                                ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
                                ;**************************************
006760  016702  172134          MOV     HOLD,R2 ;SET DELAY TIME
006764  005302                  DEC     R2
006766  001376                  BNE     .-2     ;WAIT THIS TIME
                                ;OK NOW FALL THRU AND CONTINUE TESTING.....
                                ;EXIT STAGE LEFT....CHINNNG!


                                ;;THIS TEST VERIFYS THAT THE DEVICE REGISTER TXCSR
```

```
                                                ;;IS CLEARED BY MASTER RESET
                                                ;;
                                        ;;**************************************************************************
                                        TST47:  SCOPE
007770  000004                                  
007772  012777  177777  172726                   MOV     #177777,@TXCSR  ;SET ALL POSSIBLE BITS
007000  052777  000400  172720                   BIS     #MRESET,@TXCSR  ;MASTER RESET
007006  016703  172714                           MOV     TXCSR,R3        ;FOR ERROR MESSAGE
007012  017701  172710                           MOV     @TXCSR,R1       ;SAVE ACTUAL
007016  012700  000200                           MOV     #200,R0 ;EXPECTED
007022  020001                                   CMP     R0,R1   ;EXPECTED VS ACTUAL
007024  001401                                   BEQ     .+4
007026  104001                                   ERROR   1       ;ONLY TXDONE SHOULD BE SET
                                                
                                                ;;THIS TEST VERIFYS THAT THE DEVICE REGISTER RXDBUF
                                                ;;IS CLEARED BY MASTER RESET
                                                ;;
                                        ;;**************************************************************************
                                        TST50:  SCOPE
007030  000004                                  
007032  052777  000400  172666                   BIS     #MRESET,@TXCSR  ;MASTER RESET
007040  016703  172652                           MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
007044  017701  172646                           MOV     @RXDBUF,R1      ;SAVE
007050  012700  000377                           MOV     #377,R0 ;EXPECTED
007054  020001                                   CMP     R0,R1   ;EXPECTED VS ACTUAL
007056  001401                                   BEQ     .+4
007060  104002                                   ERROR   2       ;ONLY REC DATA BITS SHOULD BE SET
                                                ;;THIS TEST VERIFYS BITS RING,CTS,CARDET,SRD,DSR
                                                ;;ALSO DSC IS GENERATED WHEN ANY OF THESE BITS ARE SET
                                                ;;OR CLEARED.....IT ALSO CHECKS THE MODEM BYPASS
                                                ;;JUMPER AND THAT THESE BITS CAN BE READ
                                                ;;NOTE: THE MODEM BYPASS JUMPER MUST BE ON (H315)
                                                ;;
                                        ;;**************************************************************************
                                        TST51:  SCOPE
007062  000004                                  
007064  005077  172622                           CLR     @RXCSR  ;TO GET RID OF STD ,RTS,DTR IF OPTCLR JUMPER #4 IS NOT ON
007070  052777  000400  172630                   BIS     #MRESET,@TXCSR  ;MASTER RESET
                                                ;TEST THAT A "YES" ANSWER WAS GIVEN TO QUESTION IN
                                                ;THE MONITOR OR BY DEFAULT
                                                ;THIS TEST WILL BE BYPASSED  IF THE EXTERNAL BYPASS
                                                ;JUMPER IS NOT INSTALLED
007076  105767  172051                           TSTB    JMRBY
007102  100402                                   BMI     .+6     ;THE ANSWER WAS YES.....
                                                                ; PERFORM THIS TEST
007104  000167  000652                           JMP     OUT1    ;JUMP AROUND THIS TEST IF THE ANSWER
                                                                ;WAS NO
007110  016703  172576                           MOV     RXCSR,R3        ;SET UP FOR ERROR MESSAGE
007114  017701  172572                           MOV     @RXCSR,R1       ;ACTUAL
007120  005000                                   CLR     R0      ;EXPECTED
007122  005701                                   TST     R1      ;IS IT = 0 ?
007124  001401                                   BEQ     .+4
007126  104001                                   ERROR   1       ;RXCSR SHOULD BE CLR
007130  052777  000002  172554                   BIS     #DTR,@RXCSR     ;SET DTR
                                                ;WAIT FOR CABLE DELAYS
                                                ;*************************************
                                                ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
                                                ;*************************************
007136  016702  171756                           MOV     HOLD,R2 ;SET DELAY TIME
```

```
 2126  007142  005302                          DEC     R2
 2127  007144  001376                          BNE     .-2     ;WAIT THIS TIME
 2128                                          ;OK NOW FALL THRU AND CONTINUE TESTING.....
 2129                                          ;EXIT STAGE LEFT....CHINNING!
 2130  007146  017701  172540                  MOV     @RXCSR,R1       ;ACTUAL
 2131  007152  012700  130002                  MOV     #130002,R0      ;DSC,CTS,CARDET,DTR
 2132  007156  020001                          CMP     R0,R1   ;EXPECTED VS ACTUAL
 2133  007160  001401                          BEQ     .+4
 2134  007162  104001                          ERROR   1       ;CHECK BYPASS CONNECTOR
 2135  007164  017701  172522                  MOV     @RXCSR,R1       ;ACTUAL
 2136  007170  012700  030002                  MOV     #30002,R0       ;CTS,CARDET,DTR
 2137  007174  020001                          CMP     R0,R1   ;EXPECTED VS ACTUAL
 2138  007176  001401                          BEQ     .+4
 2139  007200  104001                          ERROR   1       ;PREVIOUS READING OF RXCSR SHOULD
 2140                                                          ;HAVE CLEARED DSC
 2141  007202  052777  000004  172502          BIS     #RTS,@RXCSR
 2142                                          ;WAIT FOR CABLE DELAYS
 2143                                          ;*************************************
 2144                                          ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
 2145                                          ;*************************************
 2146  007210  016702  171704                  MOV     HOLD,R2 ;SET DELAY TIME
 2147  007214  005302                          DEC     R2
 2148  007216  001376                          BNE     .-2     ;WAIT THIS TIME
 2149                                          ;OK NOW FALL THRU AND CONTINUE TESTING.....
 2150                                          ;EXIT STAGE LEFT....CHINNING!
 2151  007220  017701  172466                  MOV     @RXCSR,R1
 2152  007224  012700  170006                  MOV     #170006,R0      ;DSC,RING,CTS,CARDET,RTS,DTR
 2153  007230  020001                          CMP     R0,R1   ;EXPECTED VS ACTUAL
 2154  007232  001401                          BEQ     .+4
 2155  007234  104001                          ERROR   1       ;CHECK BYPASS CONNECTOR
 2156  007236  017701  172450                  MOV     @RXCSR,R1
 2157  007242  012700  070006                  MOV     #70006,R0       ;RING,CTS,CARDET,RTS,DTR
 2158  007246  020001                          CMP     R0,R1   ;EXPECTED VS ACTUAL
 2159  007250  001401                          BEQ     .+4
 2160  007252  104001                          ERROR   1       ;PREVIOUS READING OF RXCSR SHOULD
 2161                                                          ;HAVE CLEARED DSC
 2162  007254  105767  171667                  TSTB    SEXMIT  ;IS SEC XMIT JUMPER IN ?
 2163  007260  100112                          BPL     OUT2    ;NO
 2164  007262  105767  171662                  TSTB    SEREC   ;IS SEC REC JUMPER IN ?
 2165  007266  100163                          BPL     OUT3    ;NO
 2166  007270  052777  000010  172414          BIS     #STD,@RXCSR
 2167                                          ;WAIT FOR CABLE DELAYS
 2168                                          ;*************************************
 2169                                          ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
 2170                                          ;*************************************
 2171  007276  016702  171616                  MOV     HOLD,R2 ;SET DELAY TIME
 2172  007302  005302                          DEC     R2
 2173  007304  001376                          BNE     .-2     ;WAIT THIS TIME
 2174                                          ;OK NOW FALL THRU AND CONTINUE TESTING.....
 2175                                          ;EXIT STAGE LEFT....CHINNING!
 2176  007306  017701  172400                  MOV     @RXCSR,R1
 2177  007312  012700  173016          MASK1:  MOV     #173016,R0      ;DSC,RING,CTS,CARDET,SRD,DSR
 2178                                                          ;STD,RTS,DTR
 2179  007316  020001                          CMP     R0,R1   ;EXPECTED VS ACTUAL
 2180  007320  001401                          BEQ     .+4
 2181  007322  104001                          ERROR   1       ;CHECK BYPASS CONNECTOR
```

```
2182  007324  017701  172362              MOV     @RXCSR,R1
2183  007330  012700  073016      MASK2:  MOV     #73016,R0       ;RING,CTS,CARDET,SRD,DSR,STD
2184                                                              ;RTS,DTR
2185  007334  020001                      CMP     R0,R1   ;EXPECTED VS ACTUAL
2186  007336  001401                      BEQ     .+4
2187  007340  104001                      ERROR   1       ;PREVIOUS READING OF RXCSR SHOULD
2188                                                      ;HAVE CLEARED DSC
2189  007342  042777  000002  172342      BIC     #DTR,@RXCSR
2190                                      ;WAIT FOR CABLE DELAYS
2191                                      ;*************************************
2192                                      ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
2193                                      ;*************************************
2194  007350  016702  171544              MOV     HOLD,R2 ;SET DELAY TIME
2195  007354  005302                      DEC     R2
2196  007356  001376                      BNE     .-2     ;WAIT THIS TIME
2197                                      ;OK NOW FALL THRU AND CONTINUE TESTING.....
2198                                      ;EXIT STAGE LEFT....CHINNNG!
2199  007360  017701  172326              MOV     @RXCSR,R1
2200  007364  012700  143014              MOV     #143014,R0      ;DSC,RING,SRD,DSR,STD,RTS
2201  007370  020001                      CMP     R0,R1   ;EXPECTED VS ACTUAL
2202  007372  001401                      BEQ     .+4
2203  007374  104001                      ERROR   1       ;DSC SHOULD BE SET
2204  007376  042777  000004  172306      BIC     #RTS,@RXCSR
2205                                      ;WAIT FOR CABLE DELAYS
2206                                      ;*************************************
2207                                      ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
2208                                      ;*************************************
2209  007404  016702  171510              MOV     HOLD,R2 ;SET DELAY TIME
2210  007410  005302                      DEC     R2
2211  007412  001376                      BNE     .-2     ;WAIT THIS TIME
2212                                      ;OK NOW FALL THRU AND CONTINUE TESTING.....
2213                                      ;EXIT STAGE LEFT....CHINNNG!
2214  007414  017701  172272              MOV     @RXCSR,R1
2215  007420  012700  103010      MASK3:  MOV     #103010,R0      ;DSC,SRD,DSR,STD
2216  007424  020001                      CMP     R0,R1   ;EXPECTED VS ACTUAL
2217  007426  001401                      BEQ     .+4
2218  007430  104001                      ERROR   1       ;DSC SHOULD BE SET
2219  007432  042777  000010  172252      BIC     #STD,@RXCSR
2220                                      ;WAIT FOR CABLE DELAYS
2221                                      ;*************************************
2222                                      ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
2223                                      ;*************************************
2224  007440  016702  171454              MOV     HOLD,R2 ;SET DELAY TIME
2225  007444  005302                      DEC     R2
2226  007446  001376                      BNE     .-2     ;WAIT THIS TIME
2227                                      ;OK NOW FALL THRU AND CONTINUE TESTING.....
2228                                      ;EXIT STAGE LEFT....CHINNNG!
2229  007450  017701  172236              MOV     @RXCSR,R1
2230  007454  012700  100000              MOV     #100000,R0      ;DSC
2231  007460  020001                      CMP     R0,R1   ;EXPECTED VS ACTUAL
2232  007462  001401                      BEQ     .+4
2233  007464  104001                      ERROR   1       ;DSC SHOULD BE SET
2234  007466  017701  172220              MOV     @RXCSR,R1
2235  007472  005000                      CLR     R0      ;NONE
2236  007474  005701                      TST     R1
2237  007476  001401                      BEQ     .+4
```

```
                                        INITIALIZE THE COMMON TAGS

 2238  007500  104001                          ERROR   1       ;DSC SHOULD BE CLEARED FROM PREVIOUS
 2239                                          ;READING OF RXCSR
 2240  007502  000167  000254                  JMP     OUT1    ;JUMP AROUND
 2241                                  ;THE FOLLOWING ROUTINE HANDLES THE SITUATION WHERE SEC XMIT
 2242                                  ;AND SEC REC JUMPERS ARE NOT ON
 2243  007506  052777  000010  172176  OUT2:   BIS     #STD,@RXCSR
 2244                                          ;WAIT FOR CABLE DELAYS
 2245                                          ;#######################################
 2246                                          ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
 2247                                          ;#######################################
 2248  007514  016702  171400                  MOV     HOLD,R2 ;SET DELAY TIME
 2249  007520  005302                          DEC     R2
 2250  007522  001376                          BNE     .-2     ;WAIT THIS TIME
 2251                                          ;OK NOW FALL THRU AND CONTINUE TESTING.....
 2252                                          ;EXIT STAGE LEFT....CHINNING!
 2253  007524  017701  172162                  MOV     @RXCSR,R1       ;ACTUAL
 2254  007530  012700  070016                  MOV     #70016,R0       ;EXPECTED: RING ,CTS,CARDET,STD,RTS,DTR
 2255  007534  020001                          CMP     R0,R1   ;EXPECTED VS ACTUAL
 2256  007536  001401                          BEQ     .+4
 2257  007540  104001                          ERROR   1       ;CHECK SEC XMIT & SEC REC JUMPERS
 2258  007542  042777  000004  172142          BIC     #RTS,@RXCSR
 2259                                          ;WAIT FOR CABLE DELAYS
 2260                                          ;#######################################
 2261                                          ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
 2262                                          ;#######################################
 2263  007550  016702  171344                  MOV     HOLD,R2 ;SET DELAY TIME
 2264  007554  005302                          DEC     R2
 2265  007556  001376                          BNE     .-2     ;WAIT THIS TIME
 2266                                          ;OK NOW FALL THRU AND CONTINUE TESTING.....
 2267                                          ;EXIT STAGE LEFT....CHINNING!
 2268  007560  017701  172126                  MOV     @RXCSR,R1       ;ACTUAL
 2269  007564  012700  130012                  MOV     #130012,R0      ;DSC,CTS,CARDET,DTR,STD
 2270                                  ;NOTE THAT DSC STILL ASSERTS EVEN THO THE SEC XMIT JUMPER # 6 IS NOT ON
 2271  007570  020001                          CMP     R0,R1   ;EXPECTED VS ACTUAL
 2272  007572  001401                          BEQ     .+4
 2273  007574  104001                          ERROR   1       ;CHECK BYPASS CONNECTOR
 2274  007576  042777  000002  172106          BIC     #DTR,@RXCSR
 2275                                          ;WAIT FOR CABLE DELAYS
 2276                                          ;#######################################
 2277                                          ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
 2278                                          ;#######################################
 2279  007604  016702  171310                  MOV     HOLD,R2 ;SET DELAY TIME
 2280  007610  005302                          DEC     R2
 2281  007612  001376                          BNE     .-2     ;WAIT THIS TIME
 2282                                          ;OK NOW FALL THRU AND CONTINUE TESTING.....
 2283                                          ;EXIT STAGE LEFT....CHINNING!
 2284  007614  017701  172072                  MOV     @RXCSR,R1       ;ACTUAL
 2285  007620  012700  100010                  MOV     #100010,R0      ;DSC,STD
 2286  007624  020001                          CMP     R0,R1   ;EXPECTED VS ACTUAL
 2287  007626  001401                          BEQ     .+4
 2288  007630  104001                          ERROR   1       ;ONLY DSC & STD  SHOULD BE SET
 2289  007632  000167  000124                  JMP     OUT1    ;JUMP AROUND
 2290  007636  052777  000010  172046  OUT3:   BIS     #STD,@RXCSR
 2291                                          ;WAIT FOR CABLE DELAYS
 2292                                          ;#######################################
 2293                                          ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
```

```
2294
2295  007644  016702  171250                ;****************************************
2296  007650  005302                        MOV     HOLD,R2 ;SET DELAY TIME
2297  007652  001376                        DEC     R2
2298                                        BNE     .-2     ;WAIT THIS TIME
2299                                        ;OK NOW FALL THRU AND CONTINUE TESTING.....
2300  007654  017701  172032                ;EXIT STAGE LEFT....CHINNNG!
2301  007660  012700  171016                MOV     @RXCSR,R1       ;ACTUAL
2302  007664  020001                        MOV     #171016,R0      ;EXPECTED: DSC,RING,CTS,CARDET,DSR,STD,RTS,DTR
2303  007666  001401                        CMP     R0,R1   ;EXPECTED VS ACTUAL
2304  007670  104001                        BEQ     .+4
2305  007672  042777  000004  172012        ERROR   1       ;CHECK SEC REC JUMPER
2306                                        BIC     #RTS,@RXCSR
2307                                        ;WAIT FOR CABLE DELAYS
2308                                        ;****************************************
2309                                        ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
2310  007700  016702  171214                ;****************************************
2311  007704  005302                        MOV     HOLD,R2 ;SET DELAY TIME
2312  007706  001376                        DEC     R2
2313                                        BNE     .-2     ;WAIT THIS TIME
2314                                        ;OK NOW FALL THRU AND CONTINUE TESTING.....
2315  007710  017701  171776                ;EXIT STAGE LEFT....CHINNNG!
2316  007714  012700  131012                MOV     @RXCSR,R1       ;ACTUAL
2317  007720  020001                        MOV     #131012,R0      ;EXPECTED: DSC,CTS,CARDET,DSR,STD,DTR
2318  007722  001401                        CMP     R0,R1   ;EXPECTED VS ACTUAL
2319  007724  104001                        BEQ     .+4
2320  007726  042777  000002  171756        ERROR   1       ;CHECK H315 CONNECTOR
2321                                        BIC     #DTR,@RXCSR
2322                                        ;WAIT FOR CABLE DELAYS
2323                                        ;****************************************
2324                                        ;MODIFY "HOLD:"  ACCORDINGLY FOR FASTER OR SLOWER  MACHINE
2325  007734  016702  171160                ;****************************************
2326  007740  005302                        MOV     HOLD,R2 ;SET DELAY TIME
2327  007742  001376                        DEC     R2
2328                                        BNE     .-2     ;WAIT THIS TIME
2329                                        ;OK NOW FALL THRU AND CONTINUE TESTING.....
2330  007744  017701  171742                ;EXIT STAGE LEFT....CHINNNG!
2331  007750  012700  101010                MOV     @RXCSR,R1       ;ACTUAL
2332  007754  020001                        MOV     #101010,R0      ;EXPECTED: DSC,DSR,STD
2333  007756  001401                        CMP     R0,R1   ;EXPECTED VS ACTUAL
2334  007760  104001                        BEQ     .+4
2335  007762                                ERROR   1       ;CHECK H315 CONNECTOR
                                    OUT1:
2336
2337                                        ;;THIS TEST VERIFYS THAT RECACT (REC ACTIVE) ASSERTS
2338                                        ;;IMMED. WHEN SYNC EXTERNAL MODE IS SELECTED
2339                                        ;;AND SYNC SEARCH IS SET
2340                                        ;;
2341                                    ;;**************************************************************************
2342  007762  000004                    TST52:  SCOPE
2343  007764  052777  000400  171734            BIS     #MRESET,@TXCSR  ;MASTER RESET
2344  007772  012777  020000  171722            MOV     #SYNEXT,@PARCSR ;SET THE MODE
2345  010000  052777  000400  171720            BIS     #MRESET,@TXCSR  ;MASTER RESET
2346
2347                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2348  010006  012777  064001  171712            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2349
```

INITIALIZE THE COMMON TAGS

```
                                ;SET MODE ,# OF BITS,PARITY SENSE &LOAD SYNC REG
010014  012777  026026  171700          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
010022  032777  004000  171662          BIT     #RECACT,@RXCSR
010030  001401                          BEQ     .+4
010032  104004                          ERROR   4
010034  052777  000020  171650          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
010042  032777  004000  171642          BIT     #RECACT,@RXCSR
010050  001001                          BNE     .+4
010052  104004                          ERROR   4               ;RECACT DID NOT ASSERT
010054  042777  000020  171630          BIC     #SYNSCH,@RXCSR  ;DROP SEARCH SYNC
010062  032777  004000  171622          BIT     #RECACT,@RXCSR  ;IS IT =0?
010070  001401                          BEQ     .+4
010072  104004                          ERROR   4       ;RECACT SHOULD BE 0

                                ;;THIS TEST VERIFYS THAT RECACT (REC ACTIVE) ASSERTS
                                ;;IMMED. WHEN ISOCRONOUS MODE IS SELECTED
                                ;;AND SYNC SEARCH IS SET
                                ;;
                        ;;**************************************************************
010074  000004          TST53:  SCOPE
010076  052777  000400  171622          BIS     #MRESET,@TXCSR  ;MASTER RESET
010104  012777  000000  171610          MOV     #ISYMOD,@PARCSR ;SET THE MODE
010112  052777  000400  171606          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
010120  012777  064001  171600          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                ;SET MODE ,# OF BITS,PARITY SENSE &LOAD SYNC REG
010126  012777  006026  171566          MOV     #ISYMOD!EIGHT!NOPAR!26,@PARCSR
010134  032777  004000  171550          BIT     #RECACT,@RXCSR
010142  001401                          BEQ     .+4
010144  104004                          ERROR   4               ;RECACT SHOULD NOT BE SET
010146  052777  000020  171536          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
010154  032777  004000  171530          BIT     #RECACT,@RXCSR
010162  001001                          BNE     .+4
010164  104004                          ERROR   4               ;RECACT DID NOT ASSERT
010166  042777  000020  171516          BIC     #SYNSCH,@RXCSR  ;DROP SEARCH SYNC
010174  032777  004000  171510          BIT     #RECACT,@RXCSR  ;IS IT =0?
010202  001401                          BEQ     .+4
010204  104004                          ERROR   4       ;RECACT SHOULD BE 0

                                ;;VERIFY THE MATCH DETECT & DATA RDY FLAGS BY PUMPING
                                ;;IN TWO # SYNC CHARS THRU MAINT DATA BIT
                                ;;WATCH THE RECACT BIT
                                ;;ON THE THIRD # CHARACTER IT SHOULD SET RXDONE
                                ;;#: DEPENDENT ON MONITOR.............
                                ;;IF ONE SYNC STRAP IS SELECTED THEN IT WILL ONLY
                                ;;TAKE ONE SYNC CHARACTER FOR RXDONE TO ASSERT
                                ;;ON THE SECOND CHARACTER
                                ;;ALSO CHECK THIS CHARACTER IN RXDBUF
                                ;;AND CHECK OPERATION OF SYNSCH
                                ;;MODE: SYNC INTERNAL
                                ;;LENGTH:FIVE
                                ;;
                        ;;**************************************************************
010206  000004          TST54:  SCOPE
```

```
DZDUQ-A MACY11 27(1006)  03-FEB-77  07:44  PAGE 51
DZDUQA.M11    27-JAN-77 09:46          INITIALIZE THE COMMON TAGS

 2406  010210  052777  000400  171510          BIS     #MRESET,@TXCSR  ;MASTER RESET
 2407  010216  012777  030000  171476          MOV     #SYNINT,@PARCSR ;SET THE MODE
 2408  010224  052777  000400  171474          BIS     #MRESET,@TXCSR  ;MASTER RESET
 2409
 2410                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 2411  010232  012777  064001  171466          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 2412
 2413                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 2414  010240  012777  030026  171454          MOV     #SYNINT!FIVE!NOPAR!26,@PARCSR
 2415  010246  016703  171444                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 2416  010252  052777  000020  171432          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 2417                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 2418  010260  042777  020000  171440          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 2419  010266  052777  020000  171432          BIS     #CLK,@TXCSR     ;POKE CLK UP
 2420                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 2421  010274  042777  020000  171424          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 2422  010302  052777  020000  171416          BIS     #CLK,@TXCSR     ;POKE CLK UP
 2423  010310  012767  000002  170606          MOV     #2,COUNT
 2424  010316  012767  000005  170576  1$:     MOV     #5,SHIFT        ;# OF SHIFTS
 2425  010324  012767  000026  171146          MOV     #26,STMP1       ;SYNC CHARACTER
 2426  010332  004767  006540                  JSR     PC,RPOKE
 2427  010336  005367  170562                  DEC     COUNT
 2428  010342  001403                          BEQ     2$
 2429                                          ;TEST SYNCNO TO SEE HOW MANY SYNC CHARS WERE SELECTED
 2430  010344  105767  170576                  TSTB    SYNCNO
 2431  010350  100762                          BMI     1$              ;TWO SYNC CHARS
 2432  010352  105777  171334          2$:     TSTB    @RXCSR  ;CHECK REC DONE BIT
 2433  010356  100001                          BPL     .+4
 2434  010360  104004                          ERROR   4               ;RXDONE SHOULD NOT BE ASSERTED
 2435  010362  032777  004000  171322          BIT     #RECACT,@RXCSR
 2436  010370  001001                          BNE     .+4
 2437  010372  104004                          ERROR   4               ;RECACT SHOULD BE ASSERTED
 2438  010374  012767  000005  170520          MOV     #5,SHIFT
 2439  010402  012767  000021  171070          MOV     #21,STMP1       ;ANY CHARACTER
 2440  010410  004767  006462                  JSR     PC,RPOKE
 2441  010414  105777  171272                  TSTB    @RXCSR  ;CHECK RXDONE
 2442  010420  100401                          BMI     .+4
 2443  010422  104004                          ERROR   4               ;RXDONE SHOULD BE ASSERTED
 2444  010424  032777  004000  171260          BIT     #RECACT,@RXCSR
 2445  010432  001001                          BNE     .+4
 2446  010434  104004                          ERROR   4               ;RECACT SHOULD STILL BE ASSERTED
 2447  010436  042777  000020  171246          BIC     #SYNSCH,@RXCSR  ;CLR SYNC SEARCH
 2448  010444  032777  004000  171240          BIT     #RECACT,@RXCSR  ;IT SHOULD DROP IMMEDIATELY
 2449  010452  001401                          BEQ     .+4
 2450  010454  104004                          ERROR   4               ;RECACT SHOULD BE CLR
 2451  010456  105777  171230                  TSTB    @RXCSR  ;RXDONE
 2452  010462  100401                          BMI     .+4
 2453  010464  104004                          ERROR   4               ;RXDONE SHOULD STILL BE ASSERTED
 2454  010466  012700  000021                  MOV     #21,R0  ;EXPECTED DATA
 2455  010472  017701  171220                  MOV     @RXDBUF,R1      ;ACTUAL DATA
 2456  010476  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
 2457  010500  001401                          BEQ     .+4
 2458  010502  104002                          ERROR   2               ;DATA CHARS SHOULD COMPARE
 2459  010504  105777  171202                  TSTB    @RXCSR  ;CHECK RXDONE
 2460  010510  100001                          BPL     .+4
 2461  010512  104004                          ERROR   4               ;RXDONE SHOULD BE CLR FROM
```

```
2462                                                    ;PREVIOUS READING OF RXDBUF
2463
2464                                                ;;VERIFY THE MATCH DETECT & DATA RDY FLAGS BY PUMPING
2465                                                ;;IN TWO # SYNC CHARS THRU MAINT DATA BIT
2466                                                ;;WATCH THE RECACT BIT
2467                                                ;;ON THE THIRD # CHARACTER IT SHOULD SET RXDONE
2468                                                ;;#: DEPENDENT ON MONITOR.............
2469                                                ;;IF ONE SYNC STRAP IS SELECTED THEN IT WILL ONLY
2470                                                ;;TAKE ONE SYNC CHARACTER FOR RXDONE TO ASSERT
2471                                                ;;ON THE SECOND CHARACTER
2472                                                ;;ALSO CHECK THIS CHARACTER IN RXDBUF
2473                                                ;;AND CHECK OPERATION OF SYNSCH
2474                                                ;;MODE: SYNC INTERNAL
2475                                                ;;LENGTH:SIX
2476                                                ;;
2477                                            ;;*****************************************************************
2478  010514  000004                            TSTSS:  SCOPE
2479  010516  052777  000400  171202                    BIS     #MRESET,@TXCSR  ;MASTER RESET
2480  010524  012777  030000  171170                    MOV     #SYNINT,@PARCSR ;SET THE MODE
2481  010532  052777  000400  171166                    BIS     #MRESET,@TXCSR  ;MASTER RESET
2482
2483                                            ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2484  010540  012777  064001  171160                    MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2485
2486                                            ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2487  010546  012777  032026  171146                    MOV     #SYNINT!SIX!NOPAR!26,@PARCSR
2488  010554  016703  171136                           MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2489  010560  052777  000020  171124                    BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2490                                                    ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2491  010566  042777  020000  171132                    BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2492  010574  052777  020000  171124                    BIS     #CLK,@TXCSR     ;POKE CLK UP
2493                                            ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2494  010602  042777  020000  171116                    BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2495  010610  052777  020000  171110                    BIS     #CLK,@TXCSR     ;POKE CLK UP
2496  010616  012767  000002  170300                    MOV     #2,COUNT
2497  010624  012767  000006  170270  1$:               MOV     #6,SHIFT        ;# OF SHIFTS
2498  010632  012767  000026  170640                    MOV     #26,$TMP1       ;SYNC CHARACTER
2499  010640  004767  006232                           JSR     PC,RPOKE
2500  010644  005367  170254                           DEC     COUNT
2501  010650  001403                                   BEQ     2$
2502                                                    ;TEST SYNCNO TO SEE HOW MANY SYNC CHARS WERE SELECTED
2503  010652  105767  170270                           TSTB    SYNCNO
2504  010656  100762                                   BMI     1$      ;TWO SYNC CHARS
2505  010660  105777  171026          2$:              TSTB    @RXCSR  ;CHECK REC DONE BIT
2506  010664  100001                                   BPL     .+4
2507  010666  104004                                   ERROR   4               ;RXDONE SHOULD NOT BE ASSERTED
2508  010670  032777  004000  171014                    BIT     #RECACT,@RXCSR
2509  010676  001001                                   BNE     .+4
2510  010700  104004                                   ERROR   4               ;RECACT SHOULD BE ASSERTED
2511  010702  012767  000006  170212                    MOV     #6,SHIFT
2512  010710  012767  000021  170562                    MOV     #21,$TMP1       ;ANY CHARACTER
2513  010716  004767  006154                           JSR     PC,RPOKE
2514  010722  105777  170764                           TSTB    @RXCSR  ;CHECK RXDONE
2515  010726  100401                                   BMI     .+4
2516  010730  104004                                   ERROR   4               ;RXDONE SHOULD BE ASSERTED
2517  010732  032777  004000  170752                    BIT     #RECACT,@RXCSR
```

```
2518  010740  001001                        BNE     .+4
2519  010742  104004                        ERROR   4                       ;RECACT SHOULD STILL BE ASSERTED
2520  010744  042777  000020  170740        BIC     #SYNSCH,@RXCSR  ;CLR SYNC SEARCH
2521  010752  032777  004000  170732        BIT     #RECACT,@RXCSR  ;IT SHOULD DROP IMMEDIATELY
2522  010760  001401                        BEQ     .+4
2523  010762  104004                        ERROR   4                       ;RECACT SHOULD BE CLR
2524  010764  105777  170722                TSTB    @RXCSR  ;RXDONE
2525  010770  100401                        BMI     .+4
2526  010772  104004                        ERROR   4                       ;RXDONE SHOULD STILL BE ASSERTED
2527  010774  012700  000021                MOV     #21,R0  ;EXPECTED DATA
2528  011000  017701  170712                MOV     @RXDBUF,R1      ;ACTUAL DATA
2529  011004  020001                        CMP     R0,R1   ;COMPARE EXP VS ACT
2530  011006  001401                        BEQ     .+4
2531  011010  104002                        ERROR   2                       ;DATA CHARS SHOULD COMPARE
2532  011012  105777  170674                TSTB    @RXCSR  ;CHECK RXDONE
2533  011016  100001                        BPL     .+4
2534  011020  104004                        ERROR   4               ;RXDONE SHOULD BE CLR FROM
2535                                        ;PREVIOUS READING OF RXDBUF
2536
2537                                ;;VERIFY THE MATCH DETECT & DATA RDY FLAGS BY PUMPING
2538                                ;;IN TWO # SYNC CHARS THRU MAINT DATA BIT
2539                                ;;WATCH THE RECACT BIT
2540                                ;;ON THE THIRD # CHARACTER IT SHOULD SET RXDONE
2541                                ;;*: DEPENDENT ON MONITOR.............
2542                                ;;IF ONE SYNC STRAP IS SELECTED THEN IT WILL ONLY
2543                                ;;TAKE ONE SYNC CHARACTER FOR RXDONE TO ASSERT
2544                                ;;ON THE SECOND CHARACTER
2545                                ;;ALSO CHECK THIS CHARACTER IN RXDBUF
2546                                ;;AND CHECK OPERATION OF SYNSCH
2547                                ;;MODE: SYNC INTERNAL
2548                                ;;LENGTH:SEVEN
2549                                ;;
2550                        ;;**************************************************************
2551  011022  000004                TST56:  SCOPE
2552  011024  052777  000400  170674        BIS     #MRESET,@TXCSR  ;MASTER RESET
2553  011032  012777  030000  170662        MOV     #SYNINT,@PARCSR ;SET THE MODE
2554  011040  052777  000400  170660        BIS     #MRESET,@TXCSR  ;MASTER RESET
2555
2556                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2557  011046  012777  064001  170652        MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2558
2559                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2560  011054  012777  034026  170640        MOV     #SYNINT!SEVEN!NOPAR!26,@PARCSR
2561  011062  016703  170630                MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2562  011066  052777  000020  170616        BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2563                                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2564  011074  042777  020000  170624        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2565  011102  052777  020000  170616        BIS     #CLK,@TXCSR     ;POKE CLK UP
2566                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2567  011110  042777  020000  170610        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2568  011116  052777  020000  170602        BIS     #CLK,@TXCSR     ;POKE CLK UP
2569  011124  012767  000002  167772        MOV     #2,COUNT
2570  011132  012767  000007  167762  1$:   MOV     #7,SHIFT        ;# OF SHIFTS
2571  011140  012767  000026  170332        MOV     #26,$TMP1       ;SYNC CHARACTER
2572  011146  004767  005724                JSR     PC,RPOKE
2573  011152  005367  167746                DEC     COUNT
```

```
2574  011156  001403                            BEQ     2$
2575                                            ;TEST SYNCNO TO SEE HOW MANY SYNC CHARS WERE SELECTED
2576  011160  105767  167762                    TSTB    SYNCNO
2577  011164  100762                            BMI     1$              ;TWO SYNC CHARS
2578  011166  105777  170520          2$:       TSTB    @RXCSR  ;CHECK REC DONE BIT
2579  011172  100001                            BPL     .+4
2580  011174  104004                            ERROR   4                       ;RXDONE SHOULD NOT BE ASSERTED
2581  011176  032777  004000  170506            BIT     #RECACT,@RXCSR
2582  011204  001001                            BNE     .+4
2583  011206  104004                            ERROR   4                       ;RECACT SHOULD BE ASSERTED
2584  011210  012767  000007  167704            MOV     #7,SHIFT
2585  011216  012767  000021  170254            MOV     #21,STMP1       ;ANY CHARACTER
2586  011224  004767  005646                    JSR     PC,RPOKE
2587  011230  105777  170456                    TSTB    @RXCSR  ;CHECK RXDONE
2588  011234  100401                            BMI     .+4
2589  011236  104004                            ERROR   4                       ;RXDONE SHOULD BE ASSERTED
2590  011240  032777  004000  170444            BIT     #RECACT,@RXCSR
2591  011246  001001                            BNE     .+4
2592  011250  104004                            ERROR   4                       ;RECACT SHOULD STILL BE ASSERTED
2593  011252  042777  000020  170432            BIC     #SYNSCH,@RXCSR  ;CLR SYNC SEARCH
2594  011260  032777  004000  170424            BIT     #RECACT,@RXCSR  ;IT SHOULD DROP IMMEDIATELY
2595  011266  001401                            BEQ     .+4
2596  011270  104004                            ERROR   4                       ;RECACT SHOULD BE CLR
2597  011272  105777  170414                    TSTB    @RXCSR  ;RXDONE
2598  011276  100401                            BMI     .+4
2599  011300  104004                            ERROR   4                       ;RXDONE SHOULD STILL BE ASSERTED
2600  011302  012700  000021                    MOV     #21,R0  ;EXPECTED DATA
2601  011306  017701  170404                    MOV     @RXDBUF,R1      ;ACTUAL DATA
2602  011312  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
2603  011314  001401                            BEQ     .+4
2604  011316  104002                            ERROR   2               ;DATA CHARS SHOULD COMPARE
2605  011320  105777  170366                    TSTB    @RXCSR  ;CHECK RXDONE
2606  011324  100001                            BPL     .+4
2607  011326  104004                            ERROR   4               ;RXDONE SHOULD BE CLR FROM
2608                                                                  ;PREVIOUS READING OF RXDBUF
2609
2610                                            ;;VERIFY THE MATCH DETECT & DATA RDY FLAGS BY PUMPING
2611                                            ;;IN TWO # SYNC CHARS THRU MAINT DATA BIT
2612                                            ;;WATCH THE RECACT BIT
2613                                            ;;ON THE THIRD # CHARACTER IT SHOULD SET RXDONE
2614                                            ;;#: DEPENDENT ON MONITOR.............
2615                                            ;;IF ONE SYNC STRAP IS SELECTED THEN IT WILL ONLY
2616                                            ;;TAKE ONE SYNC CHARACTER FOR RXDONE TO ASSERT
2617                                            ;;ON THE SECOND CHARACTER
2618                                            ;;ALSO CHECK THIS CHARACTER IN RXDBUF
2619                                            ;;AND CHECK OPERATION OF SYNSCH
2620                                            ;;MODE: SYNC INTERNAL
2621                                            ;;LENGTH:EIGHT
2622                                            ;;
2623                                    ;;=====================================================================
2624  011330  000004                    TST57:  SCOPE
2625  011332  052777  000400  170366            BIS     #MRESET,@TXCSR  ;MASTER RESET
2626  011340  012777  030000  170354            MOV     #SYNINT,@PARCSR ;SET THE MODE
2627  011346  052777  000400  170352            BIS     #MRESET,@TXCSR  ;MASTER RESET
2628
2629                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
```

```
DZDUQ-A MACY11 27(1006)  03-FEB-77  07:44  PAGE 55
DZDUQA.M11    27-JAN-77 09:46          INITIALIZE THE COMMON TAGS

2630  011354  012777  064001  170344          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2631
2632                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2633  011362  012777  036026  170332          MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
2634  011370  016703  170322                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2635  011374  052777  000020  170310          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2636                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2637  011402  042777  020000  170316          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2638  011410  052777  020000  170310          BIS     #CLK,@TXCSR     ;POKE CLK UP
2639                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2640  011416  042777  020000  170302          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2641  011424  052777  020000  170274          BIS     #CLK,@TXCSR     ;POKE CLK UP
2642  011432  012767  000002  167464          MOV     #2,COUNT
2643  011440  012767  000010  167454  1$:     MOV     #8.,SHIFT       ;# OF SHIFTS
2644  011446  012767  000026  170024          MOV     #26,$TMP1       ;SYNC CHARACTER
2645  011454  004767  005416                  JSR     PC,RPOKE
2646  011460  005367  167440                  DEC     COUNT
2647  011464  001403                          BEQ     2$
2648                                          ;TEST SYNCNO TO SEE HOW MANY SYNC CHARS WERE SELECTED
2649  011466  105767  167454                  TSTB    SYNCNO
2650  011472  100762                          BMI     1$      ;TWO SYNC CHARS
2651  011474  105777  170212          2$:     TSTB    @RXCSR  ;CHECK REC DONE BIT
2652  011500  100001                          BPL     .+4
2653  011502  104004                          ERROR   4               ;RXDONE SHOULD NOT BE ASSERTED
2654  011504  032777  004000  170200          BIT     #RECACT,@RXCSR
2655  011512  001001                          BNE     .+4
2656  011514  104004                          ERROR   4               ;RECACT SHOULD BE ASSERTED
2657  011516  012767  000010  167376          MOV     #8.,SHIFT
2658  011524  012767  000021  167746          MOV     #21,$TMP1       ;ANY CHARACTER
2659  011532  004767  005340                  JSR     PC,RPOKE
2660  011536  105777  170150                  TSTB    @RXCSR  ;CHECK RXDONE
2661  011542  100401                          BMI     .+4
2662  011544  104004                          ERROR   4               ;RXDONE SHOULD BE ASSERTED
2663  011546  032777  004000  170136          BIT     #RECACT,@RXCSR
2664  011554  001001                          BNE     .+4
2665  011556  104004                          ERROR   4               ;RECACT SHOULD STILL BE ASSERTED
2666  011560  042777  000020  170124          BIC     #SYNSCH,@RXCSR  ;CLR SYNC SEARCH
2667  011566  032777  004000  170116          BIT     #RECACT,@RXCSR  ;IT SHOULD DROP IMMEDIATELY
2668  011574  001401                          BEQ     .+4
2669  011576  104004                          ERROR   4               ;RECACT SHOULD BE CLR
2670  011600  105777  170106                  TSTB    @RXCSR  ;RXDONE
2671  011604  100401                          BMI     .+4
2672  011606  104004                          ERROR   4               ;RXDONE SHOULD STILL BE ASSERTED
2673  011610  012700  000021                  MOV     #21,R0  ;EXPECTED DATA
2674  011614  017701  170076                  MOV     @RXDBUF,R1      ;ACTUAL DATA
2675  011620  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
2676  011622  001401                          BEQ     .+4
2677  011624  104002                          ERROR   2               ;DATA CHARS SHOULD COMPARE
2678  011626  105777  170060                  TSTB    @RXCSR  ;CHECK RXDONE
2679  011632  100001                          BPL     .+4
2680  011634  104004                          ERROR   4               ;RXDONE SHOULD BE CLR FROM
2681                                          ;PREVIOUS READING OF RXDBUF
2682
2683                                  ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2684                                  ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2685                                  ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
```

```
                                        INITIALIZE THE COMMON TAGS

                                                ;;(OVRRUN,RXERR)
                                                ;;MODE:ISYMOD
                                                ;;LENGTH:FIVE
                                                ;;CHAR:25
                                                ;;
                                        ;;*****************************************************************
011636  000004                        TST60:  SCOPE
011640  052777  000400  170060                BIS     #MRESET,@TXCSR  ;MASTER RESET
011646  012777  000000  170046                MOV     #ISYMOD,@PARCSR ;SET THE MODE
011654  052777  000400  170044                BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
011662  012777  064001  170036                MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
011670  012777  000000  170024                MOV     #ISYMOD!FIVE!NOPAR!0,@PARCSR
011676  052777  000020  170006                BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
011704  042777  020000  170014                BIC     #CLK,@TXCSR     ;POKE CLK DOWN
011712  052777  020000  170006                BIS     #CLK,@TXCSR     ;POKE CLK UP

                                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
011720  042777  020000  170000                BIC     #CLK,@TXCSR     ;POKE CLK DOWN
011726  052777  020000  167772                BIS     #CLK,@TXCSR     ;POKE CLK UP
011734  016703  167756                        MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
011740  012700  000025                        MOV     #25,R0  ;EXPECTED
011744  012767  000007  167150                MOV     #7,$SHIFT       ;# OF SHIFTS
011752  012767  000152  167520                MOV     #152,$TMP1      ;DATA CHAR
011760  004767  005112                        JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
011764  105777  167722                        TSTB    @RXCSR  ;RXDONE ?
011770  100401                              BMI     .+4
011772  104004                              ERROR   4               ;RXDONE SHOULD BE SET
011774  017701  167716                      MOV     @RXDBUF,R1      ;ACTUAL
012000  020001                              CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
012002  001401                              BEQ     .+4
012004  104002                              ERROR   2               ;RECEIVED DATA DID NOT MATCH
                                                                    ;EXPECTED DATA - CHECK MAINT DATA
                                                                    ;OR RECEIVER LOGIC
012006  012767  000007  167106                MOV     #7,$SHIFT       ;# OF SHIFTS
012014  012767  000152  167456                MOV     #152,$TMP1      ;DATA CHAR
012022  004767  005050                        JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
                                        ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
012026  012767  000007  167066                MOV     #7,$SHIFT       ;# OF SHIFTS
012034  012767  000152  167436                MOV     #152,$TMP1      ;DATA CHAR
012042  004767  005030                        JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
012046  012700  140025                        MOV     #140000!25,R0   ;EXPECTED DATA PLUS
                                                                    ;RXERR & OVRRUN
012052  017701  167640                      MOV     @RXDBUF,R1      ;ACTUAL
012056  020001                              CMP     R0,R1   ;COMPARE EXP VS. ACT
012060  001401                              BEQ     .+4
012062  104002                              ERROR   2               ;SPECIFICALLY LOOK AT RXERR &
                                                                    ;OVRRUN BITS...THEY BOTH SHOULD BE SET

                                                ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
                                                ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
                                                ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
                                                ;;(OVRRUN,RXERR)
```

INITIALIZE THE COMMON TAGS

```
                                                ;;MODE:ISYMOD
                                                ;;LENGTH:FIVE
                                                ;;CHAR:12
                                                ;;
                                        ;;*****************************************************************
012064  000004                        TST61:  SCOPE
012066  052777  000400  167632                BIS     #MRESET,@TXCSR  ;MASTER RESET
012074  012777  000000  167620                MOV     #ISYMOD,@PARCSR ;SET THE MODE
012102  052777  000400  167616                BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
012110  012777  064001  167610                MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
012116  012777  000000  167576                MOV     #ISYMOD!FIVE!NOPAR!0,@PARCSR
012124  052777  000020  167560                BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
012132  042777  020000  167566                BIC     #CLK,@TXCSR     ;POKE CLK DOWN
012140  052777  020000  167560                BIS     #CLK,@TXCSR     ;POKE CLK UP
                                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
012146  042777  020000  167552                BIC     #CLK,@TXCSR     ;POKE CLK DOWN
012154  052777  020000  167544                BIS     #CLK,@TXCSR     ;POKE CLK UP
012162  016703  167530                       MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
012166  012700  000012                       MOV     #12,R0  ;EXPECTED
012172  012767  000007  166722               MOV     #7,SHIFT        ;# OF SHIFTS
012200  012767  000124  167272               MOV     #124,$TMP1      ;DATA CHAR
012206  004767  004664                       JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
012212  105777  167474                       TSTB    @RXCSR  ;RXDONE ?
012216  100401                               BMI     .+4
012220  104004                               ERROR   4               ;RXDONE SHOULD BE SET
012222  017701  167470                       MOV     @RXDBUF,R1      ;ACTUAL
012226  020001                               CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
012230  001401                               BEQ     .+4
012232  104002                               ERROR   2               ;RECEIVED DATA DID NOT MATCH
                                                ;EXPECTED DATA - CHECK MAINT DATA
                                                ;OR RECEIVER LOGIC
012234  012767  000007  166660               MOV     #7,SHIFT        ;# OF SHIFTS
012242  012767  000124  167230               MOV     #124,$TMP1      ;DATA CHAR
012250  004767  004622                       JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
                                        ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
012254  012767  000007  166640               MOV     #7,SHIFT        ;# OF SHIFTS
012262  012767  000124  167210               MOV     #124,$TMP1      ;DATA CHAR
012270  004767  004602                       JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
012274  012700  140012                       MOV     #140000!12,R0   ;EXPECTED DATA PLUS
                                                ;RXERR & OVRRUN
012300  017701  167412                       MOV     @RXDBUF,R1      ;ACTUAL
012304  020001                               CMP     R0,R1   ;COMPARE EXP VS. ACT
012306  001401                               BEQ     .+4
012310  104002                               ERROR   2               ;SPECIFICALLY LOOK AT RXERR &
                                                ;OVRRUN BITS...THEY BOTH SHOULD BE SET


                                                ;END OF PASS
                                                ;TYPE NAME OF TEST
                                                ;UPDATE PASS COUNT
```

```
                                                ;CHECK FOR EXIT TO ACT-11
                                                ;RESTART TEST
                                .EOP:   SCOPE
012312  000004
012314  004767  000344                  JSR     PC,CKSWR
012320  104401                          TYPE                            ;TYPE NAME OF TEST
012322  015454                          MEPASS
012324  104413  012556                  CONVRT  ,OUTCRY
012330  104401  015273                  TYPE    ,DEVICE
012334  105767  166612                  TSTB    MULTD   ;ARE YOU RUNNING MULTIPLE DEVICES ?
012340  001511                          BEQ     CCC     ;NO,JUMP AROUND
012342  005767  166620                  TST     ACTREG  ;ARE ANY DEVICES ACTIVE ?
012346  001007                          BNE     RUNIT   ;YES
012350  104401  015305                  TYPE    ,MCOW   ;NO
012354  016700  166606                  MOV     ACTREG,R0               ;DISPLAY ACTREG
012360  000000                          HALT                    ;SELECT SOMETHING TO RUN @ ACTREG:
                                        ;SELECT SWITCHES & HIT CONTINUE (PUT SWDO =1)
012362  000167  167564                  JMP     .START  ;START OVER AGAIN.....YOU DESELECTED EVERYTHING
012366  062767  000010  166560  RUNIT:  ADD     #10,BASEADD     ;NEXT BLOCK (ADDRESSES)
012374  062767  000010  166560  ZERO:   ADD     #10,BASEIV      ;NEXT BLOCK (VECTORS)
012402  000241                          CLC
012404  006167  166560                  ROL     ROTADD  ;UP DATE ROTATING POINTER
012410  103410                          BCS     2$      ;IS IT THE LAST DEVICE
                                                        ;TO BE TESTED IN THIS PASS ?
012412  036767  166552  166546          BIT     ROTADD,ACTREG   ;TEST THIS DEVICE FOR ACTIVE STATUS
012420  001762                          BEQ     RUNIT   ;IF NOT ACTIVE, TRY NEXT ADDRESS
012422  004767  000034                  JSR     PC,REPLAY       ;CALCULATE NEW PARAMETERS
012426  000167  000210                  JMP     RESTRT  ;YES IT WAS ACTIVE,TEST THIS DEVICE
012432  012767  000001  166530  2$:     MOV     #1,ROTADD               ;OK!,NOW SET UP  ROTATING
                                                        ;POINTER FOR NEXT MULTIPLE PASS
012440  016767  166512  166506          MOV     KEEPADD,BASEADD ;RESTORE BASE ADDRESS
012446  016767  166512  166506          MOV     KEEPIV,BASEIV   ;RESTORE BASE INTERRUPT VECTORS
012454  004767  000002                  JSR     PC,REPLAY       ;CALC NEW PARAMETERS
012460  000441                          BR      CCC     ;JUMP AROUND REPLAY
012462  016767  166466  004404  REPLAY: MOV     BASEADD,DUBASE  ;SET UP FOR NEW ADDRESSES
012470  004767  004246                  JSR     PC,DUADDR       ;CREATE NEW ADDRESSES
012474  016767  166462  167234          MOV     BASEIV,DURIV    ;CREATE DURIV
012502  062767  000002  166452          ADD     #2,BASEIV
012510  016767  166446  167222          MOV     BASEIV,DURIS    ;CREATE DURIS
012516  062767  000002  166436          ADD     #2,BASEIV
012524  016767  166432  167210          MOV     BASEIV,DUTIV    ;CREATE DUTIV
012532  062767  000002  166422          ADD     #2,BASEIV
012540  016767  166416  167176          MOV     BASEIV,DUTIS    ;CREATE DUTIS
012546  016767  167164  166406          MOV     DURIV,BASEIV    ;RESTORE
012554  000207                          RTS     PC

012556  000001                  OUTCRY: 1
012560      006         002             .BYTE   6,2
012562  001712                          RXCSR

012564                          CCC:
012564  005067  166612                  CLR     $TSTNM                  ;CLEAR TEST NUMBER
012570  005067  166622                  CLR     $ERRPC                  ;CLEAR LAST ERROR PC
012574  005067  166603                  CLR     $ERFLG                  ;CLEAR ERROR FLAG
012600  005267  166306                  INC     PASCNT                  ;UPDATE PASS COUNT
012604  016767  166302  166270          MOV     PASCNT,LIGHTS           ;DISPLAY PASS COUNT
```

```
012612  016767  166274  166714          MOV     PASCNT,$PASS            ;PASS COUNT TO APT
012620  013701  000042                  MOV     @#42,R1                 ;CHECK FOR ACT-11 OR DDP
012624  001406                          BEQ     RESTRT          ;IF NO CONTINUE TESTING
012626  000005                          RESET
012630  000005                          RESET
012632  004711                  $ENDAD: JSR     PC,(R1)
012634  000240                          NOP
012636  000240                          NOP
012640  000240                          NOP
012642  106427  000340          RESTRT: MTPS    #340    ;PREVENT INTERRUPTS (PRIO: 7)
012646  004767  000012                  JSR     PC,CKSWR
012652  012767  003364  166526          MOV     #TST1+2,$LPADR          ;SET LAST ADDRESS POINTER
012660  000167  170476                  JMP     TST1

                                        ;CHECK SWITCH REGISTER ROUTINE.
                                        ;CHECKS TO ALLOW FOR <↑G> TO ALLOW
                                        ;THE CHANGING OF LOCATION 176

012664  005737  000042          CKSWR:  TST     @#42
012670  001040                          BNE     OUT
012672  022767  000176  166540          CMP     #SWREG,SWR      ;SOFTWARE SWR PRESENT?
012700  001034                          BNE     OUT             ;NO--LEAVE
012702  105777  166536                  TSTB    @$TKS           ;CHECK TTY READY
012706  100031                          BPL     OUT             ;NO--LEAVE
012710  017767  166532  000422          MOV     @$TKB,.MSG      ;GET CHARACTER
012716  042767  177600  000414          BIC     #177600,.MSG    ;STRIP JUNK
012724  122767  000007  000406          CMPB    #7,.MSG         ;IS IT <↑G> ?
012732  001017                          BNE     OUT             ;NO
012734  104401  016061                  TYPE    ,MCNTG
012740  005137  013000          CNTLU:  COM     @#RDSW
012744  104401  016071                  TYPE    ,MMSWR
012750  104413                          CONVRT
012752  013002                          SWREGL
012754  104406  016102                  INSTR,MMNEW
012760  104410                          PARAM
012762  000000                          0
012764  177777                          177777
012766  000176                          SWREG
012770     000     001          .BYTE   0,1
012772  005037  013000          OUT:    CLR     @#RDSW
012776  000207                          RTS     PC
013000  000000                  RDSW:   .WORD   0
013002  000001                  SWREGL: 1
013004     006     002                  .BYTE   6,2
013006  000176                          SWREG

013010  000005                          5

                                        ;CHECK FOR FREEZE ON CURRENT DATA

013012  004767  177646          .SCOP1: JSR     PC,CKSWR
013016  032777  001000  166414          BIT     #SW09,@SWR
013024  001402                          BEQ     1$
013026  016716  166056                  MOV     LOCK,(SP)
013032  000002                  1$:     RTI
                                .SBTTL  TYPE ROUTINE
```

```
2910
2911                                    ;;*****************************************************************
2912                                    ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
2913                                    ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
2914                                    ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
2915                                    ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
2916                                    ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
2917                                    ;*
2918                                    ;*CALL:
2919                                    ;*1) USING A TRAP INSTRUCTION
2920                                    ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2921                                    ;*OR
2922                                    ;*      TYPE
2923                                    ;*      MESADR
2924                                    ;*
2925
2926 013034  105767  166417            $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
2927 013040  100002                            BPL     1$              ;;BR IF YES
2928 013042  000000                            HALT                    ;;HALT HERE IF NO TERMINAL
2929 013044  000430                            BR      3$              ;;LEAVE
2930 013046  010046                    1$:     MOV     R0,-(SP)        ;;SAVE R0
2931 013050  017600  000002                    MOV     2(SP),R0        ;;GET ADDRESS OF ASCIZ STRING
2932 013054  122767  000001  166464            CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
2933 013062  001011                            BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
2934 013064  132767  000100  166455            BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
2935 013072  001405                            BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
2936 013074  010067  000004                    MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
2937 013100  004767  000006'                   JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
2938 013104  000000                    61$:    .WORD   0               ;;MESSAGE ADDRESS
2939 013106  132767  000040  166433    62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
2940 013114  001003                            BNE     60$             ;;YES,SKIP TYPE OUT
2941 013116  112046                    2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
2942 013120  001005                            BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
2943 013122  005726                            TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
2944 013124  012600                    60$:    MOV     (SP)+,R0        ;;RESTORE R0
2945 013126  062716  000002            3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
2946 013132  000002                            RTI                     ;;RETURN
2947 013134  122716  000011            4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
2948 013140  001430                            BEQ     8$
2949 013142  122716  000200                    CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
2950 013146  001006                            BNE     5$
2951 013150  005726                            TST     (SP)+           ;;POP  <CR><LF> EQUIV
2952 013152  104401                            TYPE                    ;;TYPE A CR AND LF
2953 013154  001523                            $CRLF
2954 013156  105067  000130                    CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
2955 013162  000755                            BR      2$              ;;GET NEXT CHARACTER
2956 013164  004767  000056            5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
2957 013170  126726  166262            6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
2958 013174  001350                            BNE     2$              ;;IF NO GO GET NEXT CHAR.
2959 013176  016746  166252                    MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
2960                                                                    ;;AND THE NULL CHAR.
2961 013202  105366  000001            7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
2962 013206  002770                            BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
2963 013210  004767  000032                    JSR     PC,$TYPEC       ;;GO TYPE A NULL
2964 013214  105367  000072                    DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
2965 013220  000770                            BR      7$              ;;LOOP
```

```
                                        ;HORIZONTAL TAB PROCESSOR
                                        
013222  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
013226  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
013232  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
013240  001372                          BNE     9$              ;;TAB STOP
013242  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
013244  000724                          BR      2$              ;;GET NEXT CHARACTER
013246  105777  166176          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
013252  100375                          BPL     $TYPEC
013254  116677  000002  166170          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
013262  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
013270  001003                          BNE     1$              ;;BRANCH IF NO
013272  105067  000014                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
013276  000406                          BR      $TYPEX          ;;EXIT
013300  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
013306  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
013310  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
013312  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
013314  000207          $TYPEX: RTS     PC
                                        
                                        ;ASCII STRING INPUT ROUTINE
                                        
013316  017667  000000  000014  .INSTR: MOV     @(SP)+,.MSG     ;PICK UP MESSAGE
013324  062716  000002                  ADD     #2,(SP)         ;JUMP AROUND MESSAGE FOR RTI
013330  105767  166212                  TSTB    $ENV            ;APT CONTROL
013334  001036                          BNE     INSTR2          ;YES NO TYPE
013336  104401                  .INST1: TYPE
013340  000000                  .MSG:   0
013342  012704  016114                  MOV     #INBUF,R4       ;GET STARTING LOC OF INBUF
013346  012703  000007                  MOV     #7,R3           ;MAX # OF CHARS
013352  105777  166066          1$:     TSTB    @$TKS   ;TTY FLAG
013356  100375                          BPL     1$
013360  117714  166062                  MOVB    @$TKB,(R4)      ;TAKE CHAR
013364  142714  000200                  BICB    #200,(R4)       ;STRIP
013370  121427  000025                  CMPB    (R4),#25        ;IS IT <^G>
013374  001760                          BEQ     .INST1
013376  122427  000015                  CMPB    (R4)+,#15       ;CHECK FOR CR
013402  001413                          BEQ     INSTR2
013404  105777  166040          2$:     TSTB    @$TPS   ;TEST FLAG
013410  100375                          BPL     2$
013412  117777  166030  166032          MOVB    @$TKB,@$TPB     ;ECHO CHARACTER
013420  005303                          DEC     R3      ;DID YOU TYPE TOO MANY CHARS ?
013422  001353                          BNE     1$
013424  104401                  .INSTE: TYPE
013426  015401                          MOM     ;?
013430  000742                          BR      .INST1  ;RETRY
013432  000002                  INSTR2: RTI
                                        
                                        ;CONVERT ASCII STRING TO OCTAL
                                        
013434  011605                  .PARAM: MOV     (SP),R5 ;PUT CONTENTS OF SP INTO R5
013436  012567  000162                  MOV     (R5)+,LOLIM     ;PUT LOW LIMIT INTO LOLIM
013442  012567  000160                  MOV     (R5)+,HILIM     ;PUT HIGH LIMIT INTO HILIM
```

```
3022  013446  012567  000156                  MOV     (R5)+,DEVADR    ;PUT STORE LOC INTO DEVADR
3023  013452  112567  000154                  MOVB    (R5)+,LOBITS    ;PUT MASK INTO LOBITS
3024  013456  112567  000151                  MOVB    (R5)+,ADRCNT    ;PUT COUNT INTO ADRCNT
3025  013462  010516                          MOV     R5,(SP) ;RESTORE RETURN ADDR ON STACK FOR RTI
3026  013464  005005                  PARAM1: CLR     R5
3027  013466  012704  016114                  MOV     #INBUF,R4
3028  013472  122714  000015                  CMPB    #15,(R4)        ;CR ?
3029  013476  001420                          BEQ     PARERR  ;YOU TYPED CR TOO SOON !
3030  013500  121427  000060          1$:     CMPB    (R4),#60        ;LOW LIMIT ASCII 0
3031  013504  002415                          BLT     PARERR
3032  013506  121427  000067                  CMPB    (R4),#67        ;HIGH LIMIT ASCII 7
3033  013512  003012                          BGT     PARERR
3034  013514  142714  000060                  BICB    #60,(R4)        ;CONVERT TO OCTAL
3035  013520  152405                          BISB    (R4)+,R5        ;STORE AWAY ITS AN OK CHAR
3036  013522  122714  000015                  CMPB    #15,(R4)        ;CR ?
3037  013526  001414                          BEQ     LIMITS  ;NOW CHECK FOR HIGH &LOW LIMIT CONDS
3038  013530  006305                          ASL     R5      ;ALLOCATE ROOM FOR NEXT CHAR
3039  013532  006305                          ASL     R5
3040  013534  006305                          ASL     R5
3041  013536  000760                          BR      1$
3042  013540  122714  000015          PARERR: CMPB    #15,(R4)        ;CR?
3043  013544  001003                          BNE     120$
3044  013546  005737  013000                  TST     @#RDSW          ;CK SWR USED
3045  013552  001023                          BNE     PARTI
3046  013554  104407                  120$:   INSTER  ;RETRY
3047  013556  000742                          BR      PARAM1
3048
3049                                          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
3050
3051  013560  020567  000042          LIMITS: CMP     R5,HILIM
3052  013564  101365                          BHI     PARERR  ;THE # IS TOO HIGH
3053  013566  020567  000032                  CMP     R5,LOLIM
3054  013572  103762                          BLO     PARERR  ;THE # IS TOO LOW
3055  013574  136705  000032                  BITB    LOBITS,R5       ;TEST BY MASKINGTHE #
3056  013600  001357                          BNE     PARERR
3057
3058                                          ;STORE NUMBER AT SPECIFIED ADDRESS
3059
3060  013602  016704  000022                  MOV     DEVADR,R4       ;GET STARTING ADDR OF
3061  013606  010524                  1$:     MOV     R5,(R4)+        ;STORE AT THIS ADDR
3062  013610  062705  000002                  ADD     #2,R5
3063  013614  105367  000013                  DECB    ADRCNT  ;HOW MANY TIMES + 2 ?
3064  013620  001372                          BNE     1$
3065  013622  000002                  PARTI:  RTI
3066  013624  000000                  LOLIM:  0
3067  013626  000000                  HILIM:  0
3068  013630  000000                  DEVADR: 0
3069  013632  000000                  LOBITS: 0
3070          013633                  ADRCNT=LOBITS+1
3071
3072                                          ;SAVE PC OF TEST THAT FAILED AND R0-R5
3073
3074  013634  016667  000004  165264  .SAV05: MOV     4(SP),SAVPC
3075
3076                                          ;SAVE R0-R5
3077
```

```
                                        TYPE ROUTINE

 3078  013642  010567  165626   SVO5:   MOV     R5,$REG5
 3079  013646  010467  165620           MOV     R4,$REG4
 3080  013652  010367  165612           MOV     R3,$REG3
 3081  013656  010267  165604           MOV     R2,$REG2
 3082  013662  010167  165576           MOV     R1,$REG1
 3083  013666  010067  165570           MOV     R0,$REG0
 3084  013672  000002                   RTI
 3085
 3086                                   ;RESTORE R0-R5
 3087
 3088  013674  016700  165562   .RES05: MOV     $REG0,R0
 3089  013700  016701  165560           MOV     $REG1,R1
 3090  013704  016702  165556           MOV     $REG2,R2
 3091  013710  016703  165554           MOV     $REG3,R3
 3092  013714  016704  165552           MOV     $REG4,R4
 3093  013720  016705  165550           MOV     $REG5,R5
 3094  013724  000002                   RTI
 3095
 3096                                   ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
 3097
 3098  013726  104401           .CONVR: TYPE
 3099  013730  015405                   MCRLF   ;CR  LF
 3100  013732  017601  000000           MOV     @(SP),R1        ;PICK UP DATA POINTER
 3101  013736  062716  000002           ADD     #2,(SP) ;SET UP SP FOR RTI
 3102  013742  012167  000130           MOV     (R1)+,WRDCNT    ;PICK UP # OF WORDS FROM TABLE
 3103  013746  112167  000126   1$:     MOVB    (R1)+,CHRCNT    ;PICK UP # OF CHARS FROM TABLE
 3104  013752  112167  000123           MOVB    (R1)+,SPACNT    ;PICK UP # OF SPACES FROM TABLE
 3105  013756  013167  000120           MOV     @(R1)+,BINWRD   ;PICK UP ADDRESS OF  MSG
 3106                                                           ;FROM TABLE
 3107  013762  016704  000114   2$:     MOV     BINWRD,R4       ;SAVE
 3108  013766  116705  000106           MOVB    CHRCNT,R5       ;SAVE
 3109  013772  012700  016156           MOV     #TEMP,R0        ;STARTING ADDRESS OF TEMP BLOCK
 3110  013776  010403           3$:     MOV     R4,R3           ;SAVE
 3111  014000  042703  177770           BIC     #177770,R3      ;CLR OUT UPPER BITS .. SAVE CHAR
 3112  014004  062703  000260           ADD     #260,R3 ;CONVERT TO ASCII
 3113  014010  110320                   MOVB    R3,(R0)+        ;STORE AWAY
 3114  014012  006204                   ASR     R4      ;SHIFT FOR NEXT #
 3115  014014  006204                   ASR     R4      ;DITTO
 3116  014016  006204                   ASR     R4      ;DITTO
 3117  014020  005305                   DEC     R5      ;DEC CHAR COUNT
 3118  014022  001365                   BNE     3$      ;DO IT AGAIN ?
 3119  014024  012703  016220           MOV     #MDATA,R3       ;STARTING ADDRESS OF MDATA BLOCK
 3120  014030  114023           4$:     MOVB    -(R0),(R3)+     ;REVERSE THE ORDER OF NUMBERS
 3121  014032  105367  000042           DECB    CHRCNT  ;DEC CHAR COUNT
 3122  014036  001374                   BNE     4$      ;DO IT AGAIN ?
 3123  014040  105767  000035           TSTB    SPACNT  ;HOW MANY SPACES ?
 3124  014044  001405                   BEQ     6$      ;TYPE # IF BR =0
 3125  014046  112723  000240   5$:     MOVB    #240,(R3)+      ;"SPACE" IN ASCII
 3126  014052  105367  000023           DECB    SPACNT  ;DEC # OF SPACE COUNT
 3127  014056  001373                   BNE     5$      ;DO IT AGAIN ?
 3128  014060  105013           6$:     CLRB    (R3)    ;INSERT "0" FOR TTY OUTPUT ROUTINE
 3129  014062  104401                   TYPE
 3130  014064  016220                   MDATA   ;THIS MESSAGE
 3131  014066  005367  000004           DEC     WRDCNT  ;HOW MANY #'S ?
 3132  014072  001325                   BNE     1$      ;DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
 3133  014074  000002                   RTI     ;RETURN TO PROGRAM
```

```
                                TYPE ROUTINE

014076  000000                  WRDCNT: 0
014100  000000                  CHRCNT: 0
        014101                  SPACNT=CHRCNT+1
014102  000000                  BINWRD: 0

                                        ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
                                        ;BUFFER TO THE CHARACTERS "N" AND "Y".
                                        ;IF THE CHARACTER IS "N" CLEAR THE FLAG
                                        ;IF THE CHARACTER IS "Y" SET THE FLAG

014104  017605  000000        .SETFLG:MOV     @(SP),R5
014110  122767  000116  001776        CMPB    #'N,INBUF       ;IS IT "N" ?
014116  001002                        BNE     1$
014120  105015                        CLRB    (R5)    ;000
014122  000406                        BR      2$
014124  122767  000131  001762  1$:   CMPB    #'Y,INBUF       ;IS IT "Y" ?
014132  001005                        BNE     3$
014134  112715  177777              MOVB    #-1,(R5)        ;377
014140  062716  000002        2$:     ADD     #2,(SP)
014144  000002                        RTI
014146  104407                3$:     INSTER  ;RETRY
014150  000755                        BR      .SETFLG
                                .SBTTL  ERROR HANDLER ROUTINE

                                ;;**********************************************************************
                                ;#THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;#SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;#AND GO TO SAVIT ON ERROR
                                ;#THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;#SW15=1         HALT ON ERROR
                                ;#SW13=1         INHIBIT ERROR TYPEOUTS
                                ;#SW10=1         BELL ON ERROR
                                ;#SW09=1         LOOP ON ERROR
                                ;#CALL
                                ;#      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

014152                          $ERROR:
014152  105267  165225        7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
014156  001775                        BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
014160  016777  165216  165254        MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
014166  032777  002000  165244        BIT     #BIT10,@SWR     ;;BELL ON ERROR?
014174  001402                        BEQ     1$              ;;NO - SKIP
014176  104401  001516                TYPE    ,$BELL          ;;RING BELL
014202  005267  165204        1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
014206  011667  165204              MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
014212  162767  000002  165176        SUB     #2,$ERRPC
014220  117767  165172  165166        MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
014226  032777  020000  165204        BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
014234  001004                        BNE     20$             ;;SKIP TYPEOUTS
014236  004767  000072                JSR     PC,SAVIT        ;;GO TO USER ERROR ROUTINE
014242  104401  001523                TYPE    ,$CRLF
014246                          20$:
014246  122767  000001  165272        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
014254  001007                        BNE     2$              ;;NO,SKIP APT ERROR REPORT
014256  116767  165132  000004        MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
014264  004767  000016'              JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
```

```
014270  000000                  21$:    .BYTE   0
014271  000000                          .BYTE   0
014272  000777                  22$:    BR      22$             ;;APT ERROR LOOP,
014274  005777  165140          2$:     TST     @SWR            ;;HALT ON ERROR
014300  100001                          BPL     3$              ;;SKIP IF CONTINUE
014302  000000                          HALT                    ;;HALT ON ERROR!
014304  032777  001000  165126  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
014312  001402                          BEQ     4$              ;;BR IF NO
014314  016716  165070                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
014320  005767  165170          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
014324  001402                          BEQ     5$              ;;BR IF NONE
014326  016716  165162                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
014332                          5$:
014332  000002                          RTI                     ;;RETURN
014334  010067  164570          SAVIT:  MOV     R0,HLD0
014340  010167  164566                  MOV     R1,HLD1
014344  010267  164564                  MOV     R2,HLD2
014350  010367  164562                  MOV     R3,HLD3
014354  010467  164560                  MOV     R4,HLD4
014360  010567  164556                  MOV     R5,HLD5
014364  016767  165012  164552          MOV     $TSTNM,HLD6

                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

                                ;;**************************************************************
                                ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" (SITEMB) TO DETERMINE WHICH
                                ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),
                                ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

014372                          SERRTYP:
014372  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
014376  010046                          MOV     R0,-(SP)        ;;SAVE R0
014400  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
014402  153700  001414                  BISB    @#SITEMB,R0
014406  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                ;;TYPE THE PC OF THE ERROR
014410  016746  165002                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                ;;ERROR ADDRESS
014414  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
014416  000426                          BR      6$              ;;GET OUT
014420  005300          1$:     DEC     R0                      ;;ADJUST THE INDEX SO THAT IT WILL
014422  006300                  ASL     R0                      ;;       WORK FOR THE ERROR TABLE
014424  006300                  ASL     R0
014426  006300                  ASL     R0
014430  062700  001652                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
014434  012067  000004                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
014440  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
014442  104401                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
014444  000000          2$:     .WORD   0                       ;;"ERROR MESSAGE" POINTER GOES HERE
014446  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
014452  012067  000004          3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
014456  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
014460  104401                          TYPE                    ;;TYPE THE "DATA HEADER"
014462  000000          4$:     .WORD   0                       ;;"DATA HEADER" POINTER GOES HERE
014464  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
014470  011000          5$:     MOV     (R0),R0                 ;;PICKUP "DATA TABLE" POINTER
```

```
                                        ERROR MESSAGE TYPEOUT ROUTINE

                                                BNE     7$                      ;;GO TYPE THE DATA
                                        6$:     MOV     (SP)+,R0                ;;RESTORE R0
                                                TYPE    ,$CRLF                  ;;"CARRIAGE RETURN" & "LINE FEED"
                                                RTS     PC                      ;;RETURN
                                        7$:
                                                MOV     @(R0)+,-(SP)            ;;SAVE @(R0)+ FOR TYPEOUT
                                                TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                                                TST     (R0)                    ;;IS THERE ANOTHER NUMBER?
                                                BEQ     6$                      ;;BR IF NO
                                                TYPE    ,8$                     ;;TYPE TWO(2) SPACES
                                                BR      7$                      ;;LOOP
                                        8$:     .ASCIZ  /  /                    ;;TWO(2) SPACES
                                                .EVEN
                                        .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

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

014526  017646  000000                  $TYPOS: MOV     @(SP),-(SP)             ;;PICKUP THE MODE
014532  116667  000001  000211                  MOVB    1(SP),$OFILL            ;;LOAD ZERO FILL SWITCH
014540  112667  000207                          MOVB    (SP)+,$OMODE+1          ;;NUMBER OF DIGITS TO TYPE
014544  062716  000002                          ADD     #2,(SP)                 ;;ADJUST RETURN ADDRESS
014550  000406                          $TYPON: BR      $TYPON
014552  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL               ;;SET THE ZERO FILL SWITCH
014560  112767  000006  000165          MOVB    #6,$OMODE+1             ;;SET FOR SIX(6) DIGITS
014566  112767  000005  000154  $TYPON: MOVB    #5,$OCNT                ;;SET THE ITERATION COUNT
014574  010346                          MOV     R3,-(SP)                ;;SAVE R3
014576  010446                          MOV     R4,-(SP)                ;;SAVE R4
014600  010546                          MOV     R5,-(SP)                ;;SAVE R5
014602  116704  000145                  MOVB    $OMODE+1,R4             ;;GET THE NUMBER OF DIGITS TO TYPE
014606  005404                          NEG     R4
014610  062704  000006                  ADD     #6,R4                   ;;SUBTRACT IT FOR MAX. ALLOWED
014614  110467  000132                  MOVB    R4,$OMODE               ;;SAVE IT FOR USE
014620  116704  000125                  MOVB    $OFILL,R4               ;;GET THE ZERO FILL SWITCH
014624  016605  000012                  MOV     12(SP),R5               ;;PICKUP THE INPUT NUMBER
014630  005003                          CLR     R3                      ;;CLEAR THE OUTPUT WORD
```

BINARY TO OCTAL (ASCII) AND TYPE

```
014632  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
014634  000404                          BR      3$              ;;GO DO MSB
014636  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
014640  006105                          ROL     R5
014642  006105                          ROL     R5
014644  010503                          MOV     R5,R3
014646  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
014650  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
014654  100016                          BPL     7$              ;;BR IF NO
014656  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
014662  001002                          BNE     4$              ;;TEST FOR 0
014664  005704                          TST     R4              ;;SUPPRESS THIS 0?
014666  001403                          BEQ     5$              ;;BR IF YES
014670  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
014672  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
014676  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
014702  110367  000040                  MOVB    R3,B$           ;;SAVE FOR TYPING
014706  104401  014746                  TYPE    B$              ;;GO TYPE THIS DIGIT
014712  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
014716  003347                          BGT     2$              ;;BR IF MORE TO DO
014720  002402                          BLT     6$              ;;BR IF DONE
014722  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
014724  000744                          BR      2$              ;;GO DO THE LAST DIGIT
014726  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
014730  012604                          MOV     (SP)+,R4        ;;RESTORE R4
014732  012603                          MOV     (SP)+,R3        ;;RESTORE R3
014734  016666  000002  000004         MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
014742  012616                          MOV     (SP)+,(SP)
014744  000002                          RTI                     ;;RETURN
014746     000          B$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
014747     000                  .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
014750     000          $OCNT:  .BYTE   0                       ;;OCTAL DIGIT COUNTER
014751     000          $OFILL: .BYTE   0                       ;;ZERO FILL SWITCH
014752  000000          $OMODE: .WORD   0                       ;;NUMBER OF DIGITS TO TYPE
                                ;ENTER HERE ON POWER FAILURE


014754                  $PWRON:
014754  010046          .PFAIL: MOV     R0,-(SP)                ;SAVE R0-R5 ON PROCESSOR STACK
014756  010146                  MOV     R1,-(SP)
014760  010246                  MOV     R2,-(SP)
014762  010346                  MOV     R3,-(SP)
014764  010446                  MOV     R4,-(SP)
014766  010546                  MOV     R5,-(SP)
014770  016746  163030          MOV     24,-(SP)
014774  010667  164116          MOV     SP,SAVSP                ;SAVE STACK POINTER
015000  012767  015012  163016  MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
015006  000000                  HALT                            ;HALT ON POWER DOWN NORMAL
015010  000777                  BR      .


                                ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED


015012  016706  164100  RESTAR: MOV     SAVSP,SP                ;RESTORE STACK POINTER
015016  012605                  MOV     (SP)+,R5                ;RESTORE R0-R5
015020  012604                  MOV     (SP)+,R4
015022  012603                  MOV     (SP)+,R3
```

```
                                                MOV     (SP)+,R2
                                                MOV     (SP)+,R1
                                                MOV     (SP)+,R0
015024  012702  014754  162764                  MOV     #.PFAIL,24              ;SET UP FOR POWER FAILURE
015030  106427  000340                          MTPS    #340
015034  012706  001100                          MOV     #STACK,SP
015040  005067  001102                          CLR     TEMP
015044  005267  001076                          INC     TEMP
015050  001375                                  BNE     .-4
015052  104413                                  CONVRT
015054  015106                                  PFTAB
015056  104401                                  TYPE
015060  015410                                  MPFAIL
015062  005067  164305                          CLR     SERFLG
015066  005067  164314                          CLR     SERRPC
015072  000177  163776                          JMP     @RETURN
015076  000001                  PFTAB:  1
015100      006      002                        .BYTE   6,2
015102  000207                                  RETURN
015104  005015  042012  053125  MTITLE: .ASCIZ  <15><12><12>/DUV11 DZDUQ-A TAPE A /<15><12>
015122  030461  042040  042132
015130  050525  040455  052040
015136  050101  020105  020101
015144  005015      000
015147      015  053012  041505  MVECTO: .ASCIZ  <15><12>/VEC ADD-/
015154  040440  042104  000055
015162  005015  051461  020124  MREGAD: .ASCIZ  <15><12>/1ST DEV: REC CSR ADD-/
015170  042504  035126  051040
015176  041505  041440  051123
015204  040440  042104  000055
015212  005015  052515  052114  MMULT:  .ASCIZ  <15><12>/MULT DEV ? (Y OR N)-/
015220  042040  053105  037440
015226  024040  020131  051117
015234  047040  026451      000
015241      015  046012  051501  MLASTD: .ASCIZ  <15><12>/LAST DEV: REC CSR ADDR-/
015246  020124  042504  035126
015254  051040  041505  041440
015262  051123  040440  042104
015270  026522      000
015273      075  042504  044526  DEVICE: .ASCIZ  /=DEVICE  /
015300  042503  020040      000
015305      015  051412  046105  MCOM:   .ASCIZ  <15><12>/SELECT TO RUN @ACTREG/
015312  041505  020124  047524
015320  051040  047125  040040
015326  041501  051124  043505
015334      000
015335      015  047412  043126  MRANGE: .ASCIZ  <15><12>/OVFLO:RETYPE LAST DEV RXCSR ADDS-/
015343  047514  051072  052105
015351  050131  020105  040514
015357  052123  042040  053105
015364  051040  041530  051123
015372  040440  042104  026523
015400      000
015401      040  037440      000  MQM:    .ASCIZ  /  ?/
015405      015  000012           MCRLF:  .ASCIZ  <15><12>
015410  043120  044501  026114  MPFAIL: .ASCIZ  /PFAIL,  RESTART AT TEST IN PROGRESS/
```

```
015416 020040 042522 052123
015424 051101 020124 052101
015432 052040 051505 020124
015440 047111 050040 047522
015446 051107 051505 000123
015454 005015 047105 020104  MEPASS: .ASCIZ  <15><12>/END OF PASS TAPE A/
015462 043117 050040 051501
015470 020123 040524 042520
015476 040440    000
015501    015 051012    000  MR:     .ASCIZ  <15><12>/R/
015505    015 052012 051505  MTSTPC: .ASCIZ  <15><12>/TEST PC-/
015512 020124 041520 000055
015520 005015 047514 045503  MLOCK:  .ASCIZ  <15><12>/LOCK ON  TEST? (Y OR N)-/
015526 047440 020116 052040
015534 051505 037524 024040
015542 020131 051117 047040
015550 026451    000
015553    015 021412 047440  MSYNC:  .ASCIZ  <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/
015560 020106 054523 041516
015566 041440 040510 051522
015574 051440 046105 041505
015602 042524 020104 020050
015610 020061 051117 031040
015616 026451    000
015621    015 044412 020123  MWIRE6: .ASCIZ  <15><12>/IS SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
015626 042523 020103 046530
015634 052111 051440 044527
015642 041524 020110 032505
015650 026465 020062 047111
015656 020077 054450 047440
015664 020122 024516 000055
015672 005015 051511 051440  MWIRE5: .ASCIZ  <15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
015700 041505 051040 041505
015706 051440 044527 041524
015714 020110 032505 026465
015722 020063 047111 020077
015730 054450 047440 020122
015736 024516 000055
015742 005015 051511 047440  MWIRE4: .ASCIZ  <15><12>/IS OPT CLR ENABLE SWITCH E55-1 IN? (Y OR N)-/
015750 052120 041440 051114
015756 042440 040516 046102
015764 020105 053523 052111
015772 044103 042440 032465
016000 030455 044440 037516
016006 024040 020131 051117
016014 047040 026451    000
016021    015 005012 031510  MEXTJ:  .ASCIZ  <15><12><12>/H315 CONNECTOR ON ?(Y OR N)-/
016026 032461 041440 047117
016034 042516 052103 051117
016042 047440 020116 024077
016050 020131 051117 047040
016056 026451    000
016061    015 020012 043536  MCNTG:  .ASCIZ  <15><12>/ ↑G  /
016066 020040    000
016071    040 053523 036522  MMSWR:  .ASCIZ  / SWR=   /
016076 020040 000040
```

```
                                        BINARY TO OCTAL (ASCII) AND TYPE

016102  020040  047040  053505  MMNEW:  .ASCIZ  /   NEW= /
016110  020075  000040
                                .EVEN

                                        ;BUFFERS FOR INPUT-OUTPUT

016114  000000                  INBUF:  0
        016156                  .=.+40
016156  000000                  TEMP:   0
        016220                  .=.+40
016220  000000                  MDATA:  0
        016262                  .=.+40
                                .SBTTL  SCOPE HANDLER ROUTINE

                                ;;*****************************************************************
                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW14=1         LOOP ON TEST
                                ;*SW11=1         INHIBIT ITERATIONS
                                ;*SW09=1         LOOP ON ERROR
                                ;*SW08=1         LOOP ON TEST IN SWR<7:0>
                                ;*CALL
                                ;*      SCOPE           ;;SCOPE=IOT

016262                          $SCOPE:

                                        ;SCOPE LOOP AND INTERATION HANDLER

016262                          .SCOPE:
016262  004767  174376          JSR     PC,CKSWR
016266  005067  163124          CLR     @ERRPC          ;CLEAR LAST ERROR PC
016272  022716  003364          CMP     #TST1+2,(SP)    ;IS SCOPE AT BEGINING OF TEST 1?
016276  001413                  BEQ     $XTSTR          ;YES NO LOOP.

016300  000406                  TTST:   BR      1$              ;GO TO 1$ (IF LOCK SW02=1)
016302  105777  163136          TSTB    @$TKS           ;KEYBOARD DONE?
016306  100123                  BPL     $OVER           ;BR IF NO
016310  017766  163132  177776  MOV     @$TKB,-2(SP)    ;CLEAR DONE BIT
016316  032777  040000  163114  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
016324  001114                  BNE     $OVER           ;;YES IF SW14=1
                                ;*****START OF CODE FOR THE XOR TESTER*****
016326  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
016330  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
016334  012737  016354  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
016342  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
016346  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
016352  000463                  BR      $SVLAD          ;;GO TO THE NEXT TEST
016354  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
016356  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
016362  000423                  BR      7$              ;;LOOP ON THE PRESENT TEST
016364                          6$:;*****END OF CODE FOR THE XOR TESTER*****
016364  032777  000400  163046          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
016372  001404                  BEQ     2$              ;;BR IF NO
```

```
016374 127767 163040 163000          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
016402 001465                        BEQ     $OVER           ;;BR IF YES
016404 105767 162773          2$:    TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
016410 001421                        BEQ     3$              ;;BR IF NO
016412 126767 162777 162763          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
016420 101015                        BHI     3$              ;;BR IF NO
016422 032777 001000 163010          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
016430 001404                        BEQ     4$              ;;BR IF NO
016432 016767 162752 162746   7$:    MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
016440 000446                        BR      $OVER
016442 105067 162735          4$:    CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
016446 005067 163040                 CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
016452 000415                        BR      1$              ;;ESCAPE TO THE NEXT TEST
016454 032777 004000 162756   3$:    BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
016462 001011                        BNE     1$              ;;BR IF YES
016464 005767 163044                 TST     $PASS           ;;IF FIRST PASS OF PROGRAM
016470 001406                        BEQ     1$              ;;     INHIBIT ITERATIONS
016472 005267 162706                 INC     $ICNT           ;;INCREMENT ITERATION COUNT
016476 026767 163010 162700          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
016504 002024                        BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
016506 012767 000001 162670   1$:    MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
016514 016767 000056 162770          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
016522 105267 162654          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
016526 116767 162650 162776          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
016534 011667 162646                 MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
016540 011667 162644                 MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
016544 005067 162744                 CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
016550 112767 000001 162637          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
016556 016777 162620 162656   $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
016564 016716 162616                 MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
016570 000002                 4$:    RTI
016572 001407                 BRW:   1407
016574 000432                 BRX:   432
016576 000005                 $MXCNT: 5                       ;;MAX. NUMBER OF ITERATIONS
                              .SBTTL  TRAP DECODER

                              ;;**************************************************************
                              ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                              ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                              ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                              ;*GO TO THAT ROUTINE.

016600 010046                 $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
016602 016600 000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
016606 005740                         TST     -(R0)           ;;BACKUP BY 2
016610 111000                         MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
016612 006300                         ASL     R0              ;;POSITION FOR INDEXING
016614 016000 016634                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
016620 000200                         RTS     R0              ;;GO TO ROUTINE


                              ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

016622 011646                 $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
016624 016666 000004 000002           MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
016632 000002                         RTI                     ;;RESTORE THE PSW
```

```

                                              .SBTTL  TRAP TABLE

                                              ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                              ;*BY THE "TRAP" INSTRUCTION.

                                              ;       ROUTINE
                                              ;       -------
016634  016622                                $TRPAD: .WORD    $TRAP2
016636  013034                                        $TYPE    ;;CALL=TYPE      TRAP+1(104401)  TTY TYPEOUT ROUTINE
016640  014552                                        $TYPOC   ;;CALL=TYPOC     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
016642  014526                                        $TYPOS   ;;CALL=TYPOS     TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
016644  014566                                        $TYPON   ;;CALL=TYPON     TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)


016646  013012                                        .SCOP1   ;;CALL=SCOP1     TRAP+5(104405)
016650  013316                                        .INSTR   ;;CALL=INSTR     TRAP+6(104406)
016652  013424                                        .INSTER  ;;CALL=INSTER    TRAP+7(104407)
016654  013434                                        .PARAM   ;;CALL=PARAM     TRAP+10(104410)
016656  013634                                        .SAVOS   ;;CALL=SAVOS     TRAP+11(104411)
016660  013674                                        .RESOS   ;;CALL=RESOS     TRAP+12(104412)
016662  013726                                        .CONVRT  ;;CALL=CONVRT    TRAP+13(104413)
016664  014104                                        .SETFLG  ;;CALL=SETFLG    TRAP+14(104414)
                                              ;*********************************************************
                                                      ;UTILITIES
                                              ;*********************************************************

                                              ;THIS UTILITY CALCULATES PRIORITY LEVEL
016666  006367  000044                        DULEV:  ASL      DUPRT    ;SHIFT LEFT
016672  006367  000040                                ASL      DUPRT    ;
016676  006367  000034                                ASL      DUPRT    ;
016702  006367  000030                                ASL      DUPRT    ;
016706  006367  000024                                ASL      DUPRT    ;
016712  016767  000020  000020                        MOV      DUPRT,LESS1      ;MOVE THIS TO LESS1
016720  162767  000001  000012                        SUB      #1,LESS1         ;CREATE LESS1
016726  042767  000037  000004                        BIC      #37,LESS1        ;CLEAR TNZVC
016734  000207                                        RTS      PC
016736  000240                                DUPRT:  PR5
016740  000200                                LESS1:  PR4      ;LEVEL TO ALLOW INTERRUPTS

                                                      ;NEW DU ADDRESSES
016742  016767  000126  162742                DUADDR: MOV      DUBASE,RXCSR     ;XXX0
016750  005267  000120                                INC      DUBASE
016754  016767  000114  162732                        MOV      DUBASE,HRXCSR    ;XXX1
016762  005267  000106                                INC      DUBASE
016766  016767  000102  162722                        MOV      DUBASE,RXDBUF    ;XXX2
016774  016767  000074  162720                        MOV      DUBASE,PARCSR    ;XXX2
017002  005267  000066                                INC      DUBASE
017006  016767  000062  162704                        MOV      DUBASE,HRXDBUF   ;XXX3
017014  016767  000054  162702                        MOV      DUBASE,HPARCSR   ;XXX3
017022  005267  000046                                INC      DUBASE
017026  016767  000042  162672                        MOV      DUBASE,TXCSR     ;XXX4
017034  005267  000034                                INC      DUBASE
017040  016767  000030  162662                        MOV      DUBASE,HTXCSR    ;XXX5
017046  005267  000022                                INC      DUBASE
017052  016767  000016  162652                        MOV      DUBASE,TXDBUF    ;XXX6
```

```
3638  017060  005267  000010                  INC     DUBASE
3639  017064  016767  000004  162642          MOV     DUBASE,HTXDBUF  ;XXX7
3640  017072  000207                          RTS     PC
3641  017074  000000                  DUBASE: 0
3642
3643                                          ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3644                                          ;INFORMATION CONTAINED IN $TMP1 AND IT IS
3645                                          ;SHIFTED IN BY THE CONTENTS OF SHIFT
3646  017076  042777  040000  162622  RPOKE:  BIC     #MTDATA,@TXCSR
3647  017104  005067  162372                  CLR     $TMP2
3648  017110  006067  162364                  ROR     $TMP1   ;FORCE CARRY
3649  017114  006067  162362                  ROR     $TMP2   ;PICK UP CARRY IN BIT 15
3650  017120  006367  162356                  ASR     $TMP2   ;SHIFT INTO BIT 14
3651  017124  042767  100000  162350          BIC     #BIT15,$TMP2    ;CLR BIT 15
3652  017132  056777  162344  162566          BIS     $TMP2,@TXCSR    ;POKE MAINT DATA
3653  017140  042777  020000  162560          BIC     #CLK,@TXCSR     ;POKE CLK
3654  017146  052777  020000  162552          BIS     #CLK,@TXCSR     ;
3655  017154  005367  161742                  DEC     SHIFT
3656  017160  001346                          BNE     RPOKE
3657  017162  000207                          RTS     PC
3658
3659                                          ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3660  017164  016767  162310  162310  ODD8:   MOV     $TMP1,$TMP2     ;SAVE TEMP1
3661  017172  005067  162306                  CLR     $TMP3
3662  017176  012727  000010                  MOV     #8.,(PC)+
3663  017202  000000                  4$:     0
3664  017204  006067  162272          1$:     ROR     $TMP2
3665  017210  005567  162270                  ADC     $TMP3
3666  017214  005367  177762                  DEC     4$
3667  017220  001371                          BNE     1$
3668  017222  006067  162256                  ROR     $TMP3
3669  017226  103404                          BCS     2$
3670  017230  052767  000400  162242          BIS     #BIT8,$TMP1     ;SET ODD PARITY
3671  017236  000403                          BR      3$
3672  017240  042767  000400  162232  2$:     BIC     #BIT8,$TMP1     ;CLR EVEN PARITY
3673                                          ;$TMP1 NOW HAS ODD PARITY CHARACTER
3674  017246  000207                  3$:     RTS     PC
3675
3676                                          ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3677  017250  016767  162224  162224  EVEN8:  MOV     $TMP1,$TMP2     ;SAVE TEMP1
3678  017256  005067  162222                  CLR     $TMP3
3679  017262  012727  000010                  MOV     #8.,(PC)+
3680  017266  000000                  4$:     0
3681  017270  006067  162206          1$:     ROR     $TMP2
3682  017274  005567  162204                  ADC     $TMP3
3683  017300  005367  177762                  DEC     4$
3684  017304  001371                          BNE     1$
3685  017306  006067  162172                  ROR     $TMP3
3686  017312  103004                          BCC     2$
3687  017314  052767  000400  162156          BIS     #BIT8,$TMP1     ;SET EVEN PARITY
3688  017322  000403                          BR      3$
3689  017324  042767  000400  162146  2$:     BIC     #BIT8,$TMP1     ;CLR ODD PARITY
3690                                          ;$TMP1 NOW HAS EVEN PARITY CHARACTER
3691  017332  000207                  3$:     RTS     PC
3692  017334  062716  000002          TRPREG: ADD     #2,(SP) ;ALLOW IT TO "CRUNCH" INTO ERROR BACK
3693                                                          ;IN MAIN PART OF THE PROGRAM
```

```
DZDUQ-A MACY11 27(1006)  03-FEB-77  07:44  PAGE 74
DZDUQA.M11    27-JAN-77 09:46          TRAP TABLE

 3694  017340  000002                          RTI
 3695          000001                  .END
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
AAA     003200            1291#
ABASE = 000000             874     915
ACDW1 = 000000             874     917
ACDW2 = 000000             874     918
ACPUOP= 000000             874     889
ACTREG  001166             733#   1247*   1261*   1262*   1269*   2809    2812    2822
ADDWD = 000000             874     919
ADDWI = 000000             874     920
ADDWI0= 000000             874     929
ADDWI1= 000000             874     930
ADDWI2= 000000             874     931
ADDWI3= 000000             874     932
ADDWI4= 000000             874     933
ADDWI5= 000000             874     934
ADDW2 = 000000             874     921
ADDW3 = 000000             874     922
ADDW4 = 000000             874     923
ADDW5 = 000000             874     924
ADDW6 = 000000             874     925
ADDW7 = 000000             874     926
ADDW8 = 000000             874     927
ADDW9 = 000000             874     928
ADEVCT= 000000             874     880
ADEVM = 000000             874     916
ADRCNT= 013633            3024*   3063*   3070#
AENV  = 000000             874     885
AENVM = 000000             874     886
AFATAL= 000000             874     877
AMADR1= 000000             874     902
AMADR2= 000000             874     906
AMADR3= 000000             874     909
AMADR4= 000000             874     912
AMAMS1= 000000             874     896
AMAMS2= 000000             874     904
AMAMS3= 000000             874     907
AMAMS4= 000000             874     910
AMSGAD= 000000             874     882
AMSGLG= 000000             874     883
AMSGTY= 000000             874     876
AMTYP1= 000000             874     897
AMTYP2= 000000             874     905
AMTYP3= 000000             874     908
AMTYP4= 000000             874     911
APASS = 000000             874     879
APRIOR= 000000             874
APTCSU= 000040             534#   2939
APTENV= 000001             534#   2932    3186
APTSIZ= 000200             534#   1166
APTSPO= 000100             534#   2934
ASWREG= 000000             874     887
ATESTN= 000000             874     878
AUNIT = 000000             874     881
AUSWR = 000000             874     888
AVECT1= 000000             874     913
AVECT2= 000000             874     914
BASEAD  001154             728#   1229*   1266*   1267    1273*   1275*   2816*   2828*   2832
```

DZDUBA.M11 27-JAN-77 09:46 CROSS REFERENCE TABLE -- USER SYMBOLS

```
BASEIV  001162          731#  1239*  2817*  2829*  2834   2835*  2836   2837*  2838   2839*  2840   2841*
BBB     003026         1246   1250#
BINWRD  014102         3105*  3107   3137#
BITW  = 002000          800#   993#  1854   1861   1873   1880   1890   1905   1909   1916
BIT0  = 000001          663#   776    807    969   1000   1841
BIT00 = 000001          653#   663
BIT01 = 000002          652#   662
BIT02 = 000004          651#   661
BIT03 = 000010          650#   660
BIT04 = 000020          649#   659
BIT05 = 000040          648#   658
BIT06 = 000100          647#   657
BIT07 = 000200          646#   656
BIT08 = 000400          645#   655   3524
BIT09 = 001000          644#   654   3196   3532
BIT1  = 000002          662#   775    968
BIT10 = 002000          643#   766    800    959    993   3174
BIT11 = 004000          642#   765    958   1708   1709   1712   1713   1717   1719   3539
BIT12 = 010000          641#   764    781    957    974   1730   1731   1734   1735   1739   1741
BIT13 = 020000          640#   763    780    799    956    973    992   3181
BIT14 = 040000          639#   762    779    798    955    972    991   3510
BIT15 = 100000          638#   761    778    797    954    971    990   3651
BIT2  = 000004          661#   774    967
BIT3  = 000010          660#   773    806    966    999
BIT4  = 000020          659#   772    805    965    998
BIT5  = 000040          658#   771    804    964    997   1852
BIT6  = 000100          657#   770    803    963    996
BIT7  = 000200          656#   769    802    962    995
BIT8  = 000400          655#   768    784    801    961    977    994   3670   3672   3687   3689
BIT9  = 001000          654#   767    783    960    976
BPTVEC= 000014          670#
BREAK = 000001          807#  1000#  1606   1607   1610   1611   1615   1617   1871   1886   1903   1915   2348
                       2375   2411   2484   2557   2630   2698   2753
BRW     016572         3557#
BRX     016574         3558#
CARDET= 010000          764#   957#
CCC     012564         2808   2831   2848#
CHRCNT  014100         3103*  3108   3121*  3135*  3136
CKSWR   012664         2802   2864   2872#  2904   3501
CLK   = 020000          799#   992#  1750   1751   1754   1755   1759   1761   2348   2375   2411   2418   2419
                       2421   2422   2484   2491   2492   2494   2495   2557   2564   2565   2567   2568   2630
                       2637   2638   2640   2641   2698   2704   2705   2707   2708   2753   2759   2760   2762
                       2763   3653   3654
CNTLW   012740         1196   2883#
CONVRT= 104413         2805   2865   3367   3603#
COUNT   001124          708#  2423*  2427*  2496*  2500*  2569*  2573*  2642*  2646*
CR    = 000015          578#  2978   2988
CRLF  = 000200          579#  2949   2988
CTS   = 020000          763#   956#
DDISP = 177570          585#   641   1154
DEVADR  013630         3022*  3060   3068#
DEVICE  015273         2806   3397#
DF1     002132         1025   1029   1033   1037   1083#
DH1     002067         1023   1027   1031   1072#
DISPLA  001442          641#  1154*  1162*  1187*  3173*  3554*
DISPRE  000174          682#  1162   1187
```

DZDUQA.M11 27-JAN-77 09:46 CROSS REFERENCE TABLE -- USER SYMBOLS

```
DNA    = 100000            797#   990#  1971
DNAINT= 000040             804#   997#  1666  1667  1670  1671  1675  1677
DSC    = 100000            761#   954#  1950
DSINTE= 000040             771#   964#  1546  1547  1550  1551  1555  1557
DSR    = 001000            767#   960#
DSWR   = 177570            584#   840   1153
DTR    = 000002            775#   968#  1430  1431  1434  1435  1439  1445  1449  2120  2189  2274  2320
DTI      002116           1024  1028    1032  1078#
DT4      002126           1036  1081#
DUADDR   016742           1228  2833    3623#
DUBASE   017074           1224  1227    2832#  3623  3624#  3625  3626#  3627  3628  3629#  3630  3631  3632#
                          3633  3634#   3635   3636#  3637  3638#  3639  3641#
DULEV    016666           1206  3610#
DUPRT    016736           1205#  3610#  3611#  3612#  3613#  3614#  3615  3619#
DURIS    001740           1052#  2836#
DURIV    001736           1051#  1235    1238  1239  2834#  2841
DUTIS    001744           1054#  2840#
DUTIV    001742           1053#  2838#
EIGHT  = 000010            792#   985#  2351  2378  2633
EMTVEC= 000030             673#  1138#  1139#
EM1      001762           1022  1058#
EM2      002022           1026  1064#
EM3      002043           1030  1067#
EM4      001746           1034  1056#
ERRCNT   001114            701#
ERRVEC= 000004             666#  1151    1152#  1163#  3515  3516#  3518#  3521#
EVENB    017250           3677#
EVEPAR= 001400             795#   988#
EVPAR  = 000400            784#   977#
FIVE   = 000005            789#   982#  2414  2701  2756
FPMERR= 020000             780#   973#  2005
GNS    = ######  U        3591  3592    3593  3594  3597  3598  3599  3600  3601  3602  3603  3604
HOXEN  = 000010            806#   999#  1626  1627  1630  1631  1635  1637
HILIM    013626           3021#  3051    3067#
HLD0     001130            710#  1078    3204#
HLD1     001132            711#  1078    3205#
HLD2     001134            712#  3206#
HLD3     001136            713#  3207#
HLD4     001140            714#  3208#
HLD5     001142            715#  3209#
HLD6     001144            716#  3210#
HOLD     001120            706#  1357#  1487  1811  2062  2125  2146  2171  2194  2209  2224  2248  2263
                          2279  2295    2310  2325
HPARCS   001724           1045#  1382#  3631#
HRXCSR   001714           1041#  1351#  1841#  3625#
HRXDBU   001720           1043#  1367#  3630#
HT     = 000011            576#  2747    2808
HTXCSR   001730           1047#  1397#  1852#  3635#
HTXDBU   001734           1049#  1412#  3639#
INBUF    016114           1294  1298    2997   3027  3145  3149  3476#
INIFLG   001172            741#  1197   1200#
INSTER= 104407            1302  3046    3154   3599#
INSTR  = 104406           1219  1230    1240   1251  1276  1292  1305  1309  1313  1317  1328  2887  3598#
INSTR2   013432           2994  3006    3015#
IOTVEC= 000020             671#  1136#  1137#
ISYMOD= 000000             788#   981#  2371   2378  2694  2701  2749  2756
```

```
JMRBY   001153            724#  1205*  1320  1901  2109
KEEPAD  001156            729#  1227*  1229  1273  1275  2828
KEEPIV  001164            732#  1238*  2829
LASTAD  001160            730#  1256   1267  1281
LESS1   016740           3615*  3616*  3617* 3620#
LF    = 000012            577#  2982   2988
LIGHTS  001102            691#  2853*
LIMITS  013560           3037   3051#
LOBITS  013632           3023*  3055   3069#  3070
LOCK    001110            699#  2907
LOKFLG  001174            743#
LOLIM   013624           3020*  3053   3066#
MASK1   007312           2177#
MASK2   007330           2183#
MASK3   007420           2215#
MCNTG   016061           2882   3466#
MCOM    015305           2811   3399#
MCRLF   015405           3099   3412#
MDATA   016220           3119   3130   3480#
MEPASS  015454           2804   3419#
MEXT  = 010000            811#  1004#  1903  1915
MEXTJ   016021           1318   3460#
MINT  = 004000            810#  1003#  1871  1886  2348  2375  2411  2484  2557  2630  2698  2753
MLASTD  015241           1252   3392#
MLOCK   015520           3426#
MMNEW   016102           2887   3470#
MMSWR   016071           2884   3468#
MMULT   015212           1241   3388#
MPFAIL  015410           3370   3413#
MQM     015401           3013   3411#
MR      015501           1338   3423#
MRANGE  015335           1277   3404#
MREGAD  015162           1220   3384#
MRESET= 000400            801#   994#  1440  1469  1507  1536  1556  1576  1596  1616  1636  1656  1676
                         1696   1718   1740  1760  1780  1822  1823  1826  1827  1836  1851  1885  1914
                         1927   1938   1949  1960  1970  1981  1993  2004  2015  2026  2037  2075  2088
                         2104   2343   2345  2370  2372  2406  2408  2479  2481  2552  2554  2625  2627
                         2693   2695   2748  2750
MSYNC   015553           1293   3431#
MTDATA= 040000            798#   991#  1770  1771  1774  1775  1779  1781  1871  1876  1884  1903  1908
                         1913   2348   2375  2411  2484  2557  2630  2698  2753  3646
MTITLE  015114           1199   3377#
MTSTPC  015505           1329   3424#
MULTD   001152            723#  1243   1244  2807
MVECTO  015147           1231   3382#
MWIRE4  015742           1314   3452#
MWIRE5  015672           1310   3445#
MWIRE6  015621           1306   3438#
NEXT    001106            698#
NOPAR = 000000            793#   986#  2351  2378  2414  2487  2560  2633  2701  2756
ODDPAR= 001000            794#   987#
ODDB    017164           3660#
ONCE    002610           1198   1201#
OPTCLR  001151            722#  1316   1443  1472  1510  1796  1837  2040
OUT     012772           2873   2875   2877  2881  2893#
OUTCRY  012556           2805   2844#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
OUTMUL  003166            1249    1274    1285#
OUT1    007762            2112    2240    2289    2335#
OUT2    007506            2163    2243#
OUT3    007636            2165    2290#
OVRRUN= 040000             779#    972#    2016
PARAM = 104410            1221    1232    1253    1278    1330    2888    3600#
PARAM1  013464            3026#   3047
PARCSR  001722            1044#   1379#   2344#   2351#   2371#   2378#   2407#   2414#   2480#   2487#   2553#   2560#   2626#
                          2633#   2694#   2701#   2749#   2756#   3628#
PAREN = 001000             783#    976#
PARER = 010000             781#    974#    1994
PARERR  013540            3029    3031    3033    3042#   3052    3054    3056
PARTI   013622            3045    3065#
PASCNT  001112             700#   1174#   2852#   2853    2854
PFTAB   015106            3368    3374#
PIRQ  = 177772             583#
PIROVE= 000240             677#
POPRO = 012600             756#    949#
POP1SP= 005726             754#    947#
POP2SP= 022626             758#    951#
PR0   = 000000             600#
PR1   = 000040             601#
PR2   = 000100             602#
PR3   = 000140             603#
PR4   = 000200             604#   3620
PR5   = 000240             605#   3619
PR6   = 000300             606#
PR7   = 000340             607#   1347    1363    1378    1393    1408
PS    = 177776             580#    581
PSW   = 177776             581#
PUSHRO= 010046             755#    948#
PUSH1S= 005746             753#    946#
PUSH2S= 024646             757#    950#
PWRVEC= 000024             672#   1142#   1143#
RDSW    013000            2883#   2893#   2895#   3044
RECACT= 004000             765#    958#   1939    2352    2356    2360    2379    2383    2387    2435    2444    2448    2508
                          2517    2521    2581    2590    2594    2654    2663    2667
REPLAY  012462            2824    2830    2832#
RESTAR  015012            3348    3354#
RESTRT  012642            2825    2856    2863#
RESVEC= 000010             667#
RESOS = 104412            3602#
RETURN  001104             697#   1179#   1337#   1339    3373
RING  = 040000             762#    955#
RINTEN= 000100             770#    963#   1566    1567    1570    1571    1575    1577
ROTADD  001170             736#   1248#   1260#   1262    1264#   1269    1272#   2819#   2822    2826#
RPOKE   017076            2426    2440    2499    2513    2572    2586    2645    2659    2713    2725    2729    2768    2780
                          2784    3646#   3656
RTS   = 000004             774#    967#   1459    1460    1463    1464    1468    1474    1478    2141    2204    2258    2305
RUNA  = 000000                1      51     137     511
RUNB  = ******  U            21      56     140     516
RUNC  = ******  U            21      56     140     516
RUND  = ******  U            21      56     140     516
RUNE  = ******  U            21      56     140     516
RUNF  = ******  U            21      56     140     516
RUNIT   012366            2810    2816#   2823
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
RXCSR   001712          1040#  1348*  1430*  1431   1434*  1435   1439*  1445   1449   1459*  1460   1463*  1464
                        1468*  1474   1478   1497*  1498   1501*  1502   1506*  1512   1516   1526*  1527   1530*
                        1531   1535*  1537   1546*  1547   1550*  1551   1555*  1557   1566*  1567   1570*  1571
                        1575*  1577   1586*  1587   1590*  1591   1595*  1597   1790*  1794   1839*  1840   1842
                        1846*  1847   1928   1939   1950   2036*  2038   2039   2103*  2114   2115   2120*  2130
                        2135   2141*  2151   2156   2166*  2176   2182   2189*  2199   2204*  2214   2219*  2229
                        2234   2243*  2253   2258*  2268   2274*  2284   2290*  2300   2305*  2315   2320*  2330
                        2352   2355*  2356   2359*  2360   2379   2382*  2383   2386*  2387   2416*  2432   2435
                        2441   2444   2447*  2448   2451   2459   2489*  2505   2508   2514   2517   2520*  2521
                        2524   2532   2562*  2578   2581   2587   2590   2593*  2594   2597   2605   2635*  2651
                        2654   2660   2663   2666*  2667   2670   2678   2702*  2714   2757*  2769   2846   3623*
RXDBUF  001716          1042#  1364*  1982   1984   1994   2005   2016   2027   2089   2090   2415   2455   2488
                        2528   2561   2601   2634   2674   2709   2717   2732   2764   2772   2787   3627*
RXDONE= 000200           769#   962#  1928
RXERR = 100000           778#   971#  2027
SAVIT   014334          3183   3204#
SAVPC   001126           709#  3074#
SAVSP   001116           702#  3347*  3354
SAVOS = 104411          3601#
SCOP1 = 104405          3597#
SEND  = 000020           805#   998#  1646   1647   1650   1651   1655   1657
SEREC   001150           721#  1312   2164
SETFLG= 104414          1242   1307   1311   1315   1319   3604#
SEVEN = 004000           791#   984#  2560
SEXMIT  001147           720#  1308   2162
SHIFT   001122           707#  2424*  2438*  2497*  2511*  2570*  2584*  2643*  2657*  2711*  2723*  2727*  2766*
                        2778*  2782*  3655*
SIX   = 002000           790#   983#  2487
SPACNT= 014101          3104*  3123   3126*  3136#
SRD   = 002000           766#   959#
STACK = 001100           571#  1134   1170   1324   3363
STD   = 000010           773#   966#  1497   1498   1501   1502   1506   1512   1516   2166   2219   2243   2290
STFLG   001173           742#  1173*
STKLMT= 177774           582#
STPSYN= 000400           768#   961#  1586   1587   1590   1591   1595   1597
SVOS    013642          3078#
SWR     001440           840#  1132   1153*  1155   1161*  1168*  1184   1186*  1191   1207   1326   2874   2905
                        3174   3181   3193   3196   3510   3524   3526   3532   3539
SWREG   000176           683#  1161   1186   1191   2874   2891   2898
SWREGL  013002          2886   2896#
SW0   = 000001           635#
SW00  = 000001           625#   635   1207
SW01  = 000002           624#   634   1326
SW02  = 000004           623#   633
SW03  = 000010           622#   632
SW04  = 000020           621#   631
SW05  = 000040           620#   630
SW06  = 000100           619#   629
SW07  = 000200           618#   628
SW08  = 000400           617#   627
SW09  = 001000           616#   626   2905
SW1   = 000002           634#
SW10  = 002000           615#
SW11  = 004000           614#
SW12  = 010000           613#
SW13  = 020000           612#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
SW14   = 040000            611#
SW15   = 100000            610#
SW2    = 000004            633#
SW3    = 000010            632#
SW4    = 000020            631#
SW5    = 000040            630#
SW6    = 000100            629#
SW7    = 000200            628#
SW8    = 000400            627#
SW9    = 001000            626#
SYNCNO  001146             719#  1296#  1300#  2430  2503  2576  2649
SYNEXT= 020000             787#   980#  2344  2351
SYNINT= 030000             786#   979#  2407  2414  2480  2487  2553  2560  2626  2633
SYNSCH= 000020             772#   965#  1526  1527  1530  1531  1535  1537  2355  2359  2382  2386  2416
                          2447   2489  2520  2562  2593  2635  2666  2702  2757
SYSTST= 014000             812#  1005#
TBITVE= 000014             668#
TEMP    016156            3109  3364#  3365#  3478#
TKVEC  = 000060             675#
TPVEC  = 000064             676#
TRAPVE= 000034             674#  1140#  1141#
TRPREG  017334            1346  1362  1377  1392  1407  3692#
TRTVEC= 000014             669#
TST1    003362            1331  1337  1345#  2865  2866  3503
TST10   004042            1458#
TST11   004166            1496#
TST12   004302            1525#
TST13   004372            1545#
TST14   004462            1565#
TST15   004552            1585#
TST16   004642            1605#
TST17   004732            1625#
TST2    003442            1361#
TST20   005022            1645#
TST21   005112            1665#
TST22   005202            1685#
TST23   005272            1707#
TST24   005362            1729#
TST25   005452            1749#
TST26   005542            1769#
TST27   005632            1789#
TST3    003514            1376#
TST30   005734            1821#
TST31   005776            1835#
TST32   006150            1870#
TST33   006266            1898#
TST34   006372            1926#
TST35   006414            1937#
TST36   006436            1948#
TST37   006460            1959#
TST4    003566            1391#
TST40   006502            1969#
TST41   006524            1980#
TST42   006556            1992#
TST43   006600            2003#
TST44   006622            2014#
```

```
TST45   006644          2025#
TST46   006666          2035#
TST47   006770          2073#
TST5    003640          1406#
TST50   007030          2087#
TST51   007062          2102#
TST52   007762          2342#
TST53   010074          2369#
TST54   010206          2405#
TST55   010514          2478#
TST56   011022          2551#
TST57   011330          2624#
TST6    003712          1421#
TST60   011636          2692#
TST61   012064          2747#
TST7    003726          1429#
TTST    016300          3506#
TXCSR   001726          1046#  1394*  1440*  1469*  1507*  1536*  1556*  1576*  1596*  1606*  1607   1610*  1611
                        1615*  1616*  1617   1626*  1627   1630*  1631   1635*  1636*  1637   1646*  1647   1650*
                        1651   1655*  1656*  1657   1666*  1667   1670*  1671   1675*  1676*  1677   1686*  1687
                        1690*  1691   1695*  1696*  1697   1708*  1709   1712*  1713   1717*  1718*  1719   1730*
                        1731   1734*  1735   1739*  1740*  1741   1750*  1751   1754*  1755   1759*  1760*  1761
                        1770*  1771   1774*  1775   1779*  1780*  1781   1791*  1795   1822*  1823   1826*  1827
                        1836*  1851*  1853   1859*  1860   1871*  1873   1876*  1880   1884*  1885*  1886*  1890
                        1903*  1905   1908*  1909   1913*  1914*  1915*  1916   1927*  1938*  1949*  1960*  1961
                        1970*  1971   1981*  1993*  2004*  2015*  2026*  2037*  2074*  2075*  2076   2077   2088*
                        2104*  2343*  2345*  2348*  2370*  2372*  2375*  2406*  2408*  2411*  2418*  2419*  2421*
                        2422*  2479*  2481*  2484*  2491*  2492*  2494*  2495*  2552*  2554*  2557*  2564*  2565*
                        2567*  2568*  2625*  2627*  2630*  2637*  2638*  2640*  2641*  2693*  2695*  2698*  2704*
                        2705*  2707*  2708*  2748*  2750*  2753*  2759*  2760*  2762*  2763*  3633*  3646*  3652*
                        3653*  3654*
TXDBUF  001732          1048#  1409*  1422   3637*
TXDONE= 000200           802#   995#  1961
TXINTE= 000100           803#   996#  1686   1687   1690   1691   1695   1697
TYPE  = 104401          1199   1338   2803   2806   2811   2882   2884   2952   2995   3012   3098   3129   3176
                        3184   3220   3237   3239   3242   3244   3248   3255   3319   3369   3591#
TYPOC = 104402          3228   3252   3592#
TYPON = 104404          3594#
TYPOS = 104403          3593#
USER  = 000000           809#  1002#
VOID  = 000001           776#   969#
WRDCNT  014076          3102*  3131*  3134#
ZERO    012374          2817#
$APTHD  002136          1105   1111#
$ASTAT= ###### U         534
$ATYC   000024R          534#
$ATYI   000000R          534#
$ATYJ   000006R          534#  2937
$ATYK   000016R          534#  3189
$AUTOB  001434           837#
$BASE   001602           915#
$BDADR  001422           832#
$BDDAT  001426           834#
$BELL   001516           866#  3176   3204
$CDW1   001606           917#
$CDW2   001610           918#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
$CHARC  013312          2954*  2964*  2971   2980*  2985*
$CKSUM= ****** U        3597
$CMTAG  001400           820#  1129   1130   1138   1144   1145
$CM1    = 000006         852#   853#   854#   855#   856#   857#   858#
$CM2    = 000014         852#   853#   854#   855#   856#   857#   858#
$CM3    = 000006         850#   852
$CM4    = 000006         858#   859#   860#   861#   862#   863#   864#
$CPUOP  001554           889#
$CRLF   001523           868#  2953   2988   3184   3204   3220   3239   3244   3248
$DDW0   001612           919#
$DDW1   001614           920#
$DDW10  001636           929#
$DDW11  001640           930#
$DDW12  001642           931#
$DDW13  001644           932#
$DDW14  001646           933#
$DDW15  001650           934#
$DDW2   001616           921#
$DDW3   001620           922#
$DDW4   001622           923#
$DDW5   001624           924#
$DDW6   001626           925#
$DDW7   001630           926#
$DDW8   001632           927#
$DDW9   001634           928#
$DEVCT  001536           880#
$DEVM   001604           916#
$E      = 000002         532#
$ENDAD  012632          1092   2859#
$ENV    001546           534    885#  1201   2932   2993   3186
$ENVM   001547           534    886#  1166   2934   2939
$ERFLG  001403           823#  1175*  2851*  3171*  3204   3371*  3487   3528   3530   3536*  3560
$ERMAX  001415           829#  1146*  3530   3553*  3560
$ERROR  014152          1138   3170#
$ERRPC  001416           830#  1078   1081   1177*  2850*  3178*  3179*  3180   3204   3226   3372*  3502*
$ERRTB  001652          1020#  3234
$ERRTY  014372          3219#
$ERTTL  001412           827#  1176*  3177*  3204
$ESCAP  001514           865#  1145*  3199   3201   3204   3552*
$ETABL  001546           884#
$ETEND  001652           937#  1117
$FATAL  001530           534#   877#
$FFLG   000244R          534#*
$FILLC  001456           848#  2957   2988
$FILLS  001455           847#  2988
$GDADR  001420           831#
$GDDAT  001424           833#
$GTSUM= ****** U        3596
$HIBTS  002136          1112#
$ICNT   001404           824#  3543*  3544   3546*  3559
$INTAG  001435           838#
$ITEMB  001414           828#  3180*  3188   3204   3223
$LF     001524           869#  2988   3204
$LFLG   000243R          534#*
$LPADR  001406           825#  1147*  2865*  3534*  3550*  3555   3559
$LPERR  001410           826#  1148*  3198   3534   3551*  3559
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
SHADR1  001540            902#
SHADR2  001544            906#
SHADR3  001570            909#
SHADR4  001574            912#
SHAIL   001526            875#  1113    1117    1165    2932    3186    3549
SHANS1  001556            896#
SHANS2  001562            904#
SHANS3  001566            907#
SHANS4  001572            910#
SHHDR   002140           1113#
SHFLG   000242R           534#*
SHSGAD  001542            534*   882#
SHSGLG  001544            534*   883#
SHSGTY  001526            534*   876#
SHTYP1  001557            897#
SHTYP2  001563            905#
SHTYP3  001567            908#
SHTYP4  001573            911#
SMXCNT  016576           3547   3559#
SN      =  000000         532#   2795#
SNULL   001454            846#   2959   2988
SNWTST= 000000           1344#  1360#  1375#  1390#  1405#  1420#  1428#  1457#  1495#  1524#  1544#  1564#  1584#
                         1604#  1624#  1644#  1664#  1684#  1706#  1728#  1748#  1768#  1788#  1820#  1834#  1869#
                         1897#  1925#  1936#  1947#  1958#  1968#  1979#  1991#  2002#  2013#  2024#  2034#  2072#
                         2086#  2101#  2341#  2368#  2404#  2477#  2550#  2623#  2691#  2746#
SOCNT   014750           3291#  3320#  3333#
SOMODE  014752           3286#  3290#  3295   3298#  3309#  3335#
SOVER   016556           3508   3511   3527   3535   3545   3554#
SPASS   001534            879#  1165#  2854#  3541   3560
SPASTM  002144           1115#
SPWRON  014754           1142   3339#
SQUES   001522            867#  2988   3204
SRDCHR= ******  U        3597
SRDDEC= ******  U        3597
SRDLIN= ******  U        3597
SRDOCT= ******  U        3597
SREGAD  001460            850#
SREG0   001462            852#  3083#  3088
SREG1   001464            853#  3082#  3089
SREG2   001466            854#  3081#  3090
SREG3   001470            855#  3080#  3091
SREG4   001472            856#  3079#  3092
SREG5   001474            857#  3078#  3093
SR2A    =  ******  U      3597
SSAVRE= ******  U        3597
SSCOPE  016262           1136   3496#
SSETUP= 000017           1118#  1135   1136   1138   1140   1142   1144   1145   1147   3171   3196   3203   3497
SSTUP   =  177777        1118#
SSVLAD  016522           3519   3548#
SSVPC   =  002136        1090#  1095
SSWR    =  177400         523#   864    865    866   1144   1145   1147   1148   1346   1362   1377   1392   1407
                         1422   1430   1459   1497   1526   1546   1566   1586   1606   1626   1646   1666   1686
                         1708   1730   1750   1770   1790   1822   1836   1871   1899   1927   1938   1949   1960
                         1970   1981   1993   2004   2015   2026   2036   2074   2088   2103   2343   2370   2406
                         2479   2552   2625   2693   2748   3162   3163   3164   3165   3166   3174   3181   3193
                         3196   3204   3488   3489   3490   3491   3492   3510   3522   3524   3525   3528   3529
```

```
                    3530    3537    3538    3539    3551    3554    3559
$SWREG  001550       887#    1168
$SWMK=  000000      3492    3493    3526
$TESTN  001532       878#    3549#
$TIMES  001512       864#    1144#   3537#   3544    3547#   3559
$TKB    001446       843#    2878    3001    3009    3509
$TKS    001444       842#    2876    2999    3507
$TMPO   001476       858#
$TMP1   001500       859#    2425#   2439#   2498#   2512#   2571#   2585#   2644#   2658#   2712#   2724#   2728#   2767#
                    2779#   2783#   3648#   3660    3670#   3672#   3677    3687#   3689#
$TMP2   001502       860#    3647#   3649#   3650#   3651#   3652    3660#   3664#   3677#   3681#
$TMP3   001504       861#    3661#   3665#   3668#   3678#   3682#   3685#
$TMP4   001506       862#
$TMP5   001510       863#
$TN     = 000062     534#    1344    1346#   1360    1362#   1375    1377#   1390    1392#   1405    1407#   1420    1422#
                    1428    1430#   1457    1459#   1495    1497#   1524    1526#   1544    1546#   1564    1566#   1584
                    1586#   1604    1606#   1624    1626#   1644    1646#   1664    1666#   1684    1686#   1706    1708#
                    1728    1730#   1748    1750#   1768    1770#   1788    1790#   1820    1822#   1834    1836#   1869
                    1871#   1897    1899#   1925    1927#   1936    1938#   1947    1949#   1958    1960#   1968    1970#
                    1979    1981#   1991    1993#   2002    2004#   2013    2015#   2024    2026#   2034    2036#   2072
                    2074#   2086    2088#   2101    2103#   2341    2343#   2368    2370#   2404    2406#   2477    2479#
                    2550    2552#   2623    2625#   2691    2693#   2746    2748#
$TPB    001452       845#    2977#   2988    3009#
$TPFLG  001457       849#    2926    2988
$TPS    001450       844#    2975    2988    3007
$TRAP   016600      1140    3568#
$TRAP2  016622      3579#    3590
$TRP    = 000015    3583#    3592#   3593#   3594#   3595#   3597    3598#   3599#   3600#   3601#   3602#   3603#   3604#
                    3605#
$TRPAD  016634      3573    3590#
$TSTM   002142      1114#
$TSTMM  001402       822#    1178#   2849#   3173    3204    3210    3487    3526    3548#   3549    3554    3560
$TYPBM= ****** U    3595
$TYPDS= ****** U    3595
$TYPE   013034       534    2926#   3583    3591
$TYPEC  013246      2956    2963    2970    2975#   2976
$TYPEX  013314      2981    2983    2986#
$TYPOC  014552      3289#    3592
$TYPON  014566      3288    3291#   3594
$TYPOS  014526      3284#    3593
$UNIT   001540       881#
$UNITM  002146      1116#
$USWR   001552       888#    1203
$VECT1  001576       913#
$VECT2  001600       914#
$XTSTR  016326      3504    3513#
$OFILL  014751      3285#    3289#   3299    3334#
$40CAT= ****** U    3183    3510
.       = 017342     534#     568#    681#    684#    689#    744#    745#    819#    870    1076#   1090    1091#   1093#
                    1095#   1101    1102#   1104#   1106#   1133    1147    1148    1349    1352    1365    1368    1380
                    1383    1395    1398    1410    1413    1423    1432    1436    1446    1450    1461    1465    1475
                    1479    1489    1499    1503    1513    1517    1528    1532    1538    1548    1552    1558    1568
                    1572    1578    1588    1592    1598    1608    1612    1618    1628    1632    1638    1648    1652
                    1658    1668    1672    1678    1688    1692    1698    1710    1714    1720    1732    1736    1742
                    1752    1756    1762    1772    1776    1782    1801    1805    1813    1824    1828    1844    1849
                    1857    1863    1874    1881    1891    1906    1910    1917    1929    1940    1951    1962    1972
```

```
                    1986    1995    2006    2017    2028    2046    2053    2064    2080    2093    2110    2118    2127
                    2133    2138    2148    2154    2159    2173    2180    2186    2196    2202    2211    2217    2226
                    2232    2237    2250    2256    2265    2272    2281    2287    2297    2303    2312    2318    2327
                    2333    2353    2357    2361    2380    2384    2388    2433    2436    2442    2445    2449    2452
                    2457    2460    2506    2509    2515    2518    2522    2525    2530    2533    2579    2582    2588
                    2591    2595    2598    2603    2606    2652    2655    2661    2664    2668    2671    2676    2679
                    2715    2719    2734    2770    2774    2789    2988    3204    3258#   3350    3366    3477#   3479#
                    3481#   3559    3560
.BEGIN  003310      1195    1206    1209    1324#
.CONVR  013726      3098#   3603
.EOP    012312      2801#
.INSTE  013424      3012#   3599
.INSTR  013316      2991#   3598
.INST1  013336      2995#   3004    3014
.MSG    013340      2878*   2879*   2880    2991*   2996#
.PARAM  013434      3019#   3600
.PFAIL  014754      1172    3340#   3361
.RESUS  013674      3088#   3602
.SAVUS  013634      3074#   3601
.SCOPE  016262      3500#
.SCOPI  013012      2904#   3597
.SETFL  014104      3144#   3155    3604
.START  002152       685    1127#   1179    2815
.$ASTA= ****** U     534
.SX   = 002136      1101#   1106
```

```
COMMEN    678#
ENDCOM    678#
ERROR     572#  1350  1353  1366  1369  1381  1384  1396  1399  1411  1414  1424  1433  1437  1447
          1451  1462  1466  1476  1480  1500  1504  1514  1518  1529  1533  1539  1549  1553  1559
          1569  1573  1579  1589  1593  1599  1609  1613  1619  1629  1633  1639  1649  1653  1659
          1669  1673  1679  1689  1693  1699  1711  1715  1721  1733  1737  1743  1753  1757  1763
          1773  1777  1783  1802  1806  1825  1829  1845  1850  1858  1864  1875  1882  1892  1907
          1911  1918  1930  1941  1952  1963  1973  1987  1996  2007  2018  2029  2047  2054  2081
          2094  2119  2134  2139  2155  2160  2181  2187  2203  2218  2233  2238  2257  2273  2288
          2304  2319  2334  2354  2358  2362  2381  2385  2389  2434  2437  2443  2446  2450  2453
          2458  2461  2507  2510  2516  2519  2523  2526  2531  2534  2580  2583  2589  2592  2596
          2599  2604  2607  2653  2656  2662  2665  2669  2672  2677  2680  2716  2720  2735  2771
          2775  2790
ESCAPE    678#
GETPRI    678#
GETSWR    678#
MULT      678#
NEWTST    678#  1344  1360  1375  1390  1405  1420  1428  1457  1495  1524  1544  1564  1584  1604
          1624  1644  1664  1684  1706  1728  1748  1768  1788  1820  1834  1869  1897  1925  1936
          1947  1958  1968  1979  1991  2002  2013  2024  2034  2072  2086  2101  2341  2368  2404
          2477  2550  2623  2691  2746
POP        534   678#
PRGEND     556#  2794
PRGFRT     535#   557
PUSH       534   678#
PUSSYF     553#
REPORT     523#   678#
RSETUP     540#  2343  2370  2406  2479  2552  2625  2693  2748
SC         532#  3497
SCOPE      573#  1345  1361  1376  1391  1406  1421  1429  1458  1496  1525  1545  1565  1585  1605
          1625  1645  1665  1685  1707  1729  1749  1769  1789  1821  1835  1870  1898  1926  1937
          1948  1959  1969  1980  1992  2003  2014  2025  2035  2073  2087  2102  2342  2369  2405
          2478  2551  2624  2692  2747  2801
SC1        532#  3556
SETPRI     678#
SETTRA     523#  3583#  3592  3593  3594  3597  3598  3599  3600  3601  3602  3603  3604
SETUP      678#  1127
SKIP       678#
SLASH      678#
SPACE      678#
STARS      534    678#   815   870   873  1088  1098  1100  1107  1344  1360  1375  1390  1405  1420
          1428  1457  1495  1524  1544  1564  1584  1604  1624  1644  1664  1684  1706  1728  1748
          1768  1788  1820  1834  1869  1897  1925  1936  1947  1958  1968  1979  1991  2002  2013
          2024  2034  2072  2086  2101  2341  2368  2404  2477  2550  2623  2691  2746  2911  3158
          3214  3261  3484  3562
SWRSU      678#  1149#
TRMTRP    3583#
TSETUP     541#
TYPBIN     678#
TYPDEC     678#
TYPNAM     678#
TYPNUM     678#
TYPOCS     678#
TYPOCT     678#  3226  3250
TYPTXT     678#
$BEGIN     532#  1321
```

CROSS REFERENCE TABLE -- MACRO NAMES

```
$BINAR  552#
$BUFFE  534#  3473
$CABLE  554#  1483  1807  2058  2121  2142  2167  2190  2205  2220  2244  2259  2275  2291  2306
        2321
$CATCH  526#   568
$CK     533#  2867
$CLRVE  532#  1201
$CONVR  534#  3095
$DNA    550#
$EOP    533#  2794
$GETFL  532#  1240  1305  1309  1313  1317
$GETPA  532#  1219  1230  1250  1276  1328
$GETSY  532#  1287
$HEADE  523#   557
$INSTR  534#  2988
$ISOB   548#
$MATCH  545#
$MAR    539#  1922  1933  1944  1965  1988  1999  2010  2021
$MARW   537#  1425  1454  1492  1521  1541  1561  1581  1601  1621  1641  1661  1681  1703  1725
        1745  1765
$MRW    538#  1817
$MSG    534#  3377
$NEED   555#  3204
$PARAM  534#  3016
$PFAIL  534#  3336
$POKE   543#
$POKER  544#  2420  2493  2566  2639  2706  2761
$RCNET  555#
$RECAC  543#  2391  2464  2537  2610
$REG    534#  3071
$RESET  536#  1440  1469  1507  1536  1556  1576  1596  1616  1636  1656  1676  1696  1718  1740
        1760  1780  1826  1836  1851  1885  1914  1927  1938  1949  1960  1970  1981  1993  2004
        2015  2026  2037  2075  2088  2104  2343  2345  2370  2372  2406  2408  2479  2481  2552
        2554  2625  2627  2693  2695  2748  2750
$RXACT  542#  2337  2364
$SCOPE  532#  3482
$SCOPI  533#  2901
$SETFL  532#  3138
$SETVE  527#   678
$START  532#  1119
$STRIP  549#
$SYMBO  523#   746   939
$SYNCR  545#  2416  2489  2562  2635  2702  2757
$TRPAR  551#
$TSTNO  532#  1344  1360  1375  1390  1405  1420  1428  1457  1495  1524  1544  1564  1584  1604
        1624  1644  1664  1684  1706  1728  1748  1768  1788  1820  1834  1869  1897  1925  1936
        1947  1958  1968  1979  1991  2002  2013  2024  2034  2072  2086  2101  2341  2368  2404
        2477  2550  2623  2691  2746
$UNIBU  535#  1342  1358  1373  1388  1403
$VARIA  530#   688
$WOROF  546#
$WORDO  545#  2683  2738
$WOROP  547#
$$CMRE  813#   852   853   854   855   856   857
$$CMTM  813#   858   859   860   861   862   863
$$ESCA  678#
```

```
DZDUQ-A MACY11 27(1006)  03-FEB-77  07:44  PAGE 91
DZDUQA.M11    27-JAN-77 09:46          CROSS REFERENCE TABLE -- MACRO NAMES

SSNEWT    678#  1344  1360  1375  1390  1405  1420  1428  1457  1495  1524  1544  1564  1584  1604
         1624   1644  1664  1684  1706  1728  1748  1768  1788  1820  1834  1869  1897  1925  1936
         1947   1958  1968  1979  1991  2002  2013  2024  2034  2072  2086  2101  2341  2368  2404
         2477   2550  2623  2691  2746
SSSET    3583#  3592  3593  3594  3597  3598  3599  3600  3601  3602  3603  3604
SSSETN   1165#
SSSKIP    678#
.EQUAT    523#   568
.HEADE    523#
.SETUP    523#  1118
.SACTI    523#  1086
.SAPTB    523#   871#
.SAPTH    523#  1096
.SAPTY    523#   534
.SCATC    523#
.SCMTA    523#   813
.SEOP     523#
.SERRO    523#  3156
.SERRT    523#  3212
.SPOWE    523#
.SSCOP    523#  3482
.STRAP    523#  3560
.STYPE    523#  2909
.STYPO    523#  3259

. ABS.  017342     000

ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

DZDUQA,DZDUQA.SEQ/SOL/CRF+DZDUQ1/EQ:RUNA,DZDUQ2,DZDUQA
RUN-TIME: 19 14 1 SECONDS
RUN-TIME RATIO: 108/35=3.0
CORE USED:  31K  (62 PAGES)
```