DH11

DH11 DIAGNOSTIC

MD-11-DZDHM-C

EP-DZDHM-C-DL-B

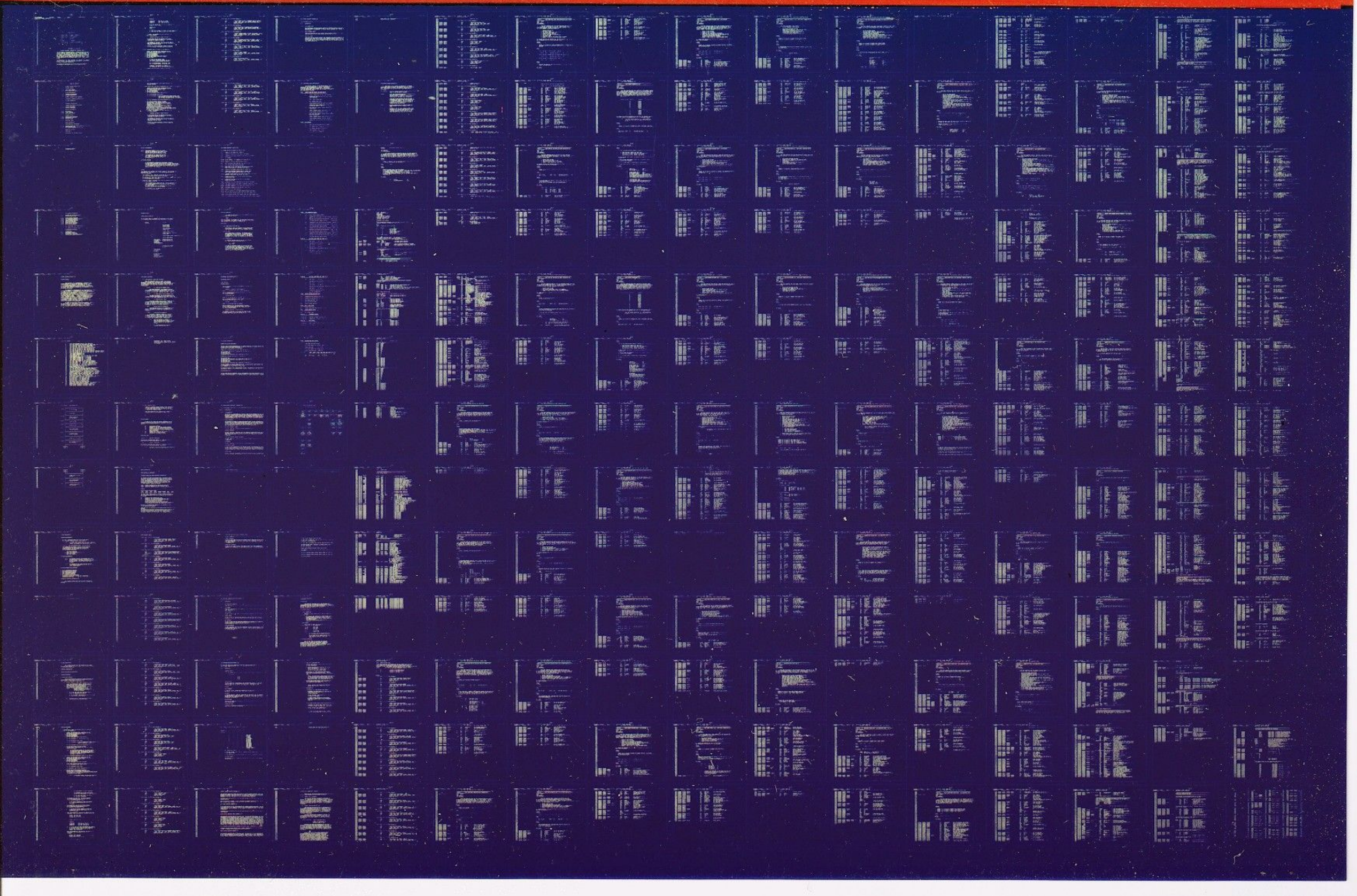
COPYRIGHT © 1977

FICHE 1 OF 2

APR 1977

digital

MADE IN U



DH11

DH11 DIAGNOSTIC

MD-11-DZDHM-C

EP-DZDHM-C-DL-B

COPYRIGHT © 1977

FICHE 2 OF 2

APR 1977

digital

MADE IN USA

```
                                        .REM `


                                                IDENTIFICATION

                                        PRODUCT CODE:   MAINDEC-11-DZDHM-C-D

                                        PRODUCT NAME:   DH11 DIAGNOSTIC

                                        DATE:           JAN 1977

                                        AUTHOR:         E. CROWLEY

                                        MAINTAINED BY:  DIAGNOSTIC ENGINEERING


                                                COPYRIGHT (C) 1976,1977

                                        DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

                                        THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
                                        SINGLE  COMPUTER  SYSTEM  AND  MAY  BE  COPIED ONLY WITH THE
                                        INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE,  OR
                                        ANY  OTHER  COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE
                                        MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR  USE  ON  SUCH
                                        SYSTEM  AND TO ONE WHO AGREES TO THESE LICENSE TERMS.  TITLE
                                        TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL  TIMES  REMAIN
                                        IN DEC.

                                        THE INFORMATION  IN  THIS  DOCUME T  IS  SUBJECT  TO  CHANGE
                                        WITHOUT  NOTICE  AND SHOULD NOT B  CONSTRUED AS A COMMITMENT
                                        BY DIGITAL EQUIPMENT CORPORATION.

                                        DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY  OF
                                        ITS SOFTWARE IN EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
```

TABLE OF CONTENTS

1.0 GENERAL PROGRAM DESCRIPTION

1.1 PROGRAM PURPOSE

1.1.1 LOGIC TEST SUMMARY
1.1.2 MD-11-DZDHM CORE MEMORY MAP

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS
1.2.2 SOFTWARE REQUIREMENTS

1.3 RELATED DOCUMENTS AND STANDARDS

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

1.5 FAILURE ASSUMPTIONS

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

2.1.1 LOADING PROCEDURES
2.1.2 STARTING PROCEDURES

2.2 SPECIAL ENVIRONMENTS

2.2.1 ACT11/APT11
2.2.2 "XXDP" SYSTEMS
2.2.3 SWITCHLESS FEATURE

2.3 PROGRAM OPTIONS

2.3.1 CONSOLE SWITCH REGISTER
2.3.2 CORE MEMORY LOCATIONS
2.3.3 REGISTER USAGE

2.4 EXECUTION TIMES

3.0 ERROR INFORMATION

3.1 ERROR REPORTING PROCEDURES

3.1.1 STANDARD SYSMAC.SML ERROR REPORTING CONVENTIONS
3.1.2 ERROR MESSAGE TABLE
3.1.3 DATA HEADER MNEUMONIC DEFINITIONS

3.2 POWER FAIL PRINTOUT

3.3 ERROR HALTS

4.0 PERFORMANCE AND PROGRESS REPORTS

```
112
113                                     4.1     PERFORMANCE REPORTS
114                                     4.2     PROGRESS REPORTS
```

5.0 DH11 DEVICE INFORMATION

5.1 ADDRESS AND VECTOR ASSIGNMENTS
5.2 REGISTER DEFINITIONS

5.2.1 SYSTEM CONTROL REGISTER
5.2.2 NEXT RECEIVED CHARACTER REGISTER
5.2.3 LINE PARAMETER REGISTER
5.2.4 CURRENT ADDRESS REGISTER
5.2.5 BYTE COUNT REGISTER
5.2.6 BUFFER ACTIVE REGISTER
5.2.7 BREAK CONTROL REGISTER
5.2.8 SILO STATUS REGISTER

5.3 DH11 FUNCTIONAL LOGIC PARTITIONING

5.4 DH11 MODULE ALLOCATION CHART

6.0 MAINTENANCE PROCEDURES

6.1 INTRODUCTION
6.2 PRELIMINARY CHECKS
6.3 MAINTENANCE CONNECTORS
6.4 COMPLETE DH11 SUB-SYSTEM CHECKOUT
6.5 MAINTENANCE HEADER DESCRIPTION

1.0 GENERAL PROGRAM DESCRIPTION

1.1 PROGRAM PURPOSE

"MD-11-DZDHM" IS A COMPREHENSIVE DIAGNOSTIC TEST PROGRAM DESIGNED TO AID IN THE ACCEPTANCE TESTING, INSTALLATION CHECKOUT, AND CORRECTIVE MAINTENANCE OF THE DH11 16. LINE ASYNCHRONOUS SERIAL LINE MULTIPLEXOR. IT CONSISTS OF 48. LOGICALLY SEQUENCED DIAGNOSTIC TESTS DESIGNED TO TEST AND VERIFY THAT THE DH11 IS OPERATING IN ACCORDANCE WITH ITS DESIGN SPECIFICATIONS.

THE PROGRAM IS CONFIGURABLE BY THE AUTOSIZER OR BY CONSOLE DIALOGUE TO ENABLE IT TO AUTOMATICALLY TEST AND VERIFY ALL 16. LINES ON UP TO 16. CONTIGUOUS DH11'S (WITH NON-CONTIGUOUS/ CONTIGUOUS VECTOR ASSIGNMENTS). INDIVIDUAL UNITS AND INDIVIDUAL LINES WITHIN A UNIT MAY BE SELECTED OR DESELECTED TO FACILITATE FAULT ISOLATION TO A PARTICULAR DH11 OR A FUNCTIONAL AREA OF LOGIC AFFECTING A PARTICULAR LINE WITHIN A UNIT. WHENEVER AN ERROR IS DETECTED A COMPREHENSIVE ERROR REPORT IS TYPED THAT ALLOWS THE USER TO ISOLATE THE FAULT TO A FUNCTIONAL AREA OF LOGIC. EXTENSIVE DOCUMENTATION IS PROVIDED TO PERMIT THE USER TO PROCEED FROM THE ERROR REPORT TO ADDITIONAL LOGIC CHECKS TO MAKE IN ORDER TO ISOLATE THE PROBLEM TO A REPLACEABLE UNIT.

IN ORDER TO FACILITATE INSTALLATION CHECKOUT, TESTS 101, AND 105 THROUGH 107 (TEST GROUP 1) OF THE MODEM CONTROL DIAGNOSTIC, DZDHK, HAVE BEEN INCLUDED IN THIS PROGRAM. IN THIS WAY ALL THE LEVEL CONVERTERS AND CABLES CAN BE CHECKED WITH JUST ONE PROGRAM USING THE H315 TURNAROUND CONNECTOR.

1.1.1 LOGIC TEST SUMMARY

T1	CHECK SSYN RESPONSE FROM ALL DH11 REGISTERS
T2	TEST THAT "MASTER CLR" CAN CLEAR THE "SCR","LPR","BKR",AND "SSR"
T3	TEST "SCR" REG R/W BITS CAN SET/CLR (NORMAL MODE)
T4	TEST "SCR" REG. READ ONLY BITS (NORMAL MODE)
T5	TEST "SCR" REG. BITS THAT CAN BE SET/CLR IN MAINT. MODE
T6	TEST THAT ALL R/W BITS IN "LPR" CAN BE SET/CLR
T7	TEST THAT ALL R/W BITS IN "BKR" CAN BE SET/CLR
T10	TEST THAT ALL R/W BITS IN "SSR" CAN BE SET/CLR
T11	TEST THAT CLR/SET OF BIT "N" IN "LPR" DOES NOT CLEAR ANY OTHER B
T12	TEST THAT CLR/SET OF BIT "N" IN "BKR" DOES NOT CLEAR ANY OTHER B
T13	TEST THAT CLR/SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER B
T14	"CAR" MEMORY ADDRESSING TEST
T15	"BCR" MEMORY ADDRESSING TEST
T16	"CAR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
T17	"BCR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
T20	"CAR" MEMORY PATTERNS TEST / 0'S DISTURB
T21	"BCR" MEMORY PATTERNS TEST / 0'S DISTURB
T22	"CAR" MEMORY PATTERNS TEST / 1'S DISTURB
T23	"BCR" MEMORY PATTERNS TEST / 1'S DISTURB
T24	TEST THAT "CAR" MEMORY EXT BITS SET/CLR PROPERLY
T25	TEST INTR. ENAB. BITS - INTR. CONDITION DISABLED
T26	TEST CHAR. AVAIL. I.E. WITH INTR. CONDITION ACTIVE
T27	TEST SILO OVFLW. I.E. WITH INTR. CONDITION ACTIVE
T30	TEST NON EX MEM I.E. WITH INTR. CONDITION ACTIVE
T31	TEST XMITTR DONE I.E. WITH INTR. CONDITION ACTIVE
T32	BASIC TRANSMITTER "NPR" LOGIC TEST 1
T33	TRANSMITTR NPR LOGIC TEST 2
T34	TEST THAT CHARACTER AVAILABLE CAN CAUSE RCVR INTERRUPT
T35	TEST THAT THE SILO STATUS REG COUNTS UP CORRECTLY
T36	TEST THAT SILO STATUS REGISTER DOWN COUNTS CORRECTLY
T37	TEST SILO ALARM LEVEL FOR COUNTS 0,1,2,4,8,16, AND 32
T40	TRANSMITTER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS
T41	RECEIVER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS
T42	VERIFY STORAGE OVERFLOW - NON MAINT MODE - ALL SELECTED LINES
T43	BASIC DATA TEST - ALL SELECTED LINES/ALL CHAR LENGTHS
T44	SINGLE LINE DATA TEST - ALL SELECTED LINES
T45	BASIC PARITY LOGIC TEST - ALL SELECTED LINES - ODD PARITY
T46	MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES
T47	AUTO ECHO TEST 1 - ALL SELECTED LINES
T50	AUTO ECHO TEST 2 - ALL SELECTED LINES
T51	AUTO ECHO TEST 3 - ALL SELECTED LINES
T52	BREAK BIT TEST - ALL SELECTED LINES
T53	HALF DUPLEX TEST - ALL SELECTED LINES
T54	VERIFY THAT OVERRUN CAN SET PROPERLY - ALL SELECTED LINES
T55	ABBREVIATED MODEM CONTROL DIAGNOSTIC. (DZDHK T101)
T56	MODEM CONTROL DIAGNOSTIC CONTINUED (DZDHK T105)
T57	MODEM CONTROL DIAGNOSTIC CONTINUED (DZDHK T106)
T60	MODEM CONTROL DIAGNOSTIC CONTINUED (DZDHK T107)

1.1.2 MD-11-DZDHM CORE MEMORY MAP

```
        ********************************
000000: *                              *
        *       VECTOR AREA            *
        *                              *
        ********************************
        *                              *
        *       STACK AREA             *
        *                              *
        ********************************
001100: *                              *
        *       SYSMAC CONSTANTS       *
        *       AND VARIABLES          *
        *                              *
        ********************************
BEGIN:  *                              *
        *       START-UP CODE          *
        *                              *
        ********************************
START1: *                              *
        *       START-UP CODE          *
        *                              *
        ********************************
TST1:   *                              *
        *       DH11 LOGIC TESTS       *
        *       TST1(8)-TST54(8)       *
        *                              *
        ********************************
$EOP:   *                              *
        *       STANDARD SYSMAC        *
        *       UTILITY ROUTINES       *
        *                              *
        ********************************
CKRST1: *                              *
        *       COMMON DH11 UTILITIES  *
        *                              *
        ********************************
DHADR:  *                              *
        *       DH11 PROGRAM CONSTANTS *
        *       AND VARIABLES          *
        *                              *
        ********************************
        *                              *
    *********                      *********
    * CONT. *                      * CONT. *
    *********                      *********
```

```
                                        *********                    *********
                                        * CONT. *                    * CONT. *
                                        *********                    *********
                                            *                            *
                                            ********************************
                                EM1:        *                              *
                                            *      SYSMAC ERROR MESSAGE    *
                                            *      BUFFERS                 *
                                            *                              *
                                            ********************************
                                TITLE:      *                              *
                                            *      DH11 MISCELLANEOUS      *
                                            *      MESSAGE BUFFERS         *
                                            *                              *
                                            ********************************
                                RBUF:       *                              *
                                            *      TRANSMIT AND RECEIVE    *
                                            *      DATA BUFFERS            *
                                            *                              *
                                            ********************************
```

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

A. ANY PDP11 COMPUTER SYSTEM WITH 12K OF CORE MEMORY
AND A CONSOLE TERMINAL DEVICE (VT50,LA36 ETC)

NOTE: FOR PAPER TAPE SYSTEMS USING THE PDP11 ABSOLUTE
LOADER, THE PROGRAM CAN LOAD AND RUN IN 8K OF CORE

B. A DH11 16. LINE ASYNCHRONOUS SERIAL LINE MULTIPLEXOR

C. TEST CONNECTORS AND MODULE (THE NO. OF EACH REQUIRED
IS DETERMINED BY THE PARTICULAR TEST APPLICATION.
REFER TO SECTION 6.3 FOR A COMPLETE DISCUSSION OF
THE MAINTENANCE CONNECTORS)

1. H315 TEST CONNECTOR
2. H8611 TEST CONNECTOR(FOR DH11-AD)
3. M974 TEST MODULE
4. H861 TEST CONNECTOR

1.2.2 SOFTWARE REQUIREMENTS

A. ACT11 APT11 THE PROGRAM CONTAINS THE REQUIRED ACT11/APT11
SOFTWARE HOOKS TO PROPERLY INTERFACE WITH THE
ACT/APT SYSTEMS. THE PROGRAM CONTAINS AN AUTOSIZER
AND CAN BE RUN IN QUICK VERIFY MODE USING
"CHAINS".

B. XXDP THE PROGRAM MAY BE LOADED AND RUN FROM ANY
"XXDP" MEDIUM PROVIDED THE SYSTEM HAS AT LEAST
12K OF CORE STORAGE.

1.3 RELATED DOCUMENTS AND STANDARDS

A. DH11-D ENGINEERING DRAWINGS
B. DH11 MANUAL EK-DH11-MM-002
C. PDP11 PERIPHERALS HANDBOOK
D. PDP11 PROCESSOR HANDBOOK
E. MD-11-DZQAC-C1 SYSMAC.SML
F. MD-11-DZQXA "XXDP" USER'S GUIDE
G. DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS
PROGRAMMING PRACTICES DOC NO. 175-003-009-00

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

MD-11-DZDHM ASSUMES THAT THE FOLLOWING DIAGNOSTICS
HAVE BEEN RUN PRIOR TO ITS EXECUTION AND THAT NO ERRORS WERE
DETECTED:

A. CPU/CORE MEMORY DIAGNOSTICS

```
                                1.5     FAILURE ASSUMPTIONS
                                        --------------------

                                                MD-11-DZDHM ASSUMES THAT THE PROGRAM CAN BE LOADED
                                        INTO CORE AND STARTED. IT ALSO ASSUMES THE CPU/MEMORY HARDWARE
                                        IS FUNCTIONING ERROR FREE.

                                2.0     OPERATING INSTRUCTIONS
                                        -----------------------

                                2.1     LOADING AND STARTING PROCEDURES
                                        --------------------------------

                                2.1.1   LOADING PROCEDURES
                                        -------------------

                                        A. PAPER TAPE SYSTEMS

                                                USE THE STANDARD PDP11 ABSOLUTE LOADER PROCEDURE FOR
                                        LOADING PAPER TAPES. AFTER LOADING THE PROGRAM MUST BE MAN-
                                        UALLY STARTED AS DESCRIBED IN SECTION 2.1.2.

                                        B. "XXDP" SYSTEMS (REFER TO "XXDP" USER'S GUIDE MD-11-DZQXA)

                                                1. MOUNT THE APPROPRIATE MEDIUM (DECTAPE,DISK ETC)
                                                   CONTAINING THE "XXDP" MONITOR AND MD-11-DZDHM.
                                                2. BOOT THE SYSTEM TO LOAD THE MONITOR
                                                3. ONCE LOADED THE "XXDP" MONITOR PRINTS AN INTRO-
                                                   DUCTORY MESSAGE AND RESPONDS WITH A ".".
                                                4. TYPE:        "DZDHMC" FOLLOWED BY EITHER A <CR>
                                                                CARRIAGE RETURN OR AN "ALTMODE"
                                                                TO LOAD THE PROGRAM.

                                                        IF A <CR> WAS TYPED THE USER MUST MANUALLY
                                                        START THE PROGRAM AFTER LOADING.

                                                        IF THE "ALTMODE" TERMINATOR WAS USED THE PROGRAM
                                                        WILL SELF START AFTER LOADING.


                                                NOTE:   WHENEVER THE DH11 CONFIGURATION IS CHANGED
                                                        THE DIAGNOSTIC SHOULD BE RELOADED.
```

2.1.2 STARTING PROCEDURES

A. TO AUTOMATICALLY START THE PROGRAM USING THE AUTOSIZER
(START AT LOC 000200(8))

1. INSTALL THE REQUIRED TEST CONNECTORS FOR THE PARTICULAR TEST APPLICATION (REFER TO SECTION 6.3)
2. SET THE HALT/ENABLE SWITCH TO HALT
3. SET THE SR=000200(8)
4. DEPRESS LOAD ADDRESS
5. SET THE SR=000000 (WORST CASE TESTING)

   SET THE SR=000002 (TO TYPE THE DEVICE MAP)

   SET THE SR=004000 (QUICK PASS)

   SET THE SR=002000 (TO SKIP AN ABBREVIATED MODEM CONTROL TEST. REFER TO SECTIONS 1.1 AND 6.3)

   SET THE SR=000400 (HALT AFTER PARAMETER SET-UP)

6. SET THE HALT/ENABLE SWITCH TO ENABLE
7. DEPRESS START - THE PROGRAM WILL TEST ALL LINES ON ALL DH'S FOUND.

   NOTE: THE CSR REGISTER ADDRESS OF THE MODEM CONTROL('S) IS LOADED ONLY FROM THE AUTOSIZER, HOWEVER, AFTER INITIAL LOAD, THE PROGRAM CAN BE STARTED AT 210(8) TO CHANGE SELECTION PARAMETERS AS DESCRIBED IN SECTION 2.1.2 D.

B. TO TYPE IN ALL REQUIRED PARAMETERS (START AT LOC 000200(8))

1. INSTALL THE REQUIRED TEST CONNECTORS FOR THE PARTICULAR TEST APPLICATION (REFER TO SECTION 6.3)
2. SET THE HALT/ENABLE SWITCH TO HALT
3. SET THE SR=000200(8)
4. DEPRESS LOAD ADDRESS
5. SET THE SR=000001 (FOR INPUT DIALOGUE)

   AFTER INPUT DIALOGUE BEGINS BUT PRIOR TO ACTUAL TESTING:
   SET THE SR=000000 (WORST CASE TESTING)

   SET THE SR=004000 (QUICK PASS)

   SET SR=000400 (HALT AFTER PARAMETER SET-UP)

6. SET THE HALT/ENABLE SWITCH TO ENABLE
7. DEPRESS START - THE PROGRAM TYPES THE TITLE AND THEN ASKS FOR THE NUMBER OF ADDRESSES BETWEEN VECTORS. TYPE EITHER 10(8) OR 20(8) DEPENDING UPON THE PARTICULAR CONFIGURATION TO BE TESTED:

   NOTES: IF THE MODEM CONTROL VECTORS ARE INTERLEAVED WITH THE DH11 VECTORS (2040 FRONT END)

```
                                                THE DISPLACEMENT IS 20(8) ADDRESSES.

                                                FOR STANDARD DH11'S WITH CONTIGUOUS
                                                VECTORS THE DISPLACEMENT IS 10(8) ADDRESSES.

                                                IF <CR> ONLY WAS TYPED, THE DEFAULT
                                                WILL BE 20(8) ADDRESSES.

                                        8. THE PROGRAM WILL ASK FOR  THE DEVICE ADDRESS.
                                           TYPE IN THE ADDRESS (OCTAL) OF THE FIRST DH11
                                           IN THE SYSTEM FOLLOWED BY A <CR>.

                                                IF AN INVALID ADDRESS IS TYPED THE PROGRAM
                                                WILL TYPE AN ERROR MESSAGE AND ASK YOU TO
                                                TRY AGAIN.

                                        9. THE PROGRAM WILL ASK FOR THE VECTOR ADDRESS.
                                           TYPE IN RECEIVER VECTOR ADDRESS (OCTAL) OF
                                           THE FIRST DH11 FOLLOWED BY A <CR>.

                                                IF AN INVALID VECTOR ADDRESS IS TYPED THE
                                                PROGRAM WILL TYPE AN ERROR MESSAGE AND ASK
                                                YOU TO TRY AGAIN.

                                        10. NEXT THE PROGRAM WILL ASK FOR THE DEVICE SELECTION
                                            PARAMETER. TYPE IN AN OCTAL NO. ENCODED AS FOLLOWS:

                                                BIT00=1 TEST DH11 #00
                                                BIT01=1 TEST DH11 #01
                                                BIT02=0 DO NOT TEST DH11 #02
                                                  "
                                                  "
                                                BIT15=1 TEST DH11 #15

                                                EXAMPLES:

                                                177777<CR>      TEST ALL 16. DH11'S
                                                100000<CR>      TEST ONLY DH11 #17(8)
                                                000005<CR>      TEST DH11 #00 AND 02

                                                IF A <CR> ONLY IS TYPED THE PROGRAM WILL DEFAULT
                                                TO THE LAST TYPED IN DEVICE SELECT PARAMETER. IF
                                                THIS IS THE INITIAL LOAD IT WILL DEFAULT TO
                                                000003  (DH11 #00 AND 01)

                                        11. NEXT THE PROGRAM WILL ASK FOR THE LINE SELECTION
                                            PARAMETERS. TYPE AN ENCODED OCTAL NO. AS
                                            FOLLOWS:

                                                BIT00=1 TEST LINE #00
                                                BIT01=1 TEST LINE #01
                                                BIT02=0 DO NOT TEST LINE #02
                                                  "
                                                  "
                                                BIT15=1 TEST LINE #15
```

```
                                        EXAMPLES:

                                        177777<CR>      TEST ALL 16. LINES
                                        100000<CR>      TEST LINE 17(8) ONLY
                                        000005<CR>      TEST LINES 00 AND 02

                                        IF A <CR> RETURN ONLY IS TYPED THE PROGRAM WILL
                                        DEFAULT TO 16. LINES.

                                        ***********
                                        NOTE
                                        **********

                                        IF MORE THAN ONE DH11 IS TESTED THE SAME COMBINATION
                                        OF LINES WILL BE TESTED ON ALL DH11'S SELECTED.


                                12. IF SR8=1, THE PROGRAM WILL HALT AND PRINT THE
                                    FOLLOWING MESSAGE:

                                        "DEPRESS CONTINUE TO START TESTING"

                                    AT THIS POINT SET UP THE DESIRED SWITCH REG-
                                    ISTER OPTIONS (REFER TO PARA 2.3.1) AND DEPRESS
                                    "CONTINUE" TO START THE TESTING.

                                    THE PURPOSE OF THIS HALT IS TO ALLOW THE USER TO
                                    DUMP THE PROGRAM AFTER SETTING UP THE CONFIGURATION
                                    PARAMETERS FOR HIS SYSTEM.

                                13. PROGRAM WILL BEGIN EXECUTION. REFER TO SECTIONS
                                    2.4, 3.0, AND 4.0 FOR ERROR AND STATUS REPORTS.


                        C. DEFAULT PARAMETER START ** (START AT LOC 000204(8))

                                1. INSTALL THE REQUIRED TEST CONNECTORS FOR THE
                                   PARTICULAR TEST APPLICATION (REFER TO SECTION 6.3)
                                2. SET THE HALT/ENABLE SWITCH TO HALT
                                3. SET THE SR=000204(8)
                                4. DEPRESS LOAD ADDRESS
                                5. SET THE SR=000000 (WORST CASE TESTING)
                                6. SET THE HALT/ENABLE SWITCH TO ENABLE
                                7. DEPRESS START

                                **      IF THIS IS THE INITIAL LOAD,
                                        THE DEFAULT PARAMETERS ASSUME TWO DH11'S
                                        WITH THE FOLLOWING ADDRESS ASSIGNMENTS

                                        DH11 #0 DEVADR=760020, VECTOR=330, BR5
                                        DH11 #1 DEVADR=760040, VECTOR=350, BR5

                                        OTHERWISE, THE PROGRAM WILL DEFAULT TO
                                        THE PARAMETERS USED IN THE PREVIOUS EXECUTION.

                                8. PROGRAM EXECUTION BEGINS. REFER TO SECTIONS 2.4, 3.0,
```

```
                                        AND 4.0 FOR EXECUTION TIMES, ERROR REPORTS, AND
                                        PROGRESS REPORTS.

                                D. TO CHANGE DEVICE AND LINE SELECT PARAMETERS ONLY (START AT LOC 000210(8))

                                      1. INSTALL THE REQUIRED TEST CONNECTORS FOR THE
                                         PARTICULAR TEST APPLICATION (REFER TO SECTION 6.3)
                                      2. SET THE HALT/ENABLE SWITCH TO HALT
                                      3. SET THE SR=000210(8)
                                      4. DEPRESS LOAD ADDRESS
                                      5. SET THE SR=000000 (WORST CASE TESTING)

                                         SET THE SP=004000 (QUICK PASS)

                                         SET THE SR=002000 (TO SKIP AN ABBREVIATED MODEM CONTROL TEST.
                                                            THIS ASSUMES THE AUTOSIZER WAS PREVIOUSLY
                                                            USED TO LOAD THE MODEM CONTROL CSR ADDRESSES.)

                                         SET THE  SR=000400 (HALT AFTER PARAMETER SET-UP)

                                      6. SET THE HALT/ENABLE SWITCH TO ENABLE
                                      7. DEPRESS START - THE PROGRAM TYPES THE TITLE
                                         AND THEN  ASKS FOR DEVICE SELECTION PARAMETER.
                                         PROCEED AS IN (B-10) ABOVE.
                                      9. PROGRAM WILL ASK FOR LINE SELECTION PARAMETERS.
                                         PROCEED AS IN (B-11) ABOVE.
                                        NOTE: THE DEVICE SELECTION AND LINE SELECTION
                                        PARAMETERS APPLY TO BOTH THE DH11 AND THE MODEM CONTROL,
                                        THAT IS, IF DH #7, LINE #3 IS CHOSEN THEN MODEM CONTROL
                                        #7 LINE #3 WILL ALSO BE TESTED.
                                      10. IF SR8=1, THE PROGRAM WILL HALT AND PRINT THE
                                          FOLLOWING MESSAGE:

                                              "DEPRESS CONTINUE TO START TESTING"

                                          AT THIS POINT SET UP THE DESIRED SWITCH REG-
                                          ISTER OPTIONS (REFER TO PARA 2.3.1) AND DEPRESS
                                          "CONTINUE" TO START THE TESTING.

                                          THE PURPOSE OF THIS HALT IS TO ALLOW THE USER TO
                                          DUMP THE PROGRAM AFTER SETTING UP THE CONFIGURATION
                                          PARAMETERS FOR HIS SYSTEM.

                                      11. PROGRAM WILL BEGIN EXECUTION. REFER TO SECTIONS
                                          2.4, 3.0, AND 4.0 FOR EXECUTION TIMES  ERROR AND
                                          STATUS REPORTS.
```

2.2 SPECIAL ENVIRONMENTS

2.2.1 ACT11/APT11

WHEN UNDER CONTROL OF THE ACT11/APT11 SYSTEMS THE PROGRAM MAY BE LOADED IN DUMP MODE AND CAN BE RUN AS PART OF A QUICK VERIFY CHAIN SINCE AN AUTOSIZER IS USED.

2.2.2 XXDP

THE PROGRAM MAY BE LOADED AND RUN FROM ANY "XXDP" MEDIUM PROVIDED THERE IS AT LEAST 12K OF CORE. IT MAY BE RUN AS PART OF AN "XXDP" CHAIN.

2.2.3 SWITCHLESS FEATURE

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

CONTROL:

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

1) TYPE CONTROL G <↑G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)

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

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

   B) IF A CONTROL U <↑U> IS DEPRESSED THEN THE PROGRAM WILL DO A <CR>. RETYPE THE DESIRED NUMBER.

2.3 PROGRAM OPTIONS

2.3.1 CONSOLE SWITCH REGISTER

THE FOLLOWING TABLE ILLUSTRATES THE FUNCTIONS OF THE CONSOLE SWITCH REGISTER DURING PROGRAM START AND DURING DH TESTING:

SWITCH REGISTER	START	TESTING
15 = 1	-----	HALT ON ERROR (AFTER TYPING ERROR MESSAGE)
14 = 1	-----	LOOP CONTINUOUSLY ON CURRENT TEST.
13 = 1	-----	INHIBIT ERROR TYPOUTS.
11 = 1	-----	INHIBIT SUB-TEST ITERATIONS (QUICK PASS)
10 = 1	-----	INHIBIT MODEM CONTROL ABBREVIATED TESTS.
9 = 1	-----	LOCK ON HARD ERRORS
8 = 1	HALTS AFTER CONFIGURATION TO PERMIT DUMPING PRE-CONFIGURED COPIES OF THE PROGRAM.	SEARCH FOR AND LOCK ON TEST SELECTED BY CONTENTS OF SR <07:00>
<07:00>		CONTAINS TEST NUMBER TO SEARCH FOR WHEN SR 08 = 1
1 = 1	TYPES DEVICE MAP GENERATED BY THE AUTOSIZER.	-----
0 = 1	ALLOWS THE USER TO INPUT DH PARAMETERS MANUALLY. (INHIBITS THE AUTOSIZER)	-----

2.3.2 CORE MEMORY LOCATIONS

A. DH11 CONFIGURATION TABLES AND VARIABLES

WHEN THE AUTOSIZER OPTION IS USED, THIS PROGRAM CAN RUN NON-STANDARD DH11 CONFIGURATIONS (NON-CONTIGUOUS ADDRESSES). THE USER CAN ALSO PATCH IN HIS OWN ADDRESSES TO MATCH HIS CONFIGURATION AND THEN USE THE DEFAULT START TO RUN THE UPDATED PROGRAM. THE TABLES AND LOCATIONS TO MODIFY ARE DES-CRIBED BELOW:

1. DHADTB: 16. WORD DEVICE ADDRESS TABLE

THE USER CAN DEPOSIT THE ADDRESSES FOR HIS NON-STANDARD CONFIGURATION IN THIS TABLE. THE POSITION OF THE ENTRY IN THE TABLE CORRESPONDS DIRECTLY TO THE DEVICE NO. (IE DH11 #00 - WORD 00, DH11 #01 - WORD 01 ETC.)

2. DHVCTB: 16. WORD DEVICE VECTOR ADDRESS TABLE

THE USER CAN DEPOSIT THE VECTOR ADDRESSES FOR HIS NON-STANDARD CONFIGURATION IN THIS TABLE. AGAIN THE POSITION IN THE TABLE CORRESPONDS DIRECTLY TO DEVICE NUMBER.

3. BRLVL: 16. WORD BR LEVEL TABLE

THIS TABLE STORES THE BR LEVELS ASSUMED BY THE INTERRUPT SERVICE ROUTINES FOR EACH DH11. THE RCVR BR LEVEL IS STORED IN THE LOW BYTE AND THE XMITTER BR LEVEL IN THE HIGH BYTE. AGAIN THE POSITION IN THE TABLE CORRESPONDS DIRECTLY TO THE DH11 DEVICE NO.

4. DHSEL: DEVICE SELECTION PARAMETER

THIS WORD MUST BE SET UP TO CORRESPOND TO THE DEFAULT CONFIGURATION DEFINED BY THE TABLE SET-UPS. REFER TO SECTION 2.1.2.(B10) FOR A DESCRIPTION OF ITS ENCODING.

5. LINSEL: LINE SELECTION PARAMETER

THIS WORD IS PROGRAM LOADED AS A 177777(8) TO SPECIFY THAT ALL LINES (16.) ARE TO BE TESTED. IT MAY BE MODIFIED AT CONFIGURATION TIME TO SPECIFY ANY COMBINATION OF LINES TO TEST. REFER TO SECTION 2.1.3.(B11) FOR A DESCRIPTION OF ITS ENCODING.

NOTE: ONCE THE PROGRAM IS STARTED IT IS TABLE DRIVEN AND USES "DHASEL", "LINSEL" AND THE CONTENTS OF THE THREE TABLES ABOVE TO DEFINE THE CONFIGURATION TO TEST.

NOTE: IT IS RECOMMENDED THAT WHEN NON-STANDARD CONFIGURATIONS ARE

```
ENCOUNTERED, THE  MODEM CONTROL
DIAGNOSTIC, DZDHK, SHOULD BE RUN, RATHER
THAN ALTERING THE MODEM CONTROL TABLES IN
THIS PROGRAM.
```

B. SUB-TEST ITERATION COUNT

THERE IS A LOCATION TAGGED "$MXCNT:" THAT DETERMINES HOW MANY TIMES EACH SUB-TEST IS REPEATED (SR11=0) IT IS PROGRAM LOADED TO 000010(8) BUT CAN BE CHANGED TO MODIFY THE ITERATION COUNT.

NOTE THAT MODIFYING THIS LOCATION WILL CHANGE THE PROGRAM EXECUTION TIME DEFINED IN PARA 2.4(B).

2.3.3 REGISTER USAGE

IN MOST OF THE TESTS THE GENERAL REGISTERS CONTAIN STANDARD INFORMATION AS SHOWN BELOW. ON PROGRAM HALTS THE REGISTERS CAN BE EXAMINED DIRECTLY TO DISPLAY THIS INFORMATION.

R0	TEST NUMBER IN OCTAL
R1	ADDRESS OF THE "SCR" REG (DEVICE ADDRESS)
R2	ADDRESS OF THE DH11 REGISTER BEING TESTED
R3	ACTUAL CONTENTS OF THE DH11 REG BEING TESTED
R4	WHAT THE CONTENTS OF THE DH11 REG BEING TESTED SHOULD HAVE BEEN
R5	GENERAL USE - REFER TO THE LISTING FOR ITS USE
R6	CONTENTS OF THE STACK POINTER
R7	CONTENTS OF THE PROGRAM COUNTER

2.4 EXECUTION TIMES

A. SR11 = 0 SUB-TEST ITERATIONS

WITH ONE DH11 SELECTED FOR TESTING 16. LINES ONE COMPLETE ERROR FREE PASS TAKES APPROXIMATELY 8 MINUTES.

B. SR11 = 1 INHIBIT ITERATIONS

WITH ONE DH11 SELECTED FOR TESTING 16. LINES ONE COMPLETE ERROR FREE PASS TAKES APPROXIMATELY ONE MINUTE

NOTE: THE ABOVE TIMES WERE DETERMINED WHEN THE PROGRAM WAS RUN ON A PDP-11/45 AND A PDP-11/40 CPU.

3.0 ERROR INFORMATION

3.1 ERROR REPORTING PROCEDURES

3.1.1 STANDARD SYSMAC.SML ERROR REPORTING CONVENTIONS

THE PROGRAM UTILIZES THE STANDARD PDP11 DIAGNOSTICS ERROR UTILITIES. THE TEST ROUTINE CALLS THESE UTILITIES USING AN "ERROR N" INSTRUCTION (CODED EMT) WHERE "N" IS THE NUMBER OF THE ERROR MESSAGE. THE UTILITY ROUTINE USES "N" TO ACCESS THE PROPER ERROR INFORMATION VIA THE ERROR TABLE DESCRIBED IN SECTION 3.1.2 BELOW. EACH MESSAGE RESULTS IN THREE LINES OF TYPEOUT AS FOLLOWS:

LINE 1 A BRIEF DESCRIPTION OF THE FAILING FUNCTION
LINE 2 LABELS TO IDENTIFY THE DATA TYPED ON LINE 3
LINE 3 THE ACTUAL ERROR DATA (UP TO 8 OCTAL OR DECIMAL NOS.)

EXAMPLE:

SYSTEM CONTROL REGISTER ERROR

(PC)	(PS)	(SP)	TEST	DEVADR	REGADR	WAS	S/B
002720	000002	001074	000003	160020	160020	000000	000001

THE ERROR TABLE ITEMS SHOWN IN THE NEXT SECTION DESCRIBE ALL THE DH ERROR MESSAGES WITHIN MD-11-DZDHM AND ARE INTERPRETED AS FOLLOWS:

EM ADDRESS OF THE MESSAGE FOR LINE 1
DH ADDRESS OF THE DATA HEADER MESSAGE FOR LINE 2
DT ADDRESS OF THE TABLE OF ADDRESSES THAT POINT TO THE DATA WORDS TO BE PRINTED
DF ADDRESS THAT POINTS TO THE DATA DESCRIPTOR TABLE THAT DEFINES WETHER AN ITEM IS OCTAL OR DECIMAL. IF THIS ENTRY IS "0" ALL DATA WORDS ARE IN OCTAL.

SECTION 3.1.3 DEFINES THE MEANING OF THE MNEUMONICS USED IN THE VARIOUS DATA HEADERS.

THERE ARE ONLY TWO MESSAGES IN THE MODEM CONTROL PORTION OF THIS PROGRAM:
ONE INFORMS THE USER THAT NO MODEM CONTROL'S WERE FOUND BY THE AUTOSIZER AND THE PROGRAM THEN CONTINUES TESTING THE DH11'S.
THE OTHER INSTRUCTS THE USER TO RUN THE MODEM CONTROL DIAGNOSTIC, DZDHK, DUE TO AN ERROR. THE PROGRAM THEN CONTINUES.

```
                        3.1.2   ERROR MESSAGE TABLES
                                ---------------------

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 1

                                        EM1             ;"DH11 REGISTER REFERENCE CAUSED TIMEOUT"
                                        DH1             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR "
                                        DT1             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2
                                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 2

                                        EM2             ;"SYSTEM CONTROL REGISTER ERROR"
                                        DH2             ; (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
                                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 3

                                        EM3             ;"DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG"
                                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
                                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 4

                                        EM4             ;"LINE PARAMETER REGISTER ERROR"
                                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
                                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 5

                                        EM5             ;"BREAK CONTROL REGISTER ERROR"
                                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
                                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 6

                                        EM6             ;"SILO STATUS REGISTER ERROR"
                                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
                                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 7

                                        EM7             ;"CURRENT ADDRESS REGISTER ERROR - LINE #XX"
                                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
                                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                        0               ;PRINT ALL OCTAL
```

```
;ERROR TABLE ITEM FOR ERROR MESSAGE 10

        EM10            ;"BYTE COUNTER REGISTER ERROR - LINE #XX"
        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 11

        EM11            ;"UNEXPECTED DH11 RCVR INTERRUPT"
        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 12

        EM12            ;"UNEXPECTED DH11 XMITTR INTERRUPT"
        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 13

        EM13            ;"CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT"
        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 14

        EM14            ;"TRANSMITTER NPR LOGIC ERROR - LINE #  "
        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 15

        EM15            ;"XMITTR FAILED TO INTERRUPT - LINE #  "
        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 16

        EM16            ;"RCVR FAILED TO INTERRUPT"
        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL
```

```
                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 17

                                                EM17            ;"TRANSMITTER TIMING ERROR - LINE #  "
                                                DH6             ;" (PC)  (PS)  (SP)  TEST  DEVADR  SPEED   TIMEB   TIMEC
                                                DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                                0               ;PRINT ALL OCTAL

                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 20

                                                EM20            ;RECEIVER TIMING ERROR - LINE #  "
                                                DH6             ;" (PC)  (PS)  (SP)  TEST  DEVADR  SPEED   TIMEB   TIMEC
                                                DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                                0               ;PRINT ALL OCTAL

                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 21

                                                EM21            ;"RCVR FAILED TO INTERRUPT - LINE #  "
                                                DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR   9S  S/B "
                                                DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                                0               ;PRINT ALL OCTAL

                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 22

                                                EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
                                                DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
                                                DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                                0               ;PRINT ALL OCTAL

                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 23

                                                EM23            ;"BASIC DATA TEST ERROR - LINE #  "
                                                DH7             ;" (PC)  (PS)  (SP)  TEST  DEVADR  CHRLNG  WAS  S/B "
                                                DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                                0               ;PRINT ALL OCTAL


                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 24

                                                EM24            ;"AUTO ECHO TEST ERROR - LINE #  "
                                                DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
                                                DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                                0               ;PRINT ALL OCTAL

                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 25

                                                EM25            ;"BREAK BIT TEST ERROR - LINE #  "
                                                DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
                                                DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                                0               ;PRINT ALL OCTAL
```

```
                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 26

                                EM26            ;"HALF-DUPLEX TEST ERROR - LINE #  "
                                DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
                                DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 27

                                EM27            ;"UNEXPECTED BUS ERROR TRAP"
                                DH3             ; (PC)  (PS)  (SP)  TEST  TRPPC  TRPPS
                                DT3             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 30

                                EM30            ;"UNEXPECTED RSVD INSTR TRAP"
                                DH3             ; (PC)  (PS)  (SP)  TEST  TRPPC  TRPPS
                                DT3             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 31

                                EM31            ;"AUTO ECHO DATA COMPARE ERROR - LINE #  "
                                DH4             ; (PC)   (PS)   (SP)    TEST   WASADR  SBADR   WAS    S/
                                DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 32

                                EM32            ;"AUTO ECHO TEST TIMEOUT - LINE #  "
                                DH5             ;" (PC)  (LPRG)   TEST""
                                DT4             ;$ERRPC,$TMPO,$TMP2
                                0               ;PRINT ALL OCTAL


                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 33

                                EM33            ;"PARITY LOGIC TEST ERROR - LINE #  "
                                DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B"
                                DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 34

                                EM34            ;"MULTI-LINE PARITY DATA TEST ERROR - LINE #   - SUBTEST
                                DH4             ;" (PC)  (PS)  (  )  TEST  WASADR  SBADR  WAS  S/B "
                                DT2             ;$ERRPC,$TMPO,$.  .6,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL
```

```
;ERROR TABLE ITEM FOR ERROR MESSAGE 35

        EM35            ;"MULTI-LINE PARITY DATA TEST TIMEOUT"
        DH14            ;" (PC)  (LPRG)  LINACT "
        DT6             ;"$ERRPC,$TMP0,$TMP3"
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 36

        EM36            ;CHAR AVAILABLE TIMEOUT"
        DH5             ;" (PC)  (LPRG)   TEST"
        DT4             ;$ERRPC,$TMP0,$TMP2
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 37

        EM37            ;"DATA COMPARE ERROR - LINE #  "
        DH4             ;" (PC)  (PS)  (SP)  TEST  WASADR  SBADR  WAS  S/B "
        DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 40

        EM40            ;"BUFFER ACTIVE REG ERROR - LINE #  "
        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
        DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 41

        EM41            ;"RCVR FALSE INTERRUPT"
        DH5             ;" (PC)  (LPRG)   TEST"
        DT4             ;$ERRPC,$TMP0,$TMP2
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 42

        EM42            ;"SILO OVERFLOW ERROR"
        DH5             ;" (PC)  (LPRG)   TEST"
        DT4             ;$ERRPC,$TMP0,$TMP2
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 43

        EM43            ;"SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT"
        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
        DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL
```

```
                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 44

                                EM44            ;"NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT"
                                DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
                                DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 45

                                EM45            ;"XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT"
                                DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
                                DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 46

                                EM46            ;"CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE #  "
                                DH10            ;" (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS
                                DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 47

                                EM47            ;"BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE #  "
                                DH10            ; (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS
                                DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 50

                                EM50            ;"TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  '
                                DH2             ;" (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS
                                DT2             ;"$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 51

                                EM51            ;"NPR LOGIC TEST 2 ERROR"
                                DH11            ;" (PC)   LINACT  LINCHK   TEST   DEVADR  REGADR   WAS
                                DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4"
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 52

                                EM52            ;"BASIC DATA COMPARE ERROR"
                                DH2             ;" (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS
                                DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
                                0               ;PRINT ALL OCTAL
```

```
                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 53

                                EM50            ;"TEST TIMEOUT WAITING FOR XMIT DONE - LINE # "
                                DH12            ;" (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS
                                DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 54

                                EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE # "
                                DH12            ;" (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS
                                DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 55

                                EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE # "
                                DH13            ;" (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG  SCRWAS
                                DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 56

                                EM56            ;"OVERRUN BIT FAILED TO SET - LINE # "
                                DH2             ; (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS
                                DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 57

                                EM57            ;"STORAGE OVERFLOW BIT FAILED - LINE # "
                                DH2             ; (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS
                                DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
                                0               ;PRINT ALL OCTAL
```

3.1.3 DATA HEADER MNEUMONIC DEFINITIONS

ALL NUMBERS PRINTED AS ERROR DATA ARE IN OCTAL

(PC)	ADDRESS OF THE ERROR CALL (ERROR PC)
(PS)	CONTENTS OF THE PSW AT THE TIME OF THE ERROR
(SP)	CONTENTS OF THE STACK POINTER AT THE TIME OF THE ERROR
TEST	TEST NUMBER
DEVADR	DEVICE ADDRESS - 1ST ADDRESS IN THE SELECTED DH11
REGADR	ADDRESS OF THE DH11 REGISTER BEING TESTED
WAS	WHAT THE ACTUAL DATA READ WAS (DH11 REG OR CORE LOC.)
S/B	WHAT THE DATA READ SHOULD HAVE BEEN
SPEED	SPEED CODE IN THE "LPR" REG AT THE TIME OF THE ERROR REFER TO SECTION 5.2.3 FOR SPEED CODE TABLES
TIMEB	CONTENTS OF SOFTWARE COUNTER USED IN TIMING TESTS
TIMEC	CONTENTS OF SOFTWARE COUNTER USED IN TIMING TESTS NOTE:"TIMEB" SHOULD ALWAYS BE LESS THAN "TIMEC"
CHRLNG	CHARACTER LENGTH CODE IN THE "LPR" AT THE TIME OF THE ERROR 00=5 BITS, 01=6 BITS, 02=7 BITS, 03= 8 BITS
TRPPC	CONTENTS OF THE PC (R7) AT THE TIME OF A BUS ERROR OR RSVD INSTR TRAP.
TRPPS	CONTENTS OF THE PSW AT THE TIME OF A BUS ERROR OR RSVD INSTR TRAP.
(LPRG)	CONTENTS OF THE "LPR" REGISTER AT THE TIME OF THE ERROR
LINACT	FLAGS USED BY MULTI-LINE TESTS TO INDICATE LINES STILL ACTIVE
WASADR	CORE MEMORY ADDRESS OF THE "WAS" DATA (ACTUAL DATA READ)
SBADR	CORE MEMORY ADDRESS OF THE S/B DATA (GOOD DATA)
SCRWAS	CONTENTS OF THE "SCR" REGISTER
SCRS/B	WHAT THE CONTENTS OF THE "SCR" REGISTER SHOULD HAVE BEEN
LINCHK	LINE NO. BEING CHECKED DURING "CAR" AND "BCR" MEMORY TESTS
LINEWR	LINE NO. BEING WRITTEN INTO DURING "CAR" AND "BCR" MEMORY TESTS
PATTRN	TEST PATTERN BEING WRITTEN INTO EITHER THE "CAR" OR "BCR" MEMORIES

3.2 POWER FAIL PRINTOUT

IF A POWER FAILURE OCCURS WHILE THE PROGRAM IS RUNNING, THE FOLLOWING PRINTOUT OCCURS:

"POWER"

AFTER THE PRINTOUT THE PROGRAM WILL BE RESTARTED AUTOMATICALLY FROM THE BEGINNING. NO ATTEMPT IS MADE TO CONTINUE THE PROGRAM FROM THE POINT OF THE POWER FAIL INTERRUPTION.

3.3 ERROR HALTS

A. SYSMAC ERROR SERVICE ROUTINE HALT

WHEN SR15=1 A "HALT" IS EXECUTED IN THE SYSMAC ERROR UTILITY AFTER THE ERROR TYPEOUT. TO RESUME TESTING FROM THE POINT OF THE "HALT" SIMPLY DEPRESS CONTINUE.

B. POWER FAIL HALT

WHEN A POWER DOWN IS DETECTED, THE PROGRAM HALTS IN THE POWER FAIL UTILITY ROUTINE. IF FOR SOME REASON THE AUTO-START FEATURE FAILS TO RESTART THE PROGRAM, THE PROGRAM WILL "LOCK" ON THIS HALT IF CONTINUE IS DEPRESSED. IN THIS CASE THE PROGRAM MUST BE RESTARTED.

C. TRAP CATCHER HALTS

ALL INACTIVE VECTORS ARE SET UP WITH THE STANDARD PDP11 TRAP CATCHER AS DESCRIBED BELOW:

```
VN   / VN+2
VN+2/ HALT
```

IF A TRAP OR INTERRUPT OCCURS TO A VECTOR THAT HAS NOT BEEN SET UP BY THE TEST ROUTINE, A "HALT" OCCURS IN THE VECTOR AREA. THE ADDRESS DISPLAY INDICATES WHICH VECTOR THE PROGRAM TRAPPED TO AND THE LAST ENTRY PUSHED ON TO THE STACK INDICATES WHERE THE PROGRAM WAS WHEN THE TRAP OR INTERRUPT OCCURRED.

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

(NONE PROVIDED)

4.2 PROGRESS REPORTS

A. WHEN THE PROGRAM IS STARTED OR RESTARTED IT PRINTS THE TITLE MESSAGE:

"MAINDEC-11-DZDHM-"X" DH11 DIAGNOSTIC"

WHERE "X" IS THE REVISION LEVEL LETTER DESIGNATION.

B. WHEN THE PROGRAM BEGINS TESTING ON EACH DH11 IT TYPES THE FOLLOWING MESSAGE:

"TESTING DH11 #NN"

WHERE "NN" IS THE DEVICE NO. IN OCTAL (00-17)

C. WHEN THE PROGRAM COMPLETES A PASS (TESTED ALL SELECTED LINES ON ALL SELECTED DH11'S) IT TYPES:

"END PASS #XXXXX"

WHERE "X" IS THE PASS COUNT IN DECIMAL.

D. WHEN THE PROGRAM IS IN THE CONFIGURATION DIALOGUE (START AT 200, OR 210) AND SR8 AND SR0=1, THE PROGRAM WILL HALT AFTER ACCEPTING THE INPUT PARAMETERS AND TYPE THE FOLLOWING MESSAGE:

"DEPRESS CONTINUE TO START TESTING"

THE PURPOSE OF THIS HALT IS TO ALLOW THE USER TO DUMP THE UPDATED PROGRAM ON THE LOAD MEDIUM FOR NON-STANDARD CONFIGURATIONS. (SEE SECTION 2.2.2 AND 2.3.2)

5.0 DH11 DEVICE INFORMATION

5.1 ADDRESS AND VECTOR ASSIGNMENTS

THE DH11 USES FLOATING ADDRESSES AND IS LOCATED AFTER DJ11'S IN THE FLOATING ADD
BECAUSE THE DH11 HAS EIGHT REGISTERS, IT MUST BE ASSIGNED AN ADDRESS THAT IS A M
SYSTEM SHOULD HAVE CONSECUTIVE ADDRESSES.

EXAMPLE #1: A SYSTEM WITH NO DJ11'S BUT TWO DH11'S.

760 010 CANNOT USE FOR DH11'S BECAUSE NOT MULTIPLE OF 20.
760 020 FIRST DH11
760 040 SECOND DH11
760 060 DH11 GAP (INDICATES THAT THERE ARE NO MORE DH11'S).

EXAMPLE #2: A SYSTEM WITH ONE DJ11, TWO DH11'S:

760 010 FIRST DJ11
760 020 DJ11 GAP (INDICATES THAT THERE ARE NO MORE DJ11'S).
760 030 CANNOT USE FOR DH11'S BECAUSE NOT MULTIPLE OF 20.
760 040 FIRST DH11
760 060 SECOND DH11
760 100 DH11 GAP (INDICATES THAT THERE ARE NO MORE DH11'S).

THE DH11 VECTORS (2) FOLLOW THOSE OF THE DJ11 IN THE FLOATING VECTOR SPACE THAT
AT 300 ARE USED IN THE FOLLOWING ORDER: DC11; KL11/DL11-A, B; DP11; DM11-A; DN11
PA611 PUNCHES; DT11; DX11; DL11-C, D, E; DH11.

THE RECEIVER VECTOR IS THE LOWER NUMBERED VECTOR. THE PRIORITY OF THE RECEIVER A
SELECTABLE BY MEANS OF TWO STANDARD PDP11 PRIORITY JUMPER PLUGS. BR LEVEL 5 IS S

5.2 REGISTER DEFINITION

THE FOLLOWING SECTION DESCRIBES THE BIT ASSIGNMENTS WITHIN EACH REGISTER: BITS M
AS ZERO. ATTEMPTING TO WRITE INTO UNUSED OR READ ONLY BITS HAS NO EFFECT ON THOS
GENERATED BY THE PROCESSOR (E.G. UPON EXECUTION OF A RESET INSTRUCTION). TRANSMI

5.2.1 THE SYSTEM CONTROL REGISTER - ADDRESS X00

THE SYSTEM CONTROL REGISTER IS A BYTE-ADDRESSABLE REGISTER. THE BIT ASSIGNMENT I

BITS	DESCRIPTION
00-03	LINE SELECTION

EACH OF THE 16 LINES SERVED BY THE DH11 HAS ITS OWN STORAGE FOR LINE PAR
BYTE COUNT. THESE STORAGE LOCATIONS ARE LOADED BY THE PROGRAM VIA THE LI
REGISTER, AND BYTE COUNT REGISTER, BUT THE HARDWARE MUST FIRST BE TOLD W
CURRENT ADDRESS, OR BYTE COUNT CHANGED. THIS ROUTING IS ACCOMPLISHED BY
THE BINARY ADDRESS (0000-1111) OF THE DESIRED LINE. THESE BITS ARE READ/

04, 05 MEMORY EXTENSION

THE INFORMATION STORED IN THESE BITS BECOMES BITS 16 AND 17 RESPECTIVELY
PROGRAM INTO THE CURRENT ADDRESS REGISTER. THESE BITS ARE READ/WRITE BUT
OF BITS 4 AND 5 OF THE SYSTEM CONTROL REGISTER, NOT THE STATUS OF ADDRES
SEE THE SILO STATUS REGISTER FOR FURTHER INFORMATION. THIS ARRANGEMENT P
SAVE THE CONTENTS OF THE SYSTEM CONTROL REGISTER ACCURATELY.

06 RECEIVER INTERRUPT ENABLE

THIS BIT, WHEN SET, ENABLES RECEIVER INTERRUPTS (BIT 7)

07 RECEIVER INTERRUPT

THIS BIT, WHEN SET, INDICATES THAT THE NUMBER OF CHARACTERS STORED IN TH
SPECIFIED BY THE LOW BYTE OF THE SILO STATUS REGISTER. THIS BIT IS READ
WHERE IT IS READ/WRITE. SETTING OF THIS BIT WILL GENERATE AN INTERRUPT R
IS ALSO SET.

08 CLEAR NON-EXISTENT MEMORY INTERRUPT

THIS BIT, WHEN SET, CLEARS THE NON-EXISTENT MEMORY INTERRUPT FLIP-FLOP (
IS READ/WRITE.

09 MAINTENANCE

THIS BIT, WHEN SET, PLACES THE DH11 IN MAINTENANCE MODE.

10 NON-EXISTENT MEMORY

THIS BIT IS SET WHENEVER THE NPR HARDWARE PLACES THE ADDRESSES OF A MEMO
NO SLAVE SYNC IS RECEIVED IN 20 S. THIS INDICATES THAT THE ADDRESSED LOC
THIS BIT CAUSES AN INTERRUPT REQUEST IF SET WHILE TRANSMITTER AND NON-EX
THIS BIT IS READ ONLY, EXCEPT IN MAINTENANCE MODE, WHERE IT IS READ/WRIT

11 MASTER CLEAR

THIS BIT, WHEN SET, GENERATES "INITIALIZE" WITHIN THE DH11, CLEARING THE
EXACT BITS CLEARED ARE DISCUSSED IN THE SECTION ON INITIALIZATION. READ

```
                                        12      STORAGE INTERRUPT ENABLE

                                                THIS BIT, WHEN SET, PERMITS THE SETTING OF BIT 14 TO GENERATE AN INTERRU
```

```
                                        13      TRANSMITTER AND NON-EX-MEM INTERRUPT ENABLE

                                                THIS BIT, WHEN SET, PERMITS THE SETTING OF BIT 10 OR 15 TO GENERATE AN I

                                        14      STORAGE INTERRUPT

                                                THIS BIT IS SET WHEN THE RECEIVER SCANNER FINDS A RECEIVER HOLDING BUFFE
                                                STORE THAT CHARACTER IN THE SILO, AND CANNOT DO SO BECAUSE OF A LACK OF
                                                AN INTERRUPT REQUEST IF BIT 12 IS SET. THIS BIT IS READ ONLY, EXCEPT IN
                                                IT IS READ/WRITE.

                                        15      TRANSMITTER INTERRUPT

                                                THIS BIT IS SET WHEN THE DH11 CONCLUDES AN NPR CYCLE THAT INCREMENTED A
                                                CHARACTER IN A MESSAGE BUFFER WAS LOADED INTO A UART TRANSMITTER HOLDING
                                                REQUEST IF BIT 13 IS SET.  THIS BIT IS READ/WRITE. (IT IS SET DURING AN
```

5.2.2 NEXT RECEIVED CHARACTER REGISTER ADDRESS X02

BITS	DESCRIPTION
00-07	NEXT RECEIVED CHARACTER THESE BITS CONTAIN THE NEXT RECEIVED CHARACTER, RIGHT JUSTIFIED. THE LEA
08-11	LINE NUMBER THESE BITS INDICATE THE LINE NUMBER ON WHICH THE NEXT RECEIVED CHARACTER LEAST SIGNIFICANT BIT.
12	PARITY ERROR THIS BIT IS SET IF THE PARITY OF THE RECEIVED CHARACTER DOES NOT AGREE W
13	FRAMING ERROR THIS BIT IS SET IF THE RECEIVER SAMPLES A LINE FOR THE FIRST STOP BIT, A (LOGICAL 0). THIS CONDITION USUALLY INDICATES THE RECEPTION OF A BREAK.
14	DATA OVERRUN THIS BIT IS SET WHEN THE RECEIVED CHARACTER WAS PRECEDED BY A CHARACTER RECEIVER SCANNER TO SERVICE THE UART RECEIVER HOLDING BUFFER. REFER TO T FURTHER DETAILS ON DOUBLE-BUFFERED RECEPTION.
15	VALID DATA PRESENT THIS BIT INDICATES THAT THE DATA PRESENTED IN BITS 14-00 IS VALID. IT PE CHARACTERS FROM THE SILO UNTIL IT IS EMPTY. THIS IS DONE BY READING THIS IS OBTAINED FOR WHICH BIT 15 IS A ZERO. THE ENTIRE NEXT RECEIVED CHARACT ONLY ON A WORD BASIS.

5.2.3 LINE PARAMETER REGISTER ADDRESS X04

THIS REGISTER SHOULD BE LOADED ONLY AFTER THE LINE SELECTION BITS OF THE SYSTEM LINE TO WHICH THESE PARAMETERS APPLY. THIS REGISTER IS WRITE ONLY.

BITS	DESCRIPTION
00-01	CHARACTER LENGTH

THESE BITS SHOULD BE SET AS SHOWN TO RECEIVE AND TRANSMIT CHARACTERS OF

BIT 01	BIT 00	
0	0	5 BIT
0	1	6 BIT
1	0	7 BIT
1	1	8 BIT

02 TWO STOP BITS

THIS BIT, WHEN SET, CONDITIONS A LINE TRANSMITTING WITH 6, 7, OR 8-BIT C MARKS. IF THE LINE IS TRANSMITTING 5-BIT CODE, ASSERTION OF THIS BIT CAU 1.5 STOP MARKS. IF THIS BIT IS NOT ASSERTED, 1 STOP MARK IS SENT.

03 NOT USED

04 PARITY ENABLED

IF THIS BIT IS SET, CHARACTERS TRANSMITTED ON THIS LINE WILL HAVE AN APP RECEIVED ON THIS LINE WILL HAVE THEIR PARITY CHECKED.

05 ODD PARITY

IF THIS BIT AND BIT 4 ARE SET, CHARACTERS OF ODD PARITY WILL BE GENERATE WILL BE EXPECTED TO HAVE ODD PARITY. IF THIS BIT IS NOT SET, BUT BIT 4 I GENERATED ON THIS LINE AND INCOMING CHARACTERS WILL BE EXPECTED TO HAVE OF THIS BIT IS IMMATERIAL.

06-09 RECEIVER SPEED

THE STATE OF THESE BITS DETERMINES THE OPERATING SPEED FOR THIS LINE'S R BELOW IS APPLICABLE.

10-13 TRANSMITTER SPEED

THE STATE OF THESE BITS DETERMINES THE OPERATING SPEED FOR THIS LINE'S T TABLE ON THE NEXT PAGE IS APPLICABLE.

SPEED TABLE FOR RECEIVER AND TRANSMITTER SPEEDS:

	BIT				
TRANSMITTER	13	12	11	10	
RECEIVER	9	8	7	6	
	--	--	--	--	
	0	0	0	0	ZERO BAUD
	0	0	0	1	50 BAUDS
	0	0	1	0	75 BAUDS
	0	0	1	1	110 BAUDS
	0	1	0	0	134.5 BAUDS
	0	1	0	1	150 BAUDS
	0	1	1	0	200 BAUDS
	0	1	1	1	300 BAUDS
	1	0	0	0	600 BAUDS
	1	0	0	1	1200 BAUDS
	1	0	1	0	1800 BAUDS
	1	0	1	1	2400 BAUDS
	1	1	0	0	4800 BAUDS
	1	1	0	1	9600 BAUDS
	1	1	1	0	EXTERNAL INPUT A
	1	1	1	1	EXTERNAL INPUT B

14 HALF DUPLEX/FULL DUPLEX

IF THIS BIT IS SET, THIS LINE WILL OPERATE IN HALF-DUPLEX MODE. IF NOT S
IN FULL-DUPLEX MODE.

IN THIS APPLICATION HALF-DUPLEX MEANS THAT THE DH11 RECEIVER IS BLINDED

15 AUTO-ECHO ENABLE

WHEN THIS BIT IS SET, CHARACTERS RECEIVED ON THIS LINE WILL BE HARDWARE
FURTHER DETAILS.

5.2.4 CURRENT ADDRESS REGISTER ADDRESS X06

THIS REGISTER SHOULD BE LOADED ONLY AFTER THE SYSTEM CONTROL REGISTER (SCR) HAS
DESIRED LINE NUMBER. WHEN THIS REGISTER IS LOADED, ADDRESS BITS 00-15 ARE TRANSF
MEMORIES IN THE DH11 FROM BITS 00-15 OF THIS REGISTER. ADDRESS BITS 16-17 ARE TR
MEMORIES IN THE DH11 FROM BITS 4-5 OF THE SYSTEM CONTROL REGISTER.

INTERRUPTS MUST BE INHIBITED OR THE SCR SAVED BETWEEN THE SETTING OF THE SCR BIT
ADDRESS REGISTER.

WHEN THIS REGISTER IS READ, IT WILL INDICATE THE CURRENT ADDRESS OF THE LINE SEL
BITS 16 AND 17 WILL APPEAR IN THE SILO STATUS REGISTER, BITS 6 AND 7.

5.2.5 BYTE COUNT REGISTER ADDRESS X10

IN THE SAME FASHION AS THE LINE PARAMETER AND CURRENT ADDRESS REGISTERS, THIS RE
FIRST SELECTING A LINE NUMBER BY MEANS OF THE LOWER-ORDER FOUR BITS OF THE SYSTE
LOADED WITH THE TWO'S COMPLEMENT OF THE NUMBER OF CHARACTERS (BYTES) TO BE TRANS
IS READ/WRITE.

INTERRUPTS MUST BE INHIBITED OR THE SCR SAVED BETWEEN THE SETTING OF THE SCR BIT
COUNT REGISTER

5.2.6 BUFFER ACTIVE REGISTER (BAR) ADDRESS X12

THIS REGISTER CONTAINS ONE BIT FOR EACH LINE. THE BITS ARE INDIVIDUALLY SET USIN
TRANSMISSION ON THE ASSOCIATED LINE. THE BIT IS CLEARED BY THE HARDWARE WHEN THE
IS LOADED INTO THE TRANSMITTER DATA HOLDING REGISTER OF THE UART FOR THIAT LINE.
THE CLEARING OF A BAR DOES INDICATE THAT A MESSAGE MAY BE SENT, IT DOES NOT INDI
FROM THE PRECEDING MESSAGE HAVE BEEN COMPLETELY SENT. SPECIFICALLY, TWO MORE CHA
BIT CLEARS. THESE ARE THE LAST TWO CHARACTERS OF THE MESSAGE; ONE OF THEM WAS JU
AND ONE WAS THAT FINAL CHARACTER THAT WAS LOADED INTO THE HOLDING REGISTER, THUS
IS A NORMAL CONSEQUENCE OF DOUBLE-BUFFERED TRANSMISSION AND IS MENTIONED HERE FO
WANT TO WRITE PROGRAMS THAT CONTROL SUCH MODEM LEADS ARE REQUEST TO SEND. REQUES
DROPPED UNTIL AT LEAST TWO CHARACTER TIMES AFTER THE BAR BIT FOR A GIVEN LI CL

THIS TIMING MAY BE EFFECTED BY SENDING TWO EXTRA (NULL) CHARACTERS IN A MESSAGE

CLEARING A BAR BIT SHOULD NOT BE USED TO ABORT TRANSMISSION ON A LINE. RATHER, T
TO ZERO. THE BUFFER ACTIVE REGISTER BITS ARE READ/WRITE.

5.2.7 BREAK CONTROL REGISTER ADDRESS X14

THIS REGISTER CONTAINS ONE BIT FOR EACH LINE. SETTING A BIT IN THIS REGISTER WIL
ON THE LINE CORRESPONDING TO THAT BIT NUMBER. CLEARING THE BIT WILL TERMINATE TH
MAY BE TIMED BY SENDING CHARACTERS DURING THE BREAK INTERVAL, SINCE THESE CHARAC
FURTHER COMMENTS CONCERNING THE TRANSMISSION OF BREAK SIGNALS MAY BE FOUND IN TH

5.2.8 SILO STATUS REGISTER ADDRESS X16

THIS REGISTER IS ACTUALLY TWO BYTE-SIZED REGISTERS. THE BIT ASSIGNMENTS ARE:

BIT	DESCRIPTION
00-05	SILO ALARM LEVEL

THE PROGRAM MAY LOAD AN INTEGRAL POWER OF 2 BETWEEN 0 AND 63 INTO THIS L
WHEN THE NUMBER OF CHARACTERS STORED IN THE SILO EXCEEDS THAT NUMBER, AN
REGISTER BIT 7) IS GENERATED, IF SYSTEM CONTROL REGISTER BIT 6 IS SET. T

06-07	READ EXTENDED MEMORY

THESE BITS ARE READ ONLY AND CONTAIN THE A16 AND A17 BITS OF THE CURRENT
SELECTION BITS OF THE SYSTEM CONTROL REGISTER ARE POINTING.

08-13	SILO FILL LEVEL

THESE BITS ARE AN UP-DOWN COUNTER THAT INDICATES THE ACTUAL NUMBER OF CH
BE NOTED THAT THERE ARE SIX BITS, HENCE NUMBERS BETWEEN 0 AND 63 CAN BE
ENTRIES AND THE FILL LEVEL APPEARS AS 00000, BUT ONE MAY EASILY TELL THE
SILO (00000) AND A FULL SILO (00000) BY CHECKING THE STORAGE OVERFLOW BI
THESE BITS ARE READ ONLY.

5.3 DH11 FUNCTIONAL LOGIC PARTITIONING

THIS SECTION LISTS ALL OF THE PRINTS FOR ALL OF THE MODULES IN THE DH11 SUBSYSTEM. IT BRIEFLY SUMMARIZES THE FUNCTIONAL LOGIC DESCRIBED ON EACH PRINT. THIS INFORMATION MAY PROVE USEFUL FOR A MODULE OR CHIP REPLACEMENT GUIDE WHEN THE FUNCTIONAL AREA OF LOGIC THAT IS FAULTY IS KNOWN TO BE INTERMITTENTLY FAILING.

M7277 CURRENT ADDR REG MEMORY AND ADDR SELECT

SH3: CONTROL STROBE MUX FOR THE "LPR" REGISTER

TRANSMITTER DATA MUX WITH AUTO-ECHO CONTROL LOGIC SELECTION

MSYN / SSYN TIMING CHAIN

DH11 MASTER CLEAR LOGIC

SH4: UNIBUS ADDRESS SELECTION LOGIC WITH JUMPERS

TRANSMITTER SCAN COUNTER WITH XMITTER STATUS MULTIPLEXOR (BAR N * TBMT N)

SH5: BYTE COUNT AND CURRENT ADDRESS MEMORY WRITE TIMING LOGIC

CURRENT ADDRESS MEMORY LOGIC BITS<17:08>

UNIBUS ADDRESS DRIVERS BITS <17:08>

SH6: BYTE COUNT AND CURRENT ADDRESS MEMORY ADDRESS SELECT MULTIPLEXOR

CURRENT ADDRESS MEMORY LOGIC FOR BITS <07:00>

UNIBUS ADDRESS DRIVERS FOR BITS <07:00>

TRANSMITTER EVEN/ODD BYTE DATA MULTIPLEXOR

M7279 FIFO BUFFER

SH1: INPUT DATA MULTIPLEXOR FOR SILO MEMORY

SH2: SILO MEMORY CHIPS (FOUR 64 X 4 CHIPS)

SILO MEMORY READ/WRITE TIMING LOGIC

"SSR" REGISTER BITS <13:08>

SILO ALARM LEVEL COMPARATOR

```
1736                                                    RECEIVED "DATA READY" STATUS FLAG
1737
```

```
M7288   LINE PARAMETER CONTROL
*******************************

        SH3:    CLOCK TIMING SIGNAL BUFFERS

        SH4:    TRANSMITTER CLOCK SELECTION MULTIPLEXORS  LINES<03:00>

        SH5:    RECEIVER CLOCK SELECTION MULTIPLEXORS  LINES <03:00>

                AUTO ECHO AND HALF DUPLEX CONTROL       LINES<03:00>

        SH6:    TRANSMITTER CLOCK SELECTION MULTIPLEXORS  LINES<07:04>

        SH7:    RECEIVER CLOCK SELECTION MULTIPLEXORS  LINES <07:04>

                AUTO ECHO AND HALF DUPLEX CONTROL       LINES <07:04>

        SH8:    TRANSMITTER CLOCK SELECTION MULTIPLEXORS  LINES<11:08>

        SH9:    RECEIVER CLOCK SELECTION MULTIPLEXORS  LINES <11:08>

                AUTO ECHO AND HALF DUPLEX CONTROL       LINES <11:08>

        SH10:   TRANSMITTER CLOCK SELECTION MULTIPLEXORS  LINES<15:12>

        SH11:   RECEIVER CLOCK SELECTION MULTIPLEXORS  LINES <15:12>

                AUTO ECHO AND HALF DUPLEX CONTROL       LINES <15:12>

M7280 #1        MULTIPLE UART CARD FOR LINES <0-7>
**************************************************

        SH2:    UART CHIPS      BIT<1:0>

                RECEIVER SCAN MULTIPLEXORS (CLR R DONE, STB RD, MASTER DA, AND M

        SH3:    UART CHIPS      BIT<3:2>

        SH4:    UART CHIPS      BIT<5:4>

                RECEIVER SCAN MULTIPLEEXORS (MASTER FE AND MASTER PE,

        SH5:    UART CHIPS      BIT<7:6>

                TRANSMITTER SCAN MULTIPLEXOR

                -12 VOLT DC REGULATOR
```

```
M7280 #2        MULTIPLE UART CARD FOR LINES (8-15)
*****************************************************

        SH2:    UART CHIPS      BIT(9:8)

                RECEIVER SCAN MULTIPLEXORS (CLR R DONE, STB RD, MASTER DA, AND M

        SH3:    UART CHIPS      BIT(11:10)

        SH4:    UART CHIPS      BIT(13:12)

                RECEIVER SCAN MULTIPLEEXORS (MASTER FE AND MASTER PE)

        SH5:    UART CHIPS      BIT(15:14)

                TRANSMITTER SCAN MULTIPLEXOR

                -12 VOLT DC REGULATOR

M7289   SYSTEM CONTROL AND RCVR SCAN
************************************

SH3:    TRANSMITTER AND RECEIVER SCAN LOGIC

SH4:    TRANSMITTER AND RECEIVER SCAN TIMING

SH5:    HALF-DUPLEX CONTROL LOGIC

        UART DATA MULTIPLEXORS

SH6:    SYSTEM CONTROL REGISTER

M4540   DC11 - DH11 CLOCK
*************************

SH1:    CRYSTAL OSCILLATOR AND FREQUENCY DIVIDERS

M796    UNIBUS MASTER CONTROL
*****************************

SH1:    NPR CONTROL LOGIC

M7281 #1        INTERRUPT CONTROL
*********************************

"A" SECTION:    RECEIVER INTERRUPT CONTROL LOGIC

"B" SECTION:    TRANSMITTER INTERRUPT CONTROL LOGIC

M7281 #2        NPR CONTROL
****************************

"A" SECTION:    USED TO GAIN CONTROL OF THE BUS FOR NPR XFERS

"B" SECTION:    (NOT USED)
```

```
M7278   REGISTER AND BYTE CONTROL
*********************************

SH3:    CLEAR "BAR" REG MULTIPLEXOR

        BYTE COUNT MEMORY AND CONTROL - BITS<15:08>

        UNIBUS RECEIVERS        BIT<15:08>

SH4:    BYTE COUNT MEMORY AND CONTROL LOGIC     BIT<07:00>

        UNIBUS DATA RECEIVERS   BIT<07:00>

SH5 - SH8:      "BAR", "BCR", "LPR", AND "SSR" REGISTERS PLUS THE
                DATA OUTPUT MUX AND UNIBUS DRIVERS FOR DATA LINES.

        SH5:    BIT<15:12>      SH7:    BIT<07:04>
        SH6:    BIT<11:08>      SH8:    BIT<03:00>
```

5.4 DH11 MODULE ALLOCATION CHART

VIEW FROM WIRING SIDE

	SLOT 1	SLOT 2	SLOT 3	SLOT 4	SLOT 5	SLOT 6	SLOT 7
	M920	M7821	M7277	M7287	M7289	M7821	M7360
	CABLE						
ROW A	UNIBUS CONNECTOR (NOTE #3)	NPR CNTL	REG & BYTE CNT	CURRENT ADDRS & ADDRS	SYSTEM CNTL & RCV SCAN	INTR CNTL	PRIORITY SELECTOR (NOTE #9)
		M796				M405	M971
							CABLE
B		UNIBUS MASTER CNTL				EXTERNAL B CLOCK (NOTE #5)	DATA CABLE (NOTES #6 & #9)
	M7247	M7247				M7280	M7280
C	* CONTROL MUX LINES 8-15 (NOTE #7)	* CONTROL MUX LINES 0-7 (NOTE #8)				MULTIPLE UART LINES 0-7	MULTIPLE UART LINES 8-15
D							
	M105	M7246					
E	* ADDRESS SELECTOR (NOTE #7)	* CONTROL SCAN (NOTES #4) & #8					
	M7821						
F	* INTR CNTL (NOTE #7)						

1918
1919
1920
1921

FIGURE 2-4 DH11 MODULE UTILIZATION DIAGRAM
PAGE 2

NOTES:

1. IF END OF BUS, REPLACE M920 WITH M930.

2. IF LAST UNIT IN BASIC BOX, REPLACE M920 WITH BC11A CABLE WHEN EXPANDING TO PERIPHERAL BOX.

3. IF FIRST UNIT IN EXPANDER BOX, REPLACE M920 WITH BC11A CABLE.

4. E02 MUST BE G727 GRANT CONTINUITY IF MODEM CONTROL MODULE SET IS NOT INSTALLED. * DENOTES MODEM CONTROL OPTION, WITH DH11-AA OR AC.

5. MODULE SLOTS PROVIDE FOR ADDITIONAL CLOCK RATES.

6. FOR DIAGNOSTIC CHECKOUT OF DH11-AA, AB, OR AC, REPLACES M971 WITH M974.

7. THIS SLOT CONTAINS MODEM CONTROL MODULE M7807 WITH DH11-AD.

8. THIS SLOT CONTAINS MODEM CONTROL MODULE M7808 WITH DH11-AD.

9. THIS SLOT CONTAINS EIA CONVERTER AND PRIORITY MODULE M5906 FOR DH11-AD OR AE.

6.0 MAINTENANCE PROCEDURES

6.1 INTRODUCTION

THIS SECTION DESCRIBES HOW TO USE MD-DZDHM AS A TROUBLESHOOTING TOOL. IT OUTLINES SOME PRELIMINARY CHECKS TO MAKE BEFORE STARTING DETAILED DEBUG PROCEDURES. IT ATTEMPTS TO PROVIDE THE USER WITH SUGGESTIONS FOR PROCEEDING FROM THE ERROR PRINTOUT TO DETAILED ISOLATION OF THE FAULT.

6.2 PRELIMINARY CHECKS

A. VISUAL INSPECTION

PERFORM A VISUAL INSPECTION OF THE DH11 SUBSYSTEM TO INSURE THAT:

1) ALL MODULES ARE INSTALLED IN THEIR PROPER SLOTS (REFER TO PARA 5.3) AND ARE PROPERLY SEATED.
2) THE CABLING IS CORRECT AND ALL CABLE CONNECTORS ARE FIRMLY SEATED.
3) THE REQUIRED MAINTENANCE CONNECTORS ARE PROPERLY INSTALLED. (REFER TO SECTION 6.3)

B. POWER CHECKS

USE A SCOPE TO CHECK THE FOLLOWING POWER SUPPLY AND CONTROL SIGNALS ON THE DH11 BACKPLANE:

+15 VDC	GRAY WIRE
-15 VDC	BLUE WIRE
+5 VDC	RED WIRE
AC LO	YELLOW WIRE
DC LO	PURPLE WIRE

NOTES: USE THE BLACK WIRE FOR GROUND REFERENCE

"AC LO" AND "DC LO" SHOULD BOTH BE "HIGH" - "LOW" GOING GLITCHES ON THESE LINES CAN CAUSE UNUSUAL AND SUBTLE SYMPTOMS.

6.3 MAINTENANCE CONNECTORS

MOST OF THE TESTS IN MD-DZDHM USE HARDWARE DIAGNOSTIC AIDS TO TURN THE DATA AROUND. THESE AIDS REQUIRE THAT THE USER INSTALL SPECIFIC TURNAROUND CONNECTORS OR MODULES BEFORE RUNNING THE PROGRAM. DEPENDENT UPON THE SPECIFIC DH11 CONFIGURATION AND THE TYPE OF TESTING DESIRED, CERTAIN MAINTENANCE AIDS MUST BE INSTALLED AS OUTLINED BELOW:

A. DH11-AA, AB, OR AC CONFIGURATIONS

1) TESTING LOGIC FOR ALL LINES WITHOUT DATA CABLES
OR LEVEL CONVERTERS.

A. REMOVE THE DATA CABLE FROM SLOT B7 IN EACH
DH11 TO BE TESTED.
B. INSTALL AN M974 MAINT JUMPER MODULE INTO
SLOT B7 OF EACH DH11 TO BE TESTED.

2) TESTING ALL 16. LINES INCLUDING DATA CABLES
WHICH CONNECT TO DISTRIBUTION PANEL. DOES NOT
TEST LEVEL CONVERTER CIRCUITS LOCATED IN DISTRIBUTION
PANEL.

A. INSTALL THE M974 MAINT JUMPER MODULE INTO SLOT B3
OF THE MULTIPLEXOR DISTRIBUTION PANEL FOR EACH
DH11 TO BE TESTED. ALL LEVEL CONVERTERS IN THE
DISTRIBUTION PANEL MUST BE REMOVED FOR THIS TEST.

3) TESTING ONE OR MORE SINGLE LINES INCLUDING EIA
LEVEL CONVERTERS AND DEVICE CABLES WHICH ARE
NOT TESTED IN 1 AND 2 ABOVE.

A. INSTALL AN H315 TEST CONNECTOR AT THE END OF THE
DEVICE CABLE FOR EACH LINE TO BE TESTED.

B. DH11-AD CONFIGURATION

1. TESTING ALL 16. LINES WITHOUT DATA CABLES

A. DISCONNECT THE DATA CABLES (2) FROM THE TWO
CONNECTORS ON THE M5906 MODULE (SLOT AB7 OF
THE DH11 BACKPLANE.

B. INSTALL TWO H8611 TEST CONNECTORS ON THE M5906
IN PLACE OF THE CABLES.

C. IF MODEM CONTROL SECTION IS TO BE TESTED,
DISCONNECT 4 BC08R CABLES FROM DISTRIBUTION
PANEL AND CONNECT CABLES TO H861 TURNAROUND
CONNECTOR.

2. TESTING ONE OR MORE SINGLE LINES INCLUDING DATA CABLES

A. DISCONNECT THE DEVICE CABLE FROM THE DH11-AD
DISTRIBUTION PANEL FOR EACH LINE TO BE TESTED.

B. INSTALL AN H315 TEST CONNECTOR IN ITS PLACE
ON THE DH11-AD DISTRIBUTION PANEL.

NOTE: TO TEST THE DEVICE CABLE AS WELL, INSTALL
THE H315 TEST CONNECTOR AT THE END OF THE

DEVICE CABLE AND LEAVE THE DEVICE CABLE
CONNECTED TO THE DISTRIBUTION PANEL.

6.4 COMPLETE DH11 SUBSYSTEM CHECKOUT

COMPLETE DH11 SUB-SYSTEM VERIFICATION INVOLVES RUNNING THE FOLLOWING PROGRAMS IN THE SEQUENCE SUGGESTED:

A. MD-11-DZDHM DH11 DIAGNOSTIC

LOAD AND RUN THE BASIC DIAGNOSTIC CONFIGURED TO TEST ALL 16. LINES ON ALL DH11'S INSTALLED IN THE SYSTEM AND ALLOW IT TO COMPLETE AT LEAST TWO COMPLETE PASSES. (THE FIRST PASS IS A QUICK VERIFY WITHOUT SUB-TEST ITERATIONS AND THE SECOND PASS INCLUDES SUB-TEST ITERATIONS). IF ANY ERRORS ARE REPORTED, STOP HERE, AND REFER TO PARA 6.5 FOR SUBSEQUENT PROCEDURES.

B. MD-11-DZDHN DATA RELIABILITY TEST

LOAD AND RUN THE DATA RELIABILITY SUB-PROGRAM OF MD-11-DZDHN CONFIGURED TO TEST ALL 16. LINES ON ALL DH11'S INSTALLED IN THE SYSTEM AND ALLOW IT TO COMPLETE AT LEAST ONE PASS WITH SR07=0 (QUICK TEST MODE). IN THIS MODE ONE PASS TAKES APPROXIMATELY 5 MINUTES. IF ANY ERRORS ARE REPORTED, REFER TO THE DOCUMENTATION FOR MD-11-DZDHN FOR SUBSEQUENT PROCEDURES.

FOR MORE COMPLETE DATA RELIABILITY TESTING ,THE PROGRAM MAY BE RUN WITH SR07=1 (COMPLETE TEST) BUT THIS WILL REQUIRE A RUN TIME OF APPROX 15. MINUTES FOR EACH SELECTED LINE, SO IN MOST CASES IT IS ONLY USED FOR OVER-NIGHT RUNS.

C. SYSTEMS EXERCISER "DHAX" EXERCISER MODULE

WHERE "X" DESIGNATES THE REVISION LEVEL IN USE.

ASSUMING THAT BOTH THE DIAGNOSTIC AND THE DATA RELIABILITY INDICATE ERROR FREE PERFORMANCE, THE FINAL STEP IS TO RUN THE SYSTEM'S EXERCISER PROGRAM THAT INCLUDES A DH11 EXERCISER MODULE. THIS IS NECESSARY TO DETECT CERTAIN CLASSES OF BUS PROBLEMS THAT ONLY MANIFEST THEMSELVES WHEN THE DH11 IS RUN CONCURRENTLY WITH ALL THE OTHER DEVICES IN THE SYSTEM

IF ALL TESTS UP TO THIS POINT INDICATE ERROR FREE PERFORMANCE OF THE DH11, THE SUB-SYSTEM SHOULD BE CAPABLE OF RUNNING SYSTEM SOFTWARE. THERE ARE, HOWEVER, CERTAIN SUBTLE OR INTERMITTENT PROBLEMS THAT COULD STILL CAUSE THE DH11 SUB-SYSTEM TO FAIL IN THE OPERATING SYSTEM ENVIRONMENT. IN THESE RARE CASES, THE USER WILL HAVE TO USE THE SYMPTOMS GATHERED FROM THE FAILING MODE TO ISOLATE THE PROBLEM. ONCE A SYMPTOM IS RECOGNIZED, THE TWO OTHER PROGRAMS IN MD-DZDHN, THE ECHO TEST AND THE DATA PATTERNS/CABLE TESTS MAY

```
2119                                    PROVE USEFUL AS A TROUBLESHOOTING AID TO DUPLICATE THE
2120                                    PROBLEM FOR FAULT ISOLATION.
2121
```

6.5 MAINTENANCE HEADER DESCRIPTION

EACH TEST IN THE LISTING IS PREFACED BY A STANDARD MAINTENANCE HEADER TO PROVIDE INFORMATION THAT WILL FACILITATE RAPID ISOLATION OF THE FAULT THAT CAUSED A PARTICULAR TEST TO FAIL. EACH HEADER HAS THE SAME FORMAT (SEE EXAMPLE BELOW) AS FOLLOWS:

TEST ABSTRACT: THIS IS A CAPSULE SUMMARY OF WHAT THE TEST IS DESIGNED TO TEST AND HOW IT OPERATES.

ERRORS: LISTS THE PARTICULAR ERROR CALLS INVOKED BY THE TEST WHEN A FAULT IS DETECTED. (REFER TO PARA 3.1.2 AND 3.1.3 FOR A DETAILED DESCRIPTION OF THE ERROR CALL INFORMATION)

SYNC: LISTS ONE OR MORE SIGNALS THAT MAY BE USED TO SYNCHRONIZE THE OSCILLOSCOPE WHEN AN ERROR LOOP IS ESTABLISHED (SR09=1). FOR (H) SIGNALS USE (-) SLOPE TO TRIGGER ON THE TRAILING EDGE OF THE SIGNAL AND FOR (L) SIGNALS USE (+) SLOPE TO TRIGGER ON THE TRAILING EDGE.

DEBUG: CONTAINS SUGGESTIONS OF THINGS TO CHECK AND WHERE POSSIBLE THE MOST PROBABLE MODULE IS GIVEN.

KEY LOGIC: CONTAINS A LIST OF LOGIC SIGNALS AND/OR LOGIC COMPONENTS WITH MODULE NAMES AND PRINT NUMBERS TO RELATE THE TEST ROUTINE FUNCTION TO THE FUNCTIONAL AREAS OF LOGIC WITHIN THE PRINTS. WHERE POSSIBLE SIGNAL PIN NOS. ARE LISTED.

```
EXAMPLE:        MAINTENANCE HEADER FOR TEST 1
********

TEST ABSTRACT:
******************

        THIS TEST ATTEMPTS TO REFERENCE EACH OF THE EIGHT REGISTERS
IN THE DH11 SELECTED FOR TEST USING ITS ASSIGNED UNIBUS ADDRESS. IF
ANY ADDRESS FAILS TO RESPOND A BUS ERROR TRAP VECTORS THE TEST TO THE
ERROR SET-UP AND CALL ROUTINE. AFTER THE ERROR IS TYPED THE TEST WILL
TEST THE NEXT DH11 ADDRESS IN SEQUENCE UNTIL ALL EIGHT ARE TESTED.

ERRORS:
*******

1.) [ERROR  1] REPORTS THAT THE REGISTER WHOSE ADDRESS IS IN R2 FAILED
    TO RESPOND WITH "SSYN" WHEN REFERENCED.

SYNC:           (NONE)
*****

DEBUG:
******

1.) PROBLEM IS MOST LIKELY THE M7277 MODULE.
2.) IF ALL EIGHT REGISTERS FAIL TO RESPOND, MAKE SURE THAT YOU CONFIGURED
    THE PROGRAM PROPERLY BEFORE STARTING.  IF YOU DID, CHECK THE SETTINGS
    OF THE ADDRESS SELECT JUMPERS ON THE M7277 MODULE.
3.) IF ONE OR MORE RESPONDED PROPERLY, SET UP AN ERROR SCOPE LOOP AND
    BACKTRACK THROUGH THE LOGIC STARTING WITH THE KEY LOGIC SIGNALS
    LISTED BELOW.

KEY LOGIC:
**********

        M7277   SH3     SSYN H                  CE2
                        DEVICE RESPONDING L     E72-6
                SH4     DEVICE SELECTED H       E09-11

\
```

```
                                .NLIST  CND,MD,MC
                                .LIST   TOC,ME,SEQ,BIN
        165400                  $SWR=165400
                                .ENABLE ABS
                                .TITLE  MAINDEC-11-DZDHM-C
                                ;*COPYRIGHT (C) 1976,1977
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY ED CROWLEY
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*
        000001                  $TN=1
                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>
                                .SBTTL  TRAP CATCHER

        000000                          .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
        000174                          .=174
000174  000000                  DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                  SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  026146                  JMP     @#INPARX        ;;JUMP TO STARTING ADDRESS OF PROGRAM
000204  000137  002156                  JMP     @#BEGIN         ;BEGIN EXECUTION WITH DEFAULT PARAMETERS
000210  000137  026136                  JMP     @#INPARC        ;INPUT PARAMETERS - DEVICE SELECTION ONLY

                                .SBTTL  ACT11 HOOKS

                                ;;*****************************************************************
                                ;HOOKS REQUIRED BY ACT11
        000214                          $SVPC=.                 ;SAVE PC
        000046                          .=46
000046  020744                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
        000052                          .=52
000052  120000                          .WORD   120000          ;;2)SET LOC.52 TO 120000
        000214                          .=$SVPC                 ;; RESTORE PC
                                .SBTTL  APT PARAMETER BLOCK

                                ;;*****************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;*****************************************************************
        000214                          .$X=.   ;;SAVE CURRENT LOCATION
        000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
```

```
000024  000200                  200      ;;FOR APT START UP
        000044                  .=44     ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  000214                  $APTHDR  ;;POINT TO APT HEADER BLOCK
        000214                  .=.$X    ;;RESET LOCATION COUNTER
                        ;;*************************************************************
                        ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                        ;INTERFACE SPEC.

000214                  $APTHD:
000214  000000          $HIBTS: .WORD   0        ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
000216  001232          $MBADR: .WORD   $MAIL    ;;ADDRESS OF APT MAILBOX (BITS 0-15)
000220  000036          $TSTM:  .WORD   30.      ;;RUN TIM OF LONGEST TEST
000222  000170          $PASTM: .WORD   120.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
000224  000170          $UNITM: .WORD   120.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
000226  000052                  .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)


                        .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
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  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  BITOO,BITO

                                ;*BASIC "CPU" TRAP VECTOR ADDRESSES
        000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
        000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
        000014                  TBITVEC=14              ;;"T" BIT
        000014                  TRTVEC= 14              ;;TRACE TRAP
        000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
        000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
        000024                  PWRVEC= 24              ;;POWER FAIL
        000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
        000034                  TRAPVEC=34              ;;"TRAP" TRAP
        000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
        000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
        000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
```

```
                                .SBTTL  COMMON TAGS

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

        001100                          .=1100
001100                          $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                          .WORD   0
001102     000                  $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103     000                  $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000                  $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000                  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000                  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000                  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114     000                  $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115     001                  $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000                  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000                  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000                  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000                  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                          .WORD   0
001134     000                  $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135     000                  $INTAG: .BYTE   0               ;;INTERPUPT MODE INDICATOR
001136  000000                          .WORD   0
001140  177570                  SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560                  $TKS:   177560                  ;;TTY KBD STATUS
001146  177562                  $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564                  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566                  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154     000                  $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155     002                  $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156     012                  $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157     000                  $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000                  $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                                ;;WHICH  ($REG0) WAS OBTAINED
001162  000000                  $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001164  000000                  $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001166  000000                  $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001170  000000                  $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001172  000000                  $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001174  000000                  $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001176  000000                  $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
001200  000000                  $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
001202  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
001204  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
001206  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
001210  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
001212  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
001214  000000                  $TMP5:  .WORD   0               ;;USER DEFINED
001216  000000                  $TMP6:  .WORD   0               ;;USER DEFINED
001220  000000                  $TMP7:  .WORD   0               ;;USER DEFINED
001222  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
```

```
001224  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001226     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
001227     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001230  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001232                          $MAIL:                          ;;APT MAILBOX
001232  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001234  000000                  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001236  000000                  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001240  000000                  $PASS:  .WORD   APASS           ;;PASS COUNT
001242  000000                  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001244  000000                  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001246  000000                  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
001250  000000                  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
001252                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
001252     000                  $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
001253     000                  $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
001254  000000                  $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
001256  000000                  $USWR:  .WORD   AUSWR           ;;USER SWITCHES
001260  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
001262     000                  $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
001263     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
001264  000000                  $MADR1: .WORD   AMADR1          ;;HIGH ADDRESS,BLK#1
                                ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001266     000                  $MAMS2: .BYTE   AMAMS2          ;;HIGH ADDRESS,M.S. BYTE
001267     000                  $MTYP2: .BYTE   AMTYP2          ;;MEM.TYPE,BLK#2
001270  000000                  $MADR2: .WORD   AMADR2          ;;MEM.LAST ADDRESS,BLK#2
001272     000                  $MAMS3: .BYTE   AMAMS3          ;;HIGH ADDRESS,M.S.BYTE
001273     000                  $MTYP3: .BYTE   AMTYP3          ;;MEM.TYPE,BLK#3
001274  000000                  $MADR3: .WORD   AMADR3          ;;MEM.LAST ADDRESS,BLK#3
001276     000                  $MAMS4: .BYTE   AMAMS4          ;;HIGH ADDRESS,M.S.BYTE
001277     000                  $MTYP4: .BYTE   AMTYP4          ;;MEM.TYPE,BLK#4
001300  000000                  $MADR4: .WORD   AMADR4          ;;MEM.LAST ADDRESS,BLK#4
001302  000000                  $VECT1: .WORD   AVECT1          ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001304  000000                  $VECT2: .WORD   AVECT2          ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001306  000000                  $BASE:  .WORD   ABASE           ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001310  000000                  $DEVM:  .WORD   ADEVM           ;;DEVICE MAP
001312  000000                  $CDW1:  .WORD   ACDW1           ;;CONTROLLER DESCRIPTION WORD#1
001314  000000                  $CDW2:  .WORD   ACDW2           ;;CONTROLLER DESCRIPTION WORD#2
001316  000000                  $DDW0:  .WORD   ADDW0           ;;DEVICE DESCRIPTOR WORD#0
001320  000000                  $DDW1:  .WORD   ADDW1           ;;DEVICE DESCRIPTOR WORD#1
001322  000000                  $DDW2:  .WORD   ADDW2           ;;DEVICE DESCRIPTOR WORD#2
```

```
001324  000000          $DDW3:   .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
001326  000000          $DDW4:   .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
001330  000000          $DDW5:   .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
001332  000000          $DDW6:   .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
001334  000000          $DDW7:   .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
001336  000000          $DDW8:   .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
001340  000000          $DDW9:   .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
001342  000000          $DDW10:  .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
001344  000000          $DDW11:  .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
001346  000000          $DDW12:  .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
001350  000000          $DDW13:  .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
001352  000000          $DDW14:  .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
001354  000000          $DDW15:  .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15


001356                  $ETEND:
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT


  001356                        $ERRTB:
                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 1

                                        EM1             ;"DH11 REGISTER REFERENCE CAUSED TIMEOUT"
  001356  030302                        DH1             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR "
  001360  030351                        DT1             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2
  001362  030430                        0               ;PRINT ALL OCTAL
  001364  000000

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 2

                                        EM2             ;"SYSTEM CONTROL REGISTER ERROR"
  001366  030446                        DH2             ; (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
  001370  030504                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
  001372  030602                        0               ;PRINT ALL OCTAL
  001374  000000

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 3

                                        EM3             ;"DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG"
  001376  030624                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
  001400  030504                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
  001402  030602                        0               ;PRINT ALL OCTAL
  001404  000000

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 4

                                        EM4             ;"LINE PARAMETER REGISTER ERROR"
  001406  030702                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
  001410  030504                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
  001412  030602                        0               ;PRINT ALL OCTAL
  001414  000000

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 5

                                        EM5             ;"BREAK CONTROL REGISTER ERROR"
  001416  030740                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
  001420  030504                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
  001422  030602                        0               ;PRINT ALL OCTAL
  001424  000000

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 6

                                        EM6             ;"SILO STATUS REGISTER ERROR"
  001426  030775                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
  001430  030504                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
  001432  030602                        0               ;PRINT ALL OCTAL
  001434  000000
```

```
                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 7

001436  031030                                  EM7             ;"CURRENT ADDRESS REGISTER ERROR - LINE #XX"
001440  030504                                  DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001442  030602                                  DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001444  000000                                  0               ;PRINT ALL OCTAL

                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 10

001446  031102                                  EM10            ;"BYTE COUNTER REGISTER ERROR - LINE #XX"
001450  030504                                  DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001452  030602                                  DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001454  000000                                  0               ;PRINT ALL OCTAL

                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 11

001456  031151                                  EM11            ;"UNEXPECTED DH11 RCVR INTERRUPT"
001460  030504                                  DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001462  030602                                  DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001464  000000                                  0               ;PRINT ALL OCTAL

                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 12

001466  031210                                  EM12            ;"UNEXPECTED DH11 XMITTR INTERRUPT"
001470  030504                                  DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001472  030602                                  DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001474  000000                                  0               ;PRINT ALL OCTAL

                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 13

001476  031251                                  EM13            ;"CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT"
001500  030504                                  DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001502  030602                                  DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001504  000000                                  0               ;PRINT ALL OCTAL

                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 14

001506  031332                                  EM14            ;"TRANSMITTER NPR LOGIC ERROR - LINE #  "
001510  030504                                  DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001512  030602                                  DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001514  000000                                  0               ;PRINT ALL OCTAL

                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 15

001516  031401                                  EM15            ;"XMITTR FAILED TO INTERRUPT - LINE #  "
001520  030504                                  DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001522  030602                                  DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001524  000000                                  0               ;PRINT ALL OCTAL

                                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 16

001526  031447                                  EM16            ;"RCVR FAILED TO INTERRUPT"
001530  030504                                  DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001532  030602                                  DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001534  000000                                  0               ;PRINT ALL OCTAL
```

```
                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 17

001536  031500                          EM17            ;"TRANSMITTER TIMING ERROR - LINE #  "
001540  031544                          DH6             ;" (PC)  (PS)  (SP)  TEST  DEVADR  SPEED   TIMEB   TIMEC"
001542  030602                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001544  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 20

001546  031642                          EM20            ;RECEIVER TIMING ERROR - LINE #  "
001550  031544                          DH6             ;" (PC)  (PS)  (SP)  TEST  DEVADR  SPEED   TIMEB   TIMEC"
001552  030602                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001554  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 21

001556  031703                          EM21            ;"RCVR FAILED TO INTERRUPT - LINE #  "
001560  030504                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001562  030602                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001564  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 22

001566  031747                          EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
001570  030504                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001572  030602                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001574  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 23

001576  032023                          EM23            ;"BASIC DATA TEST ERROR - LINE #  "
001600  032064                          DH7             ;" (PC)  (PS)  (SP)  TEST  DEVADR  CHRLNG  WAS  S/B "
001602  030602                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001604  000000                          0               ;PRINT ALL OCTAL


                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 24

001606  032161                          EM24            ;"AUTO ECHO TEST ERROR - LINE #  "
001610  030504                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001612  030602                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001614  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 25

001616  032221                          EM25            ;"BREAK BIT TEST ERROR - LINE #  "
001620  030504                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001622  030602                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001624  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 26

001626  032261                          EM26            ;"HALF-DUPLEX TEST ERROR - LINE #  "
001630  030504                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001632  030602                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
```

```
001634  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 27

001636  032323                    EM27            ;"UNEXPECTED BUS ERROR TRAP"
001640  032355                    DH3             ; (PC)  (PS)  (SP)  TEST  TRPPC  TRPPS
001642  032434                    DT3             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2
001644  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 30

001646  032452                    EM30            ;"UNEXPECTED RSVD INSTR TRAP"
001650  032355                    DH3             ; (PC)  (PS)  (SP)  TEST  TRPPC  TRPPS
001652  032434                    DT3             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2
001654  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 31

001656  032505                    EM31            ;"AUTO ECHO DATA COMPARE ERROR - LINE #  "
001660  032555                    DH4             ; (PC)   (PS)   (SP)    TEST   WASADR  SBADR   WAS    S/B "
001662  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001664  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 32

001666  032652                    EM32            ;"AUTO ECHO TEST TIMEOUT - LINE #  "
001670  032714                    DH5             ;" (PC)  (LPRG)   TEST"
001672  032742                    DT4             ;$ERRPC,$TMP0,$TMP2
001674  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 33

001676  032752                    EM33            ;"PARITY LOGIC TEST ERROR - LINE #  "
001700  030504                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B"
001702  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001704  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 34

001706  033015                    EM34            ;"MULTI-LINE PARITY DATA TEST ERROR - LINE #   - SUBTEST #  "
001710  032555                    DH4             ;" (PC)  (PS)  (SP)  TEST  WASADR  SBADR  WAS  S/B "
001712  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001714  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 35

001716  033110                    EM35            ;"MULTI-LINE PARITY DATA TEST TIMEOUT"
001720  033154                    DH14            ;" (PC)  (LPRG)  LINACT "
001722  033204                    DT6             ;"$ERRPC,$TMP0,$TMP3"
001724  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 36

001726  033214                    EM36            ;CHAR AVAILABLE TIMEOUT"
001730  032714                    DH5             ;" (PC)  (LPRG)   TEST"
```

```
001732  032742                    DT4             ;$ERRPC,$TMP0,$TMP2
001734  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 37

001736  033243                    EM37            ;"DATA COMPARE ERROR - LINE #  "
001740  032555                    DH4             ;" (PC)  (PS)  (SP)  TEST  WASADR  SBADR  WAS  S/B "
001742  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001744  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 40

001746  033301                    EM40            ;"BUFFER ACTIVE REG ERROR - LINE #  "
001750  030504                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001752  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001754  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 41

001756  033344                    EM41            ;"RCVR FALSE INTERRUPT"
001760  032714                    DH5             ;" (PC)  (LPRG)   TEST"
001762  032742                    DT4             ;$ERRPC,$TMP0,$TMP2
001764  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 42

001766  033371                    EM42            ;"SILO OVERFLOW ERROR"
001770  032714                    DH5             ;" (PC)  (LPRG)   TEST"
001772  032742                    DT4             ;$ERRPC,$TMP0,$TMP2
001774  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 43

001776  033415                    EM43            ;"SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT"
002000  030504                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
002002  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
002004  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 44

002006  033475                    EM44            ;"NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT"
002010  030504                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
002012  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
002014  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 45

002016  033557                    EM45            ;"XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT"
002020  030504                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
002022  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
002024  000000                    0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 46

002026  033635                    EM46            ;"CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE #  "
002030  033723                    DH10            ;" (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS     S/B"
```

```
002032  034020                    DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4
002034  000000                    0               ;PRINT ALL OCTAL

                          ;ERROR TABLE ITEM FOR ERROR MESSAGE 47

002036  034042                    EM47            ;"BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE #  "
002040  033[illegible]            DH10            ; (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS     S/B"
002042  034020                    DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4
002044  000000                    0               ;PRINT ALL OCTAL

                          ;ERROR TABLE ITEM FOR ERROR MESSAGE 50

002046  034123                    EM50            ;"TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  '
002050  030504                    DH2             ;" (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
002052  030602                    DT2             ;"$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
002054  000000                    0               ;PRINT ALL OCTAL

                          ;ERROR TABLE ITEM FOR ERROR MESSAGE 51

002056  034201                    EM51            ;"NPR LOGIC TEST 2 ERROR"
002060  034230                    DH11            ;" (PC)   LINACT  LINCHK   TEST   DEVADR  REGADR   WAS     S/B"
002062  034020                    DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4"
002064  000000                    0               ;PRINT ALL OCTAL

                          ;ERROR TABLE ITEM FOR ERROR MESSAGE 52

002066  034325                    EM52            ;"BASIC DATA COMPARE ERROR"
002070  030504                    DH2             ;" (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
002072  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
002074  000000                    0               ;PRINT ALL OCTAL

                          ;ERROR TABLE ITEM FOR ERROR MESSAGE 53

002076  034123                    EM50            ;"TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  "
002100  034356                    DH12            ;" (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
002102  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
002104  000000                    0               ;PRINT ALL OCTAL

                          ;ERROR TABLE ITEM FOR ERROR MESSAGE 54

002106  031747                    EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
002110  034356                    DH12            ;" (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
002112  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
002114  000000                    0               ;PRINT ALL OCTAL

                          ;ERROR TABLE ITEM FOR ERROR MESSAGE 55

002116  031747                    EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
002120  034453                    DH13            ;" (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG  SCRWAS  SCRS/B
002122  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
002124  000000                    0               ;PRINT ALL OCTAL

                          ;ERROR TABLE ITEM FOR ERROR MESSAGE 56

002126  034552                    EM56            ;"OVERRUN BIT FAILED TO SET - LINE #  "
002130  030504                    DH2             ;" (PC)   (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
```

E06


```
002132  030602                          DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
002134  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 57

002136  034617                          EM57            ;"STORAGE OVERFLOW BIT FAILED - LINE #  "
002140  030504                          DH2             ;" (PC)   (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
002142  030602                          DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
002144  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 60
002146  036073                          MSG4            ;MODEM CONTROL ERROR.. RUN DZDHK
002150  034666                          DH15            ;" (PC) DEVADR  LINE    "
002152  032742                          DT4             ;$ERRPC,$TMP0,$TMP2
002154  000000                          0
```

```
002156  005000                  BEGIN:  CLR     R0              ;INIT R0 TO INDICATE DEFAULT PARAMETERS
002160  005067  026104          BEGIN:  CLR     TITFLG          ;INIT TITLE MESSAGE FLAG
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
002164  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
002170  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
002172  022706  001140                  CMP     #SWR,R6 ;;DONE?
002176  001374                          BNE     .-6             ;;LOOP BACK IF NO
002200  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
002204  012737  021000  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
002212  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002220  012737  021270  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002226  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002234  012737  024054  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002242  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
002250  012737  024136  000024          MOV     #$PWRON,@#PWRVEC ;;POWER FAILURE VECTOR
002256  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002264  016767  016422  016412          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
002272  005067  176724                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
002276  005067  176722                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002302  112767  000001  176605          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002310  012767  002310  176570          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002316  012767  002316  176564          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1"  SETUP FOR A SOFTWARE SWITCH REGISTER.
002324  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002330  012737  002364  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002336  012767  177570  176574          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002344  012767  177570  176570          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002352  022777  177777  176560          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002360  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
002362  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
002364  012716  002372          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
002370  000002                          RTI
002372  012767  000176  176540  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
002400  012767  000174  176534          MOV     #DISPREG,DISPLAY
002406  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

002412  005067  176622                  CLR     $PASS           ;;CLEAR PASS COUNT
002416  132767  000200  176627          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
002424  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
002426  012767  001254  176504          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
002434                          67$:
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
002434  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
002440  001012                          BNE     68$             ;;BRANCH IF YES
002442  126727  176604  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
002450  001406                          BEQ     68$             ;;BRANCH IF YES
002452  026727  176462  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
002460  001005                          BNE     69$             ;;BRANCH IF NO
002462  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
002464  000403                          BR      69$
002466  112767  000001  176440  68$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
002474                          69$:
```

```
002474  012767  026620  175302  START1: MOV     #BUSER,ERRVEC   ;SET UP THE BUS ERROR VECTOR
002502  012767  000340  175276          MOV     #340,ERRVEC+2
002510  012767  026674  175272          MOV     #RESERR,RESVEC  ;SET UP THE RSVD INSTR VECTOR
002516  012767  000340  175266          MOV     #340,RESVEC+2
002524  005767  025540                  TST     TITFLG          ;HAVE WE TYPED TITLE ONCE ?
002530  001012                          BNE     1$              ;BR IF YES
002532  104401                          TYPE                    ;GO TYPE PROGRAM TITLE
002534  034722                          TIT_E
002536  005167  025526                  COM     TITFLG          ;SET FLAG - TYPE TITLE ONLY ONCE PER LOAD
002542  032777  000001  176370          BIT     #BIT0,@SWR      ;DO WE WANT TO AUTOSIZE?
002550  001002                          BNE     1$              ;BRANCH IF NOT.
002552  004767  022352                  JSR     PC,AUTOSZ       ;GO AUTOSIZE.
002556  005767  025140          1$:     TST     VCFLG           ;START AT 200 ??
002562  001413                          BEQ     11$             ;BR IF NOT
002564  032777  000001  176346          BIT     #BIT0,@SWR      ;ARE PARAMETERS TO BE INPUT MANUALLY?
002572  001003                          BNE     9$              ;BRANCH IF YES.
002574  016700  025430                  MOV     ADRVEC,R0       ;OTHERWISE, GET ADDRESSES BETWEEN VECTORS FROM AUTOSIZER
002600  000402                          BR      10$
002602  004767  023272          9$:     JSR     PC,INPARA       ;GO ASK FOR PARAMETERS
002606  005067  025110          10$:    CLR     VCFLG           ;RE INIT VECTOR FLAG
002612  005700                  11$:    TST     R0              ;USE DEFAULT PARAMETERS ?
002614  001407                          BEQ     START2          ;BR IF YES
002616  022700  177777                  CMP     #-1,R0          ;CHANGE DH SELECT PARAM ONLY ?
002622  001002                          BNE     2$              ;BR IF NOT
002624  000167  023420                  JMP     INPAR3          ;GO ASK FOR SELECT PARAM.
002630  000167  023324          2$:     JMP     INPAR           ;GO ASK FOR ALL PARAMETERS

002634  012767  027620  025406  START2: MOV     #DHADTB-2,ADPTR ;GET POINTER TO ADDRESS TABLE
002642  012767  027660  025402          MOV     #DHVCTB-2,VCPTR ;GET POINTER TO VECTOR TABLE
002650  012767  027722  025376          MOV     #BRLVL-2,BRPTR  ;GET POINTER TO BR LEVEL TABLE
002656  012767  177777  025356          MOV     #-1,DHNUM       ;START WITH DH #00
002664  012767  000001  024336          MOV     #1,SELMSK       ;SET UP DH11 BIT TEST MARKER

002672  005267  025344          RESTRT: INC     DHNUM           ;GENERATE DH11 DEV NUMBER
002676  062767  000002  025344          ADD     #2,ADPTR        ;UPDATE TABLE POINTERS
002704  062767  000002  025340          ADD     #2,VCPTR
002712  062767  000002  025334          ADD     #2,BRPTR
002720  036767  024304  024304          BIT     SELMSK,DHSEL    ;TEST FOR SELECTED DH11
002726  001004                          BNE     RSTRTA          ;BR IF SELECTED FOR TEST
002730  006367  024274          REST1:  ASL     SELMSK          ;SHIFT MARKER TO TEST NEXT DH11
002734  001737                          BEQ     START2          ;BR IF 16 TESTED - START OVER
002736  000755                          BR      RESTRT          ;GO TEST IF THIS ONE SELECTED
002740  017767  025304  024256  RSTRTA: MOV     @ADPTR,DHADR    ;SET UP DH11 ADDRESS
002746  017767  025300  024252          MOV     @VCPTR,DHVCT    ;SET UP THE DH11 VECTOR ENTRY
002754  017767  025274  025256          MOV     @BRPTR,DHRLVL   ;GET BR LEVEL VALUES
002762  004567  021640                  JSR     R5,SUNUM        ;GO SET DH NUMBER IN THE MESSAGE BUFFER
002766  030242                          DHNUM
002770  035012                          TITLE2+20
002772  104401                          TYPE                    ;GO PRINT "TESTING DH11 #XX"
002774  034772                          TITLE2
002776  004767  021316                  JSR     PC,LDTBF1       ;GO LOAD XMITTR OUTPUT BUFFER WITH
                                                                ;BINARY COUNT PATTERN
003002  012767  003002  176076          MOV     #.,$LPADR       ;INIT SCOPE LOOP RETURN
```

```
                                        ;;*****************************************************************
                                        ;*TEST 1        CHECK SSYN RESPONSE FROM ALL DH11 REGISTERS
                                        ;;*****************************************************************
                                        ;;
003010  000004                          TST1:   SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        *****************

                                                THIS TEST ATTEMPTS TO REFERENCE EACH OF THE EIGHT REGISTERS
                                        IN THE DH11 SELECTED FOR TEST USING ITS ASSIGNED UNIBUS ADDRESS. IF
                                        ANY ADDRESS FAILS TO RESPOND A BUS ERROR TRAP VECTORS THE TEST TO THE
                                        ERROR SET-UP AND CALL ROUTINE. AFTER THE ERROR IS TYPED THE TEST WILL
                                        TEST THE NEXT DH11 ADDRESS IN SEQUENCE UNTIL ALL EIGHT ARE TESTED.

                                        ERRORS:
                                        *******

                                        1.) [ERROR  1] REPORTS THAT THE REGISTER WHOSE ADDRESS IS IN R2 FAILED
                                            TO RESPOND WITH "SSYN" WHEN REFERENCED.

                                        SYNC:           (NONE)
                                        *****

                                        DEBUG:
                                        ******

                                        1.) PROBLEM IS MOST LIKELY THE M7277 MODULE.
                                        2.) IF ALL EIGHT REGISTTERS FAIL TO RESPOND, MAKE SURE THAT YOU CONFIGURED
                                            THE PROGRAM PROPERLY BEFORE STARTING.  IF YOU DID, CHECK THE SETTINGS
                                            OF THE ADDRESS SELECT JUMPERS ON THE M7277 MODULE.
                                        3.) IF ONE OR MORE RESPONDED PROPERLY, SET UP AN ERROR SCOPE LOOP AND
                                            BACKTRACK THROUGH THE LOGIC STARTING WITH THE KEY LOGIC SIGNALS
                                            LISTED BELOW.

                                        KEY LOGIC:
                                        **********

                                                M7277   SH3     SSYN H                  CE2
                                                                DEVICE RESPONDING L     E72-6
                                                        SH4     DEVICE SELECTED H       E09-11
                                        %
003012  010102                                  MOV     R1,R2           ;COPY IT INTO R2
003014  010205                                  MOV     R2,R5           ;ALSO R5
003016  062705  000020                          ADD     #20,R5          ;R5 WILL TELL US WHEN WE'VE TESTED ALL 8
003022  016746  174756                          MOV     ERRVEC,-(SP)    ;SAVE BUS ERROR VECTOR
003026  012767  003054  174750                  MOV     #3$,ERRVEC      ;GO TO 3$ IF REG FAILS TO RESPOND
003034  162702  000002                          SUB     #2,R2           ;SO WE START WITH FIRST REG

003040  062702  000002                  1$:     ADD     #2,R2           ;POINT TO A DH11 REGISTER
003044  020205                                  CMP     R2,R5           ;TESTED ALL EIGHT ??
003046  001412                                  BEQ     4$              ;BR IF YES
003050  005712                          2$:     TST     (R2)            ;ACCESS DH11 REG ADDR
003052  000772                                  BR      1$              ;BR WHEN ALL 8 ARE DONE

003054  004767  021262                  3$:     JSR     PC,SUER1        ;GO SET UP ERROR INFO
003060  022626                                  CMP     (SP)+,(SP)+     ;FIX SP BECAUSE OF TRAP
```

```
3025  003062  012767  003050  176020          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
3026  003070  104001                          ERROR   1               ;DH11 REGISTER FAILED TO RESPOND TO MSYN
3027
3028  003072  000762                          BR      1$              ;GO TEST NEXT ONE
3029  003074  012667  174704          4$:     MOV     (SP)+,ERRVEC    ;RESTORE BUS ERROR VECTOR
```

```
                                ;;*****************************************************************
                                ;;*TEST 2        TEST THAT "MASTER CLR" CAN CLEAR THE "SCR","LPR","BKR",AND "SSR" REGS
                                ;;*****************************************************************
003100  000004                  TST2:   SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST SETS ALL WRITEABLE BITS IN THE TARGET REGISTER
                                (SCR, LPR, BKR, AND SSR) THEN SETS BIT11 IN THE "SCR" (MASTER CLEAR)
                                AND CHECKS THAT IT INDEED CLEARED ALL BITS IN THE TARGET REGISTER.
                                IT PERFORMS THIS SEQUENCE FOR ALL TARGET REGISTERS.
                                IF A REGISTER FAILS TO CLEAR PROPERLY, THE ERROR IS REPORTED, AND THEN
                                THE ROUTINE TESTS THE NEXT REGISTER IN SEQUENCE.

                                ERRORS:
                                *******

                                1.) [ERROR  3] REPORTS THAT THE REGISTER WHOSE ADDRESS IS IN R2 FAILED
                                    TO CLEAR WHEN MASTER CLEAR WAS ACTIVATED.

                                SYNC:
                                *****

                                        M7289   SH6     SCR 11 H (MASTER CLEAR)         FK2

                                DEBUG:
                                ******

                                1.) IF THE ERROR REPORTS INDICATE THAT ALL REGISTERS ARE FAILING, ESTABLISH
                                    AN ERROR SCOPE LOOP AND PROCEED TO BACKTRACK THROUGH THE KEY LOGIC
                                    SIGNALS LISTED BELOW.
                                2.) IF ONLY ONE REGISTER FAILS WITH ALL SET BITS -- THEN LET TESTS 03-10
                                    RUN -- THEY WILL PROBABLY GIVE BETTER ISOLATION. USE ONE OF THESE
                                    TESTS TO DEBUG THE FAULT.

                                KEY LOGIC:
                                **********

                                        M7277   SH3     INIT A L                FR2
                                                        INIT B L                FM2
                                                        INIT A H                EF2
                                                        INIT B H                FV2
                                                SH4     LOAD SCR HIGH BYTE H    CP1

                                        M7289   SH6     SCR 11 H (MASTER CLEAR) FK2

                                        M7278   SH3     BUFF DATA 11 H          AA1
                                %
003102  012705  027242                  MOV     #MSTCLR,R5      ;GET POINTER TO ADDRESS TABLE
003106  010125                          MOV     R1,(R5)+        ;SET UP THE TEST ADDRESS TABLE SO THAT
003110  010125                          MOV     R1,(R5)+        ;IT CONTAINS THE ADDRESSES OF THE
003112  010125                          MOV     R1,(R5)+        ;SCR,LPR,BKR, AND SSR REGISTERS
003114  010125                          MOV     R1,(R5)+
003116  062745  000016                  ADD     #SSR,-(R5)      ;GENERATE SSR ADDRESS
003122  062745  000014                  ADD     #BKR,-(R5)      ;GENERATE BKR ADDRESS
```

```
003126  062745  000004                  ADD     #LPR,-(R5)        ;GENERATE LPR ADDRESS
003132  005745                          TST     -(R5)             ;POINT R5 TO FIRST ADDR ENTRY (SCR)
003134  005004                          CLR     R4                ;RESULT S/B 000000 AFTER MASTER CLEAR

003136  012502                  1$:     MOV     (R5)+,R2          ;GET REG ADDRESS
003140  022705  027252                  CMP     #MSTCLR+10,R5     ;DONE ALL FOUR REGS ??
003144  001415                          BEQ     TST3              ;;BR IF YES
003146  012712  177777          2$:     MOV     #-1,(R2)          ;SET 1'S IN REGISTER
003152  052711  004000                  BIS     #BIT11,(R1)       ;ISSUE MASTER CLEAR
003156  011203                          MOV     (R2),R3           ;GET CONTENT OF REGISTER
003160  001766                          BEQ     1$                ;;BR IF IT'S ALL ZEROES

003162  004767  021214                  JSR     PC,SUER2          ;GO SET UP ERROR INFO
003166  012767  003146  175714          MOV     #2$,$LPERR        ;SET UP ERROR LOOP RETURN
003174  104003                          ERROR   3                 ;MASTER CLR FAILED TO CLR SEL. REG.
003176  000757                          BR      1$                ;GO TEST NEXT REGISTER
```

```
                                ;;*****************************************************************
                                ;*TEST 3        TEST "SCR" REG R/W BITS CAN SET/CLR (NORMAL MODE)
                                ;;*****************************************************************
003200  000004                  TST3:   SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT EACH R/W BIT IN THE "SCR" REGISTER
                                CAN BE INDIVIDUALLY SET AND CLEARED IN NORMAL MODE (MAINT BIT = 0)
                                A BIT MASK (RGMSK1:  131177) IS USED TO DEFINE THE R/W BITS ( ALL BUT
                                BITS 14, 11, 10, 8, AND 7). THE TEST IS REPEATED ELEVEN TIMES WITH A
                                DIFFERENT BIT SELECTED FOR EACH TEST. R5 CONTAINS THE BIT CURRENTLY
                                BEING TESTED. IF AN ERROR IS DETECTED, IT IS REPORTED AND THEN THE TEST
                                RESUMES WITH THE NEXT BIT IN SEQUENCE UNTIL ALL HAVE BEEN TESTED.

                                ERRORS:
                                *******

                                1.) [ERROR  2] IS CALLED TO REPORT A FAILURE TO SET PROPERLY AND
                                AGAIN TO REPORT A FAILURE TO CLEAR PROPERLY.

                                SYNC:
                                *****

                                1.) SET FAILURE         M7277   SH4     LOAD SSR LOW BYTE H     CR1

                                2.) CLR FAILURE         M7277   SH4     LOAD SSR HIGH BYTE H    CP2

                                DEBUG:
                                ******

                                1.) IF ALL BITS FAIL - SUSPECT THE "LOAD SCR" SIGNALS ON THE M7277 SH4.
                                2.) IF ONLY ONE OR TWO BITS FAIL - SUSPECT EITHER THE "SCR" REGISTER
                                    FLOPS ON THE M7289 SH6, THE BUS RECEIVERS ON THE M7278 SH3 AND SH4,
                                    OR THE MULTIPLEXORS AND BUS DRIVERS ON THE M7289 SH5-8.

                                KEY LOGIC:
                                **********

                                        M7278   SH3     BUFF DATA <15:08> H
                                                SH4     BUFF DATA <07:00> H

                                        M7277   SH4     LOAD SCR LOW BYTE H     CT2
                                                        LOAD SCR HIGH BYTE H    CP1
                                                        DATA TO BUS H           EN2
                                                        DATA SOURCE <A,B,C> H   DU1,DU2,DT2

                                        M7289   SH5     BUF DATA TO BUS B H     E05-12
                                                        BUS DATA <15:12> L
                                                SH6     BUS DATA <11:08> L
                                                        SCR <15:00> H
                                                SH7     BUF DATA TO BUS A H     E05-8
                                                        BUS DATA <07:04> L
                                                SH8     BUS DATA <03:00> L
                                %
```

```
003202  012767  003232  175700          MOV     #4$,$LPERR      ;SET UP ERROR LOOP RETURN
003210  010102                          MOV     R1,R2           ;GET REGISTER ADDRESS
003212  012705  000001                  MOV     #1,R5           ;SET UP TO START WITH BIT00

003216  030567  024362          1$:     BIT     R5,RGMSK1       ;SHALL WE TEST THIS BIT ?
003222  001003                          BNE     4$              ;BR IF YES
003224  006305                  2$:     ASL     R5              ;SHIFT TO TST NEXT BIT
003226                          3$:
003226  001430                          BEQ     TST4            ;;<BR IF DONE ALL R/W BITS>
003230  000772                          BR      1$              ;GO TEST NEXT BIT

003232  010504                  4$:     MOV     R5,R4           ;RESULT S/B IN R4
003234  005012                          CLR     (R2)            ;INIT REG BEING TESTED
003236  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
003244  010512                          MOV     R5,(R2)         ;SET THE BIT
003246  011203                          MOV     (R2),R3         ;GET THE WAS DATA
003250  020403                          CMP     R4,R3           ;RESULT = S/B DATA ??
003252  001403                          BEQ     5$              ;BR IF YES

003254  004767  021122                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
003260  104002                          ERROR   2               ;SELECTED BIT FAILED TO SET IN SCR

003262  005004                  5$:     CLR     R4              ;SET UP TO CLEAR THE BIT S/B=000000
003264  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
003272  040512                          BIC     R5,(R2)         ;CLR THE SELECTED BIT
003274  011203                          MOV     (R2),R3         ;GET THE WAS DATA
003276  001403                          BEQ     6$              ;BR IF IT CLEARED

003300  004767  021076                  JSR     PC,SUER2        ;GO SET UP THE ERROR INFO
003304  104002                          ERROR   2               ;SELECTED BIT FAILED TO CLEAR IN SCR
003306  000746                  6$:     BR      2$              ;GO SELECT NEXT BIT
```

```
                                        ;;*****************************************************************
                                        ;*TEST 4          TEST "SCR" REG. READ ONLY BITS (NORMAL MODE)
                                        ;;*****************************************************************
003310  000004                          TST4:   SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        **************

                                                THIS TEST VERIFIES THAT THE "SCR" REGISTER READ ONLY BITS CAN
                                        NOT BE SET OR CLEARED IN NORMAL MODE. A BIT MASK (RGMSK2:  046600)
                                        IS USED TO DEFINE THE READ ONLY BITSS (14,11,10,8, AND 7). THE TEST
                                        IS REPEATED FIVE TIMES, ONCE FOR EACH BIT TO BE TESTED, AND ANY
                                        ERRORS DETECTED ARE REPORTED. AFTER THE ERROR REPORT THE TEST RESUMES
                                        WITH THE NEXT BIT IN SEQUENCE UNTIL ALL BITS HAVE BEEN TESTED.

                                        ERRORS:
                                        *******

                                        1.) [ERROR  2] IS CALLED TO REPORT ANY READ ONLY BIT THAT FAILED
                                            TO RESPOND PROPERLY.

                                        SYNC:
                                        *****

                                                M7277   SH4     LOAD SSR LOW BYTE H     CR1

                                        DEBUG:
                                        ******

                                                SAME AS FOR TEST 03

                                        KEY LOGIC:
                                        **********

                                                SAME AS FOR TEST 03

                                        %
003312  010102                                  MOV     R1,R2           ;MAKE IT THE REG. ADDR ALSO
003314  012705  000001                          MOV     #1,R5           ;INIT BIT TEST MARKER
003320  030567  024262                  1$:     BIT     R5,RGMSK2       ;IS IT A READ ONLY BIT ??
003324  001003                                  BNE     3$              ;BR IF IT IS - GO TEST IT
003326  006305                          2$:     ASL     R5              ;SHIFT BIT MARKER
003330  001420                                  BEQ     TST5            ;;BR IF DONE ALL BITS
003332  000772                                  BR      1$              ;GO TEST THIS BIT

003334  005004                          3$:     CLR     R4              ;RESULT S/B = 000000
003336  005012                                  CLR     (R2)            ;INIT REG BEING TESTED
003340  112761  000000  000016                  MOVB    #0,SSR(R1)      ;SCOPE SYNC
003346  010512                                  MOV     R5,(R2)         ;ATTEMPT TO SET A READ ONLY BIT
003350  011203                                  MOV     (R2),R3         ;GET THE WAS DATA
003352  001765                                  BEQ     2$              ;BR IF THE BIT DIDN'T SET

003354  004767  021022                          JSR     PC,SUER2        ;GO SET UP ERROR INFO
003360  012767  003334  175522                  MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN ADDR
003366  104002                                  ERROR   2               ;READ ONLY BIT SET IN "SCR"
003370  000756                                  BR      2$              ;CONTINUE WITH NEXT BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 5        TEST "SCR" REG. BITS THAT CAN BE SET/CLR IN MAINT. MODE
                                ;;*****************************************************************
003372  000004                  TST5:   SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST VERIFIES THAT THE "SCR" REGISTER READ-ONLY BITS
                                (14, 10, AND 07) CAN BE SET/CLR IN MAINT. MODE (SCR09=1) ONLY.
                                A BIT MASK (RGMSK6:  042200) IS USED TO DEFINE THE BITS TO TEST.
                                THE TEST PERFORMS THE FOLLOWING SEQUENCE:

                                        1. SELECT A BIT TO TEST
                                        2. SET MAINT. MODE
                                        3. SET THE BIT AND VERIFY THAT IT SET
                                        4. CLEAR THE MAINT. MODE BIT
                                        5. ATTEMPT TO CLEAR THE TEST BIT
                                        6. TEST TO SEE THAT IT DID NOT CLEAR
                                        7. SET MAINT. MODE
                                        8. CLEAR THE TEST BIT AND VERIFY THAT IT CLEARED
                                        9. REPEAT 1-7 UNTIL ALL BITS HAVE BEEN TESTED

                                ANY ERRORS DETECTED ARE REPORTED AND THE TEST RESUMES WITH THE NEXT
                                BIT IN SEQUENCE UNTIL ALL BITS HAVE BEEN TESTED.

                                ERRORS:
                                ********

                                1.) [ERROR  2] IS CALLED AT THREE DIFFERENT POINTS TO REPORT ONE OF
                                    THE THREE POSSIBLE FAILURE MODES DESCRIBED IN 3, 6, AND 7 IN THE
                                    ABSTRACT.

                                SYNC:
                                ******

                                1.) STEP 3 FAILURE TO SET WITH MAINT. MODE SET

                                        M7277   SH4     LOAD SSR LOW BYTE H     CR1

                                2.) STEP 5 FAILURE TO REMAIN SET WITH MAINT MODE NOT SET

                                        M7277   SH4     LOAD SSR HIGH BYTE H    CP2

                                3.) STEP 8 FAILURE TO CLEAR WITH MAINT. MODE SET

                                        M7277   SH4     LOAD LPR H              EP2

                                DEBUG:
                                ******

                                1.) ASSUMING THE PREVIOUS TESTS RAN SUCCESSFULLY THE FAULT IS MOST
                                    LIKELY THE M7289 MODULE.

                                KEY LOGIC:
                                **********
```

```
                                                SAME AS TEST 03 WITH THE FOLLOWING ADDITION

                                                M7289   SH4     74121 ONE-SHOTS E35-6, E23-6
                                        %
003374  012767  003424  175506                  MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
003402  010102                                  MOV     R1,R2           ;MAKE IT REG ADDR TOO
003404  012705  000001                          MOV     #1,R5           ;INIT BIT TEST MARKER
003410  030567  024202                  1$:     BIT     R5,RGMSK6       ;IS IT A READ ONLY BIT ??
003414  001003                                  BNE     3$              ;BR IF YES - TEST IT
003416  006305                          2$:     ASL     R5              ;SHIFT THE BIT MARKER
003420  001457                                  BEQ     TST6            ;;BR IF DONE ALL SELECTED BITS
003422  000772                                  BR      1$              ;GO TEST FOR THIS BIT

003424  010504                          3$:     MOV     R5,R4           ;SET UP S/B DATA
003426  052704  001000                          BIS     #BIT09,R4       ;PUT IN THE MAINT. BIT
003432  005012                                  CLR     (R2)            ;INIT REG BEING TESTED
003434  052712  001000                          BIS     #BIT09,(R2)     ;TURN ON MAINT. MODE
003440  112761  000000  000016                  MOVB    #0,SSR(R1)      ;SCOPE SYNC
003446  050512                                  BIS     R5,(R2)         ;SET THE SELECTED BIT
003450  011203                                  MOV     (R2),R3         ;GET THE WAS DATA
003452  020304                                  CMP     R3,R4           ;DID SELECTED BIT GET SET ??
003454  001404                                  BEQ     4$              ;BR IF IT DID

003456  004767  020720                          JSR     PC,SUER2        ;GO SET UP ERROR INFO
003462  104002                                  ERROR   2               ;SELECTED BIT FAILED TO SET IN MAINT MODE
003464  000754                                  BR      2$              ;GO TEST NEXT BIT

003466  042712  001000                  4$:     BIC     #BIT09,(R2)     ;TURN OFF MAINT. MODE
003472  042704  001000                          BIC     #BIT09,R4       ;CLR MAINT BIT IN S/B DATA
003476  112761  000000  000017                  MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
003504  040512                                  BIC     R5,(R2)         ;ATTEMPT TO CLR SELECTED BIT
003506  011203                                  MOV     (R2),R3         ;GET THE WAS DATA
003510  020304                                  CMP     R3,R4           ;DID BIT GET CLEARED ??
003512  001404                                  BEQ     5$              ;BR IF IT DIDN'T

003514  004767  020662                          JSR     PC,SUER2        ;GO SET UP ERROR INFO
003520  104002                                  ERROR   2               ;SELECTED BIT GOT CLEARED WITH MAINT MODE OFF
003522  000735                                  BR      2$              ;GO TEST NEXT BIT

003524  012704  001000                  5$:     MOV     #BIT09,R4       ;SET UP S/B DATA
003530  050412                                  BIS     R4,(R2)         ;SET MAINT. MODE
003532  012761  000000  000004                  MOV     #0,LPR(R1)      ;SCOPE SYNC
003540  040512                                  BIC     R5,(R2)         ;NOW CLR SELECTED BIT
003542  011203                                  MOV     (R2),R3         ;GET THE WAS DATA
003544  020304                                  CMP     R3,R4           ;DID BIT GET CLEARED OK ??
003546  001723                                  BEQ     2$              ;BR IF YES

003550  004767  020626                          JSR     PC,SUER2        ;GO SET UP ERROR INFO
003554  104002                                  ERROR   2               ;FAILED TO CLR SELECTED BIT IN MAINT MODE
003556  000717                                  BR      2$              ;GO SELECT NEXT BIT FOR TEST
```

```
                                ;;*****************************************************************
                                ;*TEST 6        TEST THAT ALL R/W BITS IN "LPR" CAN BE SET/CLR
                                ;;*****************************************************************
003560  000004                  TST6:   SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST VERIFIES THAT ALL R/W BITS IN THE "LPR" REG-
                                ISTER CAN BE SET AND CLEARED INDIVIDUALLY. A BIT MASK (RGMSK3:  177767)
                                IS USED TO DEFINE THE BITS TO BE TESTED (ALL BUT BIT03). THE TEST
                                SEQUENCE IS AS FOLLOWS:

                                        1. SELECT A BIT TO TEST
                                        2. SET THE BIT AND VERIFY IT SET
                                        3. CLEAR THE BIT AND VERIFY IT CLEARED
                                        4. REPEAT 1 THRU 3 UNTIL ALL BITS TESTED

                                ANY ERRORS DETECTED ARE REPORTED AND AFTER THE ERROR, THE TEST
                                RESUMES WITH THE NEXT BIT IN SEQUENCE.

                                ERRORS:
                                *******

                                1.) [ERROR  4] IS CALLED TO REPORT BOTH FAIL TO SET AND FAIL TO
                                    CLEAR FAULTS.

                                SYNC:
                                *******

                                1.) FAIL TO SET:        M7277   SH4     LOAD SSR LOW BYTE H     CR1
                                2.) FAIL TO CLEAR:      M7277   SH4     LOAD SSR HIGH BYTE H    CP2

                                DEBUG:
                                ******

                                1.) IF ALL BITS FAIL THE PROBLEM IS MOST LIKELY THE M7277 MODULE
                                    (LPR LOAD SIGNALS)
                                2.) IF NOT THEN IT IS PROBABLY AN "LPR" REGISTER CHIP OR
                                    BAD OUTPUT DATA MUX CHIP, BOTH ON THE M7278 MODULE.

                                KEY LOGIC:
                                **********

                                        M7277   SH4     LOAD LPR H              EP2

                                        M7278   SH5     LPR <15:12> L   (E52)
                                                SH6     LPR <11:08> L   (E37)
                                                SH7     LPR <07:04> L   (E59)
                                                SH8     LPR <03:00> L   (E61)
                                                SH5,6,7,8       OUTPUT MUX CHIPS  (74151'S PIN 2)

                                %
003562  012767  003616  175320          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
003570  010102                          MOV     R1,R2           ;COPY IT IN R2
003572  062702  000004                  ADD     #LPR,R2         ;GENERATE REGADR IN R2
```

```
003576  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MARKER
003602  030567  024002          1$:     BIT     R5,RGMSK3       ;TEST THIS BIT ??
003606  001003                          BNE     3$              ;BR IF YES
003610  006305                  2$:     ASL     R5              ;SHIFT THE MARKER
003612  001430                          BEQ     TST7            ;BR IF DONE ALL BITS
003614  000772                          BR      1$              ;GO TEST NXT BIT

003616  010504                  3$:     MOV     R5,R4           ;SET UP S/B DATA
003620  005012                          CLR     (R2)            ;INIT REG BEING TESTED
003622  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
003630  010512                          MOV     R5,(R2)         ;SET LPR BIT
003632  011203                          MOV     (R2),R3         ;GET THE WAS DATA
003634  020304                          CMP     R3,R4           ;DID IT SET
003636  001403                          BEQ     4$              ;BR IF IT SET PROPERLY

003640  004767  020536                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
003644  104004                          ERROR   4               ;LPR BIT FAILED TO SET PROPERLY

003646  005004                  4$:     CLR     R4              ;GET READY TO CLEAR SELECTED BIT
003650  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
003656  040512                          BIC     R5,(R2)         ;CLEAR THE BIT
003660  011203                          MOV     (R2),R3         ;GET THE WAS DATA
003662  001752                          BEQ     2$              ;BR IF BIT CLEARED PROPERLY

003664  004767  020512                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
003670  104004                          ERROR   4               ;LPR BIT FAILED TO CLEAR PROPERLY
003672  000746                          BR      2$              ;GO SELECT NEXT BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 7        TEST THAT ALL R/W BITS IN "BKR" CAN BE SET/CLR
                                ;;*****************************************************************
003674  000004                  TST7:   SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT ALL BITS IN THE BREAK CONTROL REGISTER
                                CAN BE SET AND CLEARED INDIVIDUALLY. IT USES A BIT MASK (RGMSK4:  177777)
                                TO DEFINE THE R/W BITS (ALL 16.). R5 ALWAYS CONTAINS THE BIT CURRENTLY
                                SELECTED FOR TEST. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SELECT A BIT TO TEST
                                        2. SET THE BIT AND VERIFY THAT IT SET PROPERLY
                                        3. CLEAR THE BIT AND VERIFY THAT IT CLEARED PROPERLY
                                        4. REPEAT 1 THRU 4 UNTIL ALL BITS HAVE BEEN TESTED.

                                ANY ERROR DETECTED IS REPORTED AND THE TEST RESUMES WITH THE NEXT BIT
                                IN SEQUENCE.

                                ERRORS:
                                ********

                                1.) [ERROR  5] IS CALLED TO REPORT BOTH FAIL TO SET PROPERLY AND FAIL
                                    TO CLEAR PROPERLY FAULTS.

                                SYNC:
                                ******

                                1.) FAIL TO SET:        M7277   SH4     LOAD SSR LOW BYTE H     CR1
                                2.) FAIL TO CLR:        M7277   SH4     LOAD SSR HIGH BYTE H    CP2

                                DEBUG:
                                ******

                                1.) THE ONLY DIFFERENCES IN THE DATA PATH HERE AND THAT FOT THE PREVIOUS
                                    TESTS ARE THE ACTUAL REGISTER CHIPS AND THE INPUT SELECTED ON THE
                                    OUTPUT DATA MULTIPLEXORS.

                                2.) IF ALL BITS FAIL THE PROBLEM IS MOST LIKELY THE M7277.

                                3.) IF ONLY ONE OR TWO FAIL THE PROBLEM IS MOST LIKELY THE M7278.

                                KEY LOGIC:
                                ***********

                                        M7277   SH4     LOAD BCR H              FU1
                                                        DATA TO BUS H           EN2
                                                        DATA SOURCE (A,B,C) H   DU1,DU2,DT2

                                        M7278   SH5 - SH8 74175 REGISTER CHIPS (E51,E38,E67,E60)
                                                SH5 - SH8 74151'S MUX CHIPS INPUT PIN 13
                                %
003676  012767  003732  175204          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
003704  010102                          MOV     R1,R2           ;GENERATE "BKR" ADDRESS IN R2
```

```
003706  062702  000014                  ADD     #BKR,R2
003712  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MARKER
003716  030567  023670          1$:     BIT     R5,RGMSK4       ;TEST THIS BIT ??
003722  001003                          BNE     3$              ;BR IF YES
003724  006305                  2$:     ASL     R5              ;SHIFT BIT MARKER
003726  001430                          BEQ     TST10           ;;BR IF ALL BITS TESTED
003730  000772                          BR      1$              ;GO TEST THE BIT

003732  010504                  3$:     MOV     R5,R4           ;SET UP S/B DATA
003734  005012                          CLR     (R2)            ;INIT REG BEING TESTED
003736  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
003744  050512                          BIS     R5,(R2)         ;SET THE SELECTED BIT IN "BKR"
003746  011203                          MOV     (R2),R3         ;GET THE WAS DATA
003750  020304                          CMP     R3,R4           ;DID BIT SET OK
003752  001403                          BEQ     4$              ;BR IF YES

003754  004767  020422                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
003760  104005                          ERROR   5               ;BKR BIT FAILED TO SET PROPERLY

003762  005004                  4$:     CLR     R4              ;SET UP S/B DATA
003764  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPEE SYNC
003772  040512                          BIC     R5,(R2)         ;CLEAR BKR BIT
003774  011203                          MOV     (R2),R3         ;GET THE BKR WAS DATA
003776  001752                          BEQ     2$              ;BR IF BKR BIT CLEARED OK

004000  004767  020376                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
004004  104005                          ERROR   5               ;BKR BIT FAILED TO CLR PROPERLY
004006  000746                          BR      2$              ;GO SELECT NEXT BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 10       TEST THAT ALL R/W BITS IN "SSR" CAN BE SET/CLR
                                ;;*****************************************************************
004010  000004                  TST10:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT ALL R/W BITS IN THE SILO STATUS REGISTER
                                (SSR) CAN BE SET AND CLEARED INDIVIDUALLY. IT USES A BIT MASK (RGMSK5:  100077)
                                TO DEFINE THE R/W BITS (15,5,4,3,2,1, AND 0). R5 ALWAYS CONTAINS THE BIT
                                CURRENTLY SELECTED FOR TEST. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SELECT A BIT TO TEST
                                        2. SET THE BIT AND VERIFY THAT IT SET PROPERLY
                                        3. CLEAR THE BIT AND VERIFY THAT IT CLEARED PROPERLY
                                        4. REPEAT 1 THRU 3 UNTIL ALL BITS ARE TESTED

                                ANY ERRORS DETECTED ARE REPORTED AND THEN THE TEST RESUMES WITH THE
                                NEXT BIT IN SEQUENCE.

                                ERRORS:
                                ********

                                1.) [ERROR  6] IS CALLED TO REPOORT BOTH FAIL TO SET PROPERLY AND
                                    FAIL TO CLEAR PROPERLY FAULTS.

                                SYNC:
                                *****

                                1.) FAIL TO SET:  M7277 SH4     LOAD LPR H      EP2
                                2.) FAIL TO CLR:  M7277 SH4     LOAD BCR H      FU1

                                DEBUG:
                                ******

                                1.) THE ONLY DIFFERENCES BETWEEN THEE DATA PATHS USED BY THIS TEST AND
                                    THAT USED BY THE PREVIOUS TESTS ARE THE ACTUAL "SSR" REGISTER
                                    CHIPS AND THE INPUT PIN SELECTED ON THE OUTPUT DATA MULTIPLEXORS.
                                2.) IF ALL BITS FAIL IT IS MOST LIKELY THE M7277
                                3.) IF BITS <13:08> FAIL IT IS MOST LIKELY THE M7279
                                4.) IF JUST ONE OR TWO BITS FAIL IT IS MOST LIKELY THE M7278

                                KEY LOGIC:
                                ***********

                                        M7277   SH4     LOAD SSR LOW BYTE H     CR1
                                                        LOAD SSR HIGH BYTE H    CP2
                                                        DATA TO BUS H           EN2
                                                        DATA SOURCE (A,B,C) H   DU1,DU2,DT2

                                        M7279   SH2     SSR <13:08> H  (E20 AND E24)

                                        M7278   SH5 - SH8       REGISTER CHIPS E53,E68, OR E69 (74175'S)
                                                                OUTPUT MUX CHIPS - (74151'S  PIN 12)
                                %
```

```
004012  012767  004046  175070          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
004020  010102                          MOV     R1,R2           ;GENERATE "SSR" ADDRESS IN R2
004022  062702  000016                  ADD     #SSR,R2
004026  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MARKER
004032  030567  023556          1$:     BIT     R5,RGMSK5       ;TEST THIS BIT ??
004036  001003                          BNE     3$              ;BR IF YES
004040  006305                  2$:     ASL     R5              ;SHIFT BIT MARKER
004042  001435                          BEQ     TST11           ;;BR IF ALL BITS TESTED
004044  000772                          BR      1$              ;GO TEST THE BIT

004046  010504                  3$:     MOV     R5,R4           ;SET UP S/B DATA
004050  005012                          CLR     (R2)            ;INIT REG BEING TESTED
004052  012761  000000  000004          MOV     #0,LPR(R1)      ;SCOPE SYNC
004060  050512                          BIS     R5,(R2)         ;SET THE SELECTED BIT IN "SSR"
004062  011203                          MOV     (R2),R3         ;GET THE WAS DATA
004064  042703  077700                  BIC     #77700,R3       ;CLEAR OUT DON'T CARE BITS
004070  020304                          CMP     R3,R4           ;DID BIT SET OK
004072  001403                          BEQ     4$              ;BR IF YES

004074  004767  020302                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
004100  104006                          ERROR   6               ;SSR BIT FAILED TO SET PROPERLY

004102  005004                  4$:     CLR     R4              ;SET UP S/B DATA
004104  012761  000000  000014          MOV     #0,BKR(R1)      ;SCOPE SYNC
004112  040512                          BIC     R5,(R2)         ;CLEAR SSR BIT
004114  011203                          MOV     (R2),R3         ;GET THE SSR WAS DATA
004116  042703  077700                  BIC     #77700,R3       ;CLEAR JUNK BITS
004122  020304                          CMP     R3,R4           ;DID THE SSR BIT GET CLEARED ??
004124  001745                          BEQ     2$              ;BR IF SSR BIT CLEARED OK

004126  004767  020250                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
004132  104006                          ERROR   6               ;SSR BIT FAILED TO CLR PROPERLY
004134  000741                          BR      2$              ;GO SELECT NEXT BIT
```

```
                                ;;*****************************************************************
                                ;;*TEST 11        TEST THAT CLR/SET OF BIT "N" IN "LPR" DOES NOT CLEAR ANY OTHER BITS
                                ;;*****************************************************************
                                ;;
004136  000004                  TST11:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST VERIFIES THAT SETTING AND CLEARING EACH R/W BIT IN
                                THE "LPR" REGISTER DOES NOT DISTURB (CLEAR) ANY OTHER BIT IN THE REGISTER.
                                A BIT MASK (RGMSK3:  177767) IS USED TO DEFINE THE R/W BITS (ALL BUT BIT 03).
                                R5 ALWAYS CONTAINS THE BIT CURRENTLY SELECTED FOR TEST. THE TEST SEQUENCE
                                IS AS FOLLOWS:

                                        1. SELECT A BIT TO TEST
                                        2. SET ALL THE WRITABLE BITS
                                        3. CLEAR THE SELECTED BIT - VERIFY IT CLEARED PROPERLY
                                        4. SET THE SELECTED BIT - VERIFY IT SET PROPERLY
                                        5. REPEAT 1 THRU 4 UNTIL ALL BITS ARE TESTED

                                ANY ERRORS DETECTED ARE REPORTED AND THEN THE TEST RESUMES WITH THE
                                NEXT BIT IN SEQUENCE .

                                ERRORS:
                                *******

                                1.) [ERROR  4] IS CALLED TO REPORT BOTH FAIL TO CLEAR PROPERLY AND
                                    FAIL TO SET PROPERLY FAULTS.

                                SYNC:
                                *******

                                1.) FAIL TO CLR:        M7277   LOAD SSR LOW BYTE H     CR1
                                2.) FAIL TO SET:        M7277   LOAD SSR HIGH BYTE H    CP2

                                DEBUG:
                                ******

                                1.) PROBLEMS DETECTED BY THIS TEST INDICATE ADJACENT BIT INTERFERENCE
                                    CAUSED BY CROSS TALK OR NOISE. PROBLEM IS MOST LIKELY THE M7278.

                                KEY LOGIC:      (SAME AS FOR TEST 6)
                                ***********
                                %
004140  012767  004174  174742          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
004146  010102                          MOV     R1,R2           ;SET UP THE REG ADDR
004150  062702  000004                  ADD     #LPR,R2
004154  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MASK
004160  030567  023424          1$:     BIT     R5,RGMSK3       ;TEST THIS BIT ??
004164  001003                          BNE     3$              ;BR IF YES
004166  006305                  2$:     ASL     R5              ;SHIFT THE BIT TEST MASK
004170  001436                          BEQ     TST12           ;;BR IF TESTED ALL BITS
004172  000772                          BR      1$              ;GO TEST THIS BIT

004174  016704  023410          3$:     MOV     RGMSK3,R4       ;SET UP S/B DATA
004200  005012                          CLR     (R2)            ;INIT REG BEING TESTED
```

```
3664  004202  112761  000000  000016          MOVB    #0,SSR(R1)        ;SCOPE SYNC
3665  004210  040504                          BIC     R5,R4             ;CLR BIT "N"
3666  004212  016712  023372                  MOV     RGMSK3,(R2)       ;SET ALL R/W BITS IN LPR
3667  004216  040512                          BIC     R5,(R2)           ;CLEAR BIT "N" IN LPR
3668  004220  011203                          MOV     (R2),R3           ;GET THE WAS DATA
3669  004222  020304                          CMP     R3,R4             ;DID IT CLEAR OK ?
3670  004224  001404                          BEQ     4$                ;BR IF YES
3671
3672  004226  004767  020150                  JSR     PC,SUER2          ;GO SET UP ERROR INFO
3673  004232  104004                          ERROR   4                 ;BIT "N" FAILED TO CLR PROPERLY
3674  004234  000754                          BR      2$                ;GO TEST NEXT BIT
3675
3676  004236  050504                  4$:     BIS     R5,R4             ;SET  .T "N" IN S/B DATA
3677  004240  112761  000000  000017          MOVB    #0,SSR+1(R1)      ;SCOPE SYNC
3678  004246  050512                          BIS     R5,(R2)           ;SET BIT "N" IN LPR
3679  004250  011203                          MOV     (R2),R3           ;GET THE WAS DATA
3680  004252  020304                          CMP     R3,R4             ;DID BIT "N" SET PROPERLY ?
3681  004254  001744                          BEQ     2$                ;BR IF YES
3682
3683  004256  004767  020120                  JSR     PC,SUER2          ;GO SET UP ERROR INFO
3684  004262  104004                          ERROR   4                 ;BIT "N" FAILED TO SET PROPERLY
3685  004264  000740                          BR      2$                ;GO SELECT NEXT BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 12        TEST THAT CLR/SET OF BIT "N" IN "BKR" DOES NOT CLEAR ANY OTHER BITS
                                ;;*****************************************************************
004266  000004                  TST12:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST VERIFIES THAT CLEARING AND SETTING EACH R/W
                                BIT IN THE BREAK CONTROL REGISTER INDIVIDUALLY DOES NOT DISTURB ANY
                                OF THE OTHER BITS. A BIT MASK (RGMSK4:  177777) IS USED TO DEFINE
                                THE R/W BITS (ALL 16.). R5 ALWAYS CONTAINS THE BIT CURRENTLY SELECTED
                                FOR TEST. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SELECT A BIT TO TEST
                                        2. SET ALL WRITABLE BITS IN THE "BKR"
                                        3. CLEAR THE SELECTED BIT AND VERIFY THAT IT CLEARED PROPERLY
                                        4. SET THE SELECTED BIT AND VERIFY THAT IT SET PROPERLY
                                        5. REPEAT 1 THRU 4 UNTIL ALL BITS HAVE BEEN TESTED

                                ANY ERROR DETECTED IS REORTRD AND THEN THE TEST RESUMES WITH THE
                                NEXT BIT IN SEQUENCE.

                                ERRORS:
                                ********

                                1.) [ERROR  5] IS CALLED TO REPORT BOTH CLEAR AND SET FAULTS.

                                SYNC:
                                *****

                                1.) FAIL TO CLR:        M7277   SH4     LOAD SSR LOW BYTE H     CR1
                                2.) FAIL TO SET:        M7277   SH4     LOAD SSR HIGH BYTE H    CP2

                                DEBUG:
                                ******

                                1.) LIKE THE PREVIOUS TEST, FAILURES HERE INDICATE ADJACENT BIT
                                    INTERFERENCE CAUSED BY CROSS TALK OR NOISE. THE FAULT IS MOST
                                    LIKELY THE M7278 MODULE.

                                KEY LOGIC:      (SAME AS FOR TEST 7)
                                **********
                                %
004270  012767  004324  174612          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
004276  010102                          MOV     R1,R2           ;SET UP THE REG ADDR
004300  062702  000014                  ADD     #BKR,R2
004304  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MASK
004310  030567  023276          1$:     BIT     R5,RGMSK4       ;TEST THIS BIT ??
004314  001003                          BNE     3$              ;BR IF YES
004316  006305                  2$:     ASL     R5              ;SHIFT THE BIT TEST MASK
004320  001436                          BEQ     TST13           ;;BR IF TESTED ALL BITS
004322  000772                          BR      1$              ;GO TEST THIS BIT

004324  016704  023262          3$:     MOV     RGMSK4,R4       ;SET UP S/B DATA
004330  040504                          BIC     R5,R4           ;CLR BIT "N"
```

```
004332  005012                          CLR     (R2)            ;INIT REG BEING TESTED
004334  016712  023252                  MOV     RGMSK4,(R2)     ;SET ALL R/W BITS IN BKR
004340  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
004346  040512                          BIC     R5,(R2)         ;CLEAR BIT "N" IN BKR
004350  011203                          MOV     (R2),R3         ;GET THE WAS DATA
004352  020304                          CMP     R3,R4           ;DID IT CLEAR OK ?
004354  001404                          BEQ     4$              ;BR IF YES

004356  004767  020020                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
004362  104005                          ERROR   5               ;BIT "N" FAILED TO CLR PROPERLY
004364  000754                          BR      2$              ;GO TEST NEXT BIT

004366  050504                  4$:     BIS     R5,R4           ;SET BIT "N" IN S/B DATA
004370  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
004376  050512                          BIS     R5,(R2)         ;SET BIT "N" IN BKR
004400  011203                          MOV     (R2),R3         ;GET THE WAS DATA
004402  020304                          CMP     R3,R4           ;DID BIT "N" SET PROPERLY ?
004404  001744                          BEQ     2$              ;BR IF YES

004406  004767  017770                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
004412  104005                          ERROR   5               ;BIT "N" FAILED TO SET PROPERLY
004414  000740                          BR      2$              ;GO SELECT NEXT BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 13       TEST THAT CLR/SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER BITS
                                ;;*****************************************************************
004416  000004                  TST13:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST VERIFIES THAT CLEARING AND SETTING EACH R/W
                                BIT IN THE SILO STATUS REGISTER INDIVIDUALLY DOES NOT DISTURB ANY
                                OF THE OTHER BITS. A BIT MASK (RGMSK5:  100077) IS USED TO DEFINE
                                THE R/W BITS (15,5,4,3,2,1, AND 0). R5 ALWAYS CONTAINS THE BIT CURRENTLY SELECTED
                                FOR TEST. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SELECT A BIT TO TEST
                                        2. SET ALL WRITABLE BITS IN THE "SSR"
                                        3. CLEAR THE SELECTED BIT AND VERIFY THAT IT CLEARED PROPERLY
                                        4. SET THE SELECTED BIT AND VERIFY THAT IT SET PROPERLY
                                        5. REPEAT 1 THRU 4 UNTIL ALL BITS HAVE BEEN TESTED

                                ANY ERROR DETECTED IS REORTRD AND THEN THE TEST RESUMES WITH THE
                                NEXT BIT IN SEQUENCE.

                                ERRORS:
                                ********

                                1.) [ERROR  6] IS CALLED TO REPORT BOTH CLEAR AND SET FAULTS.

                                SYNC:
                                *****

                                1.) FAIL TO CLR:        M7277   SH4     LOAD LPR H      EP2
                                2.) FAIL TO SET:        M7277   SH4     LOAD BCR H      FU1

                                DEBUG:
                                ******

                                1.) LIKE THE PREVIOUS TEST, FAILURES HERE INDICATE ADJACENT BIT
                                    INTERFERENCE CAUSED BY CROSS TALK OR NOISE. THE FAULT IS MOST
                                    LIKELY THE M7278 MODULE.

                                KEY LOGIC:      (SAME AS FOR TEST 10)
                                **********
                                %
004420  012767  004454  174462          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
004426  010102                          MOV     R1,R2           ;SET UP THE REG ADDR
004430  062702  000016                  ADD     #SSR,R2
004434  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MASK
004440  030567  023150          1$:     BIT     R5,RGMSK5       ;TEST THIS BIT ??
004444  001003                          BNE     3$              ;BR IF YES
004446  006305                  2$:     ASL     R5              ;SHIFT THE BIT TEST MASK
004450  001442                          BEQ     TST14           ;;BR IF TESTED ALL BITS
004452  000772                          BR      1$              ;GO TEST THIS BIT

004454  016704  023134          3$:     MOV     RGMSK5,R4       ;SET UP S/B DATA
004460  040504                          BIC     R5,R4           ;CLR BIT "N"
```

```
004462  005012                          CLR     (R2)            ;INIT REG BEING TESTED
004464  016712  023124                  MOV     RGMSK5,(R2)     ;SET ALL R/W BITS IN SSR
004470  012761  000000  000004          MOV     #0,LPR(R1)      ;SCOPE SYNC
004476  040512                          BIC     R5,(R2)         ;CLEAR BIT "N" IN SSR
004500  011203                          MOV     (R2),R3         ;GET THE WAS DATA
004502  042703  077700                  BIC     #77700,R3       ;CLEAR JUNK BITS
004506  020304                          CMP     R3,R4           ;DID IT CLEAR OK ?
004510  001404                          BEQ     4$              ;BR IF YES

004512  004767  017664                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
004516  104006                          ERROR   6               ;BIT "N" FAILED TO CLR PROPERLY
004520  000752                          BR      2$              ;GO TEST NEXT BIT

004522  050504                  4$:     BIS     R5,R4           ;SET BIT "N" IN S/B DATA
004524  012761  000000  000014          MOV     #0,BKR(R1)      ;SCOPE SYNC
004532  050512                          BIS     R5,(R2)         ;SET BIT "N" IN SSR
004534  011203                          MOV     (R2),R3         ;GET THE WAS DATA
004536  042703  077700                  BIC     #77700,R3       ;CLEAR JUNK BITS
004542  020304                          CMP     R3,R4           ;DID BIT "N" SET PROPERLY ?
004544  001740                          BEQ     2$              ;BR IF YES

004546  004767  017630                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
004552  104006                          ERROR   6               ;BIT "N" FAILED TO SET PROPERLY
004554  000734                          BR      2$              ;GO SELECT NEXT BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 14        "CAR" MEMORY ADDRESSING TEST
                                ;;*****************************************************************
004556  000004                  TST14:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST VERIFIES THAT EACH LOCATION IN THE CURRENT ADDRESS
                                MEMORY CAN BE UNIQUELY ADDRESSED. IT WRITES THE PATTERN SHOWN BELOW
                                INTO THE MEMORY AND THEN READS BACK EACH LOCATION TO VERIFY THAT IT
                                WAS WRITTEN CORRECTLY. SINCE THE MEMORY LOGIC IS PARTITIONED INTO
                                FOUR 16 X 4 READ/WRITE MEMORY CHIPS, THE PATTERN RESULTS IN THE LINE
                                NUMBER (00 - 17(8)) BEING WRITTEN AS DATA INTO TO EACH LOCATION
                                (00 - 17(8)) IN EACH CHIP. THAT IS:  LOC00 = 00, LOC 01 = 01,
                                .......LOC 17 = 17.


                                        MEMORY PATTERN: LOCATION        CONTENTS        (BOTH OCTAL)

                                                        00              000000
                                                        01              010421
                                                        02              021042
                                                        03              031463
                                                        04              042104
                                                        05              052525
                                                        06              063146
                                                        07              073567
                                                        10              104210
                                                                11              114631
                                                        12              125252
                                                        13              135673
                                                        14              146314
                                                        15              156735
                                                        16              167356
                                                        17              177777

                                ANY ERRORS DETECTED ARE REPORTED AND THEN THE TEST RESUMES CHECKING
                                THEE NEXT LOCATION IN SEQUENCE UNTIL ALL 16. HAVE BEEN CHECKED.

                                        NOTE:   THIS TEST ALWAYS CHECKS ALL 16. LINES REGARDLESS
                                                OF HOW THE LINE SELECTION PARAMETER WAS INITIALLY
                                                SET UP.

                                ERRORS:
                                ********

                                1.) [ERROR  7] IS CALLED TO REPORT ANY LINES (LOCATIONS) THAT FAIL.
                                    THE FAILING LINE # IS INCLUDED AS PART OF THE ERROR HEADER MESSAGE.

                                SYNC:
                                ******

                                1.) WRITE SYNC: M7277   SH4     LOAD SSR LOW BYTE H     CR1
                                2.) READ SYNC:  M7277   SH4     LOAD SSR HIGH BYTE H    CP2
```

```
                                        DEBUG:
                                        ******

                                        1.) ANALYZE THE ERROR REPORTS CAREFULLY ASKING THE FOLLOWING QUESTIONS:

                                                A. DOES THE FAULT AFFECT ONLY ONE LINE ?
                                                B. DOES THE FAULT AFFECT ONLY ONE 4-BIT DATA GROUP ?
                                                   IE <15:12>, <11:08>, <07:04>, OR <03:00>
                                                C. DOES ANY DATA AT ALL APPEAR TO BE WRITTEN ?

                                        2.) IF "A" IS TRUE THEN SUSPECT AN ADDRESSING PROBLEM IN THE MEMORY
                                            ADDRESS MUX.

                                        3.) IF "B" IS TRUE THEN SUSPECT A DATA MUX, UP-COUNTER, MEMORY,
                                            OR INVERTER CHIP PROBLEM.

                                        4.) IF "C" IS TRUE SUSPECT A MEMORY WRITE TIMING PROBLEM.

                                        5.) IN MOST CASES THE FAULT IS MOST LIKELY THE M7277 OR M7278.

                                        KEY LOGIC:
                                        **********

                                                M7277   SH4     LOAD CA H               E58-13
                                                                DATA TO BUS H           EN2
                                                                DATA SOURCE (A,B,C)     DU1,DU2,DT2

                                                        SH5     MEMADD SOURCE SEL H     E55-8
                                                                CA MEM WRITE ENAB L     E50-1
                                                                BUF ADDRS TO BUS H      E33-1 (SHD BE LOW)
                                                                74157 MUX CHIPS E33,E27,E20 BITS<17:08>
                                                                74193 COUNTER CHIPS E19,E26,E32 BITS<17:08>
                                                                7489 MEMORY CHIPS E18,E25,E31 BITS<17:08>
                                                                7404 INVERTER CHIPS E30,E24,E17 BITS<17:08>

                                                        SH5     74157 MUX CHIP  E48
                                                                74157 DATA MUX CHIPS E13,E06 BITS<07:00>
                                                                74193 COUNTER CHIPS E12,E05 BITS<07:00>
                                                                7489 MEMORY CHIPS E11,E04 BITS<07:00>


                                                M7278   SH5 THRU SH8   74151 DATA MUX OUTPUT CHIPS (PIN 1 INPUT)
                                        %
004560  010102                          MOV     R1,R2           ;COPY IT IN R2
004562  062702  000006                  ADD     #CAR,R2         ;SET UP REGADR IN R2
004566  016767  022442  174424          MOV     LINSEL,$TMP7    ;SAVE LINE SELECT PARAMETER
004574  012767  177777  022432          MOV     #-1,LINSEL      ;DO ALL LINES FOR THIS TEST
004602  004767  017726          1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
004606  000415                          BR      3$              ;;BR IF DONE ALL SELECTED LINES
004610  105767  023430                  TSTB    LINE            ;DOING LINE 00 ?
004614  001001                          BNE     2$              ;BR IF NOT
004616  005004                          CLR     R4              ;INIT TEST DATA

004620  156711  023420          2$:     BISB    LINE,(R1)       ;SELECT A LINE
004624  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
004632  010412                          MOV     R4,(R2)         ;LOAD THE CAR REG.
```

```
004634  062704  010421                  ADD     #10421,R4       ;GENERATE NEW DATA
004640  000760                          BR      1$              ;GO DO NEXT LINE

004642  004767  017666          3$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
004646  000434                          BR      7$              ;;BR IF CHECKED ALL LINES
004650  105767  023370                  TSTB    LINE            ;DOING LINE 00 ?
004654  001001                          BNE     4$              ;BR IF NOT
004656  005004                          CLR     R4              ;INIT S/B DATA

004660  156711  023360          4$:     BISB    LINE,(R1)       ;SELECT A LINE
004664  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
004672  011203                          MOV     (R2),R3         ;GET CONTENTS OF CAR
004674  020304                          CMP     R3,R4           ;WAS DATA OK ?
004676  001412                          BEQ     5$              ;BR IF YES

004700  004767  017476                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
004704  004567  017716                  JSR     R5,SUNUM        ;SET UP LINE NO. IN MSG BUFFER
004710  030244                          LINE
004712  031077                          EM7+47
004714  012767  004732  174166          MOV     #6$,$LPERR      ;SET UP ERROR LOOP RETURN
004722  104007                          ERROR   7               ;CAR ADDRESSING ERROR

004724  062704  010421          5$:     ADD     #10421,R4       ;GENERATE NEW S/B DATA
004730  000744                          BR      3$              ;GO CHECK NEXT LINE

004732  005067  023306          6$:     CLR     LINE            ;RESTART AT LINE 00 IF LOOPING
004736  000721                          BR      1$              ;GO RESTART

004740  016767  174254  022266  7$:     MOV     $TMP7,LINSEL    ;RESTORE LINE SELECT PARAMETER
```

```
                                ;;*****************************************************************
                                ;*TEST 15        "BCR" MEMORY ADDRESSING TEST
                                ;;*****************************************************************
004746  000004                  TST15:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST VERIFIES THAT EACH LOCATION IN THE BYTE COUNT
                                MEMORY CAN BE UNIQUELY ADDRESSED. IT WRITES THE PATTERN SHOWN BELOW
                                INTO THE MEMORY AND THEN READS BACK EACH LOCATION TO VERIFY THAT IT
                                WAS WRITTEN CORRECTLY. SINCE THE MEMORY LOGIC IS PARTITIONED INTO
                                FOUR 16 X 4 READ/WRITE MEMORY CHIPS, THE PATTERN RESULTS IN THE LINE
                                NUMBER (00 - 17(8)) BEING WRITTEN AS DATA INTO TO EACH LOCATION
                                (00 - 17(8)) IN EACH CHIP. THAT IS:  LOC00 = 00, LOC 01 = 01,
                                .......LOC 17 = 17.


                                        MEMORY PATTERN: LOCATION        CONTENTS        (BOTH OCTAL)

                                                        00              000000
                                                        01              010421
                                                        02              021042
                                                        03              031463
                                                        04              042104
                                                        05              052525
                                                        06              063146
                                                        07              073567
                                                        10              104210
                                                        11              114631
                                                        12              125252
                                                        13              135673
                                                        14              146314
                                                        15              156735
                                                        16              167356
                                                        17              177777

                                ANY ERRORS DETECTED ARE REPORTED AND THEN THE TEST RESUMES CHECKING
                                THEE NEXT LOCATION IN SEQUENCE UNTIL ALL 16. HAVE BEEN CHECKED.

                                        NOTE:   THIS TEST ALWAYS CHECKS ALL 16. LINES REGARDLESS
                                                OF HOW THE LINE SELECTION PARAMETER WAS INITIALLY
                                                SET UP.

                                ERRORS:
                                ********

                                1.) [ERROR  10] IS CALLED TO REPORT ANY LINES (LOCATIONS) THAT FAIL.
                                    THE FAILING LINE # IS INCLUDED AS PART OF THE ERROR HEADER MESSAGE.

                                SYNC:
                                ******

                                1.) WRITE SYNC: M7277   SH4     LOAD SSR LOW BYTE H     CR1
                                2.) READ SYNC:  M7277   SH4     LOAD SSR HIGH BYTE H    CP2
```

```
                                        DEBUG:
                                        ******

                                        1.) ANALYZE THE ERROR REPORTS CAREFULLY ASKING THE FOLLOWING QUESTIONS:

                                                A. DOES THE FAULT AFFECT ONLY ONE LINE ?
                                                B. DOES THE FAULT AFFECT ONLY ONE 4-BIT DATA GROUP ?
                                                   IE <15:12>, <11:08>, <07:04>, OR <03:00>
                                                C. DOES ANY DATA AT ALL APPEAR TO BE WRITTEN ?

                                        2.) IF "A" IS TRUE THEN SUSPECT AN ADDRESSING PROBLEM IN THE MEMORY
                                            ADDRESS MUX.

                                        3.) IF "B" IS TRUE THEN SUSPECT A DATA MUX, UP-COUNTER, MEMORY,
                                            OR INVERTER CHIP PROBLEM.

                                        4.) IF "C" IS TRUE SUSPECT A MEMORY WRITE TIMING PROBLEM.

                                        5.) IN MOST CASES THE FAULT IS MOST LIKELY THE M7277 OR M7278.

                                        KEY LOGIC:
                                        **********

                                                M7277   SH4     LOAD BC H               FU2
                                                                DATA TO BUS H           EN2
                                                                DATA SOURCE (A,B,C)     DU1,DU2,DT2

                                                        SH5     MEMADD SOURCE SEL H     E55-8
                                                                BC MEM WRITE ENAB L     E57-4
                                                                BUF ADDRS TO BUS H      E33-1 (SHD BE LOW)
                                                M7278   SH3     BITS<15:08>
                                                                74157 INPUT MUX CHIPS E18,E19
                                                                74193 UP COUNTER CHIPS E27,E26
                                                                7489 MEMORY CHIPS E33,E34
                                                                7404 INVERTER CHIPS E41,E42

                                                        SH4     BITS<07:00>
                                                                74157 INPUT MUX CHIPS E16,E17
                                                                74193 UP COUNTER CHIPS E24,E25
                                                                7489 MEMORY CHIPS E31,E32
                                                                7404 INVERTER CHIPS E39,E40

                                                        SH5 THRU SH8    74151 DATA MUX OUTPUT CHIPS (PIN 1 INPUT)
                                        %
004750  010102                          MOV     R1,R2           ;COPY IT IN R2
004752  062702  000010                  ADD     #BCR,R2         ;SET UP REGADR IN R2
004756  016767  022252  174234          MOV     LINSEL,$TMP7    ;SAVE LINE SELECT PARAMETER
004764  012767  177777  022242          MOV     #-1,LINSEL      ;DO ALL LINES FOR THIS TEST
004772  004767  017536          1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
004776  000415                          BR      3$              ;;BR IF DONE ALL SELECTED LINES
005000  105767  023240                  TSTB    LINE            ;DOING LINE 00 ?
005004  001001                          BNE     2$              ;BR IF NOT
005006  005004                          CLR     R4              ;INIT TEST DATA

005010  156711  023230          2$:     BISB    LINE,(R1)       ;SELECT A LINE
005014  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
```

```
4097  005022  010412                          MOV     R4,(R2)          ;LOAD THE BCR REG.
4098  005024  062704  010421                  ADD     #10421,R4        ;GENERATE NEW DATA
4099  005030  000760                          BR      1$               ;GO DO NEXT LINE
4100
4101  005032  004767  017476          3$:     JSR     PC,SELINE        ;GO SELECT A LINE NO.
4102  005036  000434                          BR      7$               ;;BR IF CHECKED ALL LINES
4103  005040  105767  023200                  TSTB    LINE             ;DOING LINE 00 ?
4104  005044  001001                          BNE     4$               ;BR IF NOT
4105  005046  005004                          CLR     R4               ;INIT S/B DATA
4106
4107  005050  156711  023170          4$:     BISB    LINE,(R1)        ;SELECT A LINE
4108  005054  112761  000000  000017          MOVB    #0,SSR+1(R1)     ;SCOPE SYNC
4109  005062  011203                          MOV     (R2),R3          ;GET CONTENTS OF BCR
4110  005064  020304                          CMP     R3,R4            ;WAS DATA OK ?
4111  005066  001412                          BEQ     5$               ;BR IF YES
4112
4113  005070  004767  017306                  JSR     PC,SUER2         ;GO SET UP ERROR INFO
4114  005074  004567  017526                  JSR     R5,SUNUM         ;GO SET UP LINE NO. IN MSG BUFFER
4115  005100  030244                          LINE
4116  005102  031146                          EM10+44
4117  005104  012767  005122  173776          MOV     #6$,$LPERR       ;SET UP ERROR LOOP RETURN
4118  005112  104010                          ERROR   10               ;BCR ADDRESSING ERROR
4119
4120  005114  062704  010421          5$:     ADD     #10421,R4        ;GENERATE NEW S/B DATA
4121  005120  000744                          BR      3$               ;GO CHECK NEXT LINE
4122
4123  005122  005067  023116          6$:     CLR     LINE             ;RESTART AT LINE 00 IF LOOPING
4124  005126  000721                          BR      1$               ;GO RESTART
4125
4126  005130  016767  174064  022076  7$:     MOV     $TMP7,LINSEL     ;RESTORE THE LINE SELECT PARAMETER
```

```
                                ;;*****************************************************************
                                ;;*TEST 16        "CAR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
                                ;;*****************************************************************
                                ;;
005136  000004                  TST16:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THE ABILITY TO SET AND CLEAR ALL BITS IN
                                ALL THE SELECTED LOCATIONS (LINES) OF THE CURRENT ADDRESS MEMORY.
                                IT USES THE CONFIGURATION PARAMETER (LINSEL:) TO DEFINE WHICH LINES TO
                                TEST. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SELECT A LINE # TO TEST
                                        2. LOAD THE SELECTED LOCATION WITH 177777
                                        3. READ IT BACK TO VERIFY ALL BITS SET
                                        4. LOAD THE SELECTED LOCATION WITH 000000
                                        5. READ IT BACK TO VERIFY ALL BITS CLEARED
                                        6. REPEAT STEPS 1 THRU 5 UNTIL ALL SELECTED LINES ARE TESTED.

                                ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH THE NEXT
                                LINE # IN SEQUENCE AS DEFINED BY "LINSEL".

                                ERRORS:
                                ********

                                1.) [ERROR  7] IS CALLED TO REPORT ALL DATA COMPARE ERRORS

                                SYNC:
                                *****

                                1.) WRITE 1'S:  M7277   SH4     LOAD SSR LOW BYTE H     CR1

                                2.) WRITE 0'S:  M7277   SH4     LOAD SSR HIGH BYTE H    CP2

                                DEBUG:  (REFER TO TEST 14)
                                ******

                                KEY LOGIC:      (REFER TO TEST 14)
                                **********

                                %
005140  010102                          MOV     R1,R2           ;COPY IT INTO R2
005142  062702  000006                  ADD     #CAR,R2         ;R2 GETS CAR ADDRESS
005146  004767  017362          1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
005152  000443                          BR      TST17           ;;BR IF DONE ALL SELECTED LINES
005154  012704  177777                  MOV     #-1,R4          ;RESULT IN CAR S/B = 177777
005160  156711  023060                  BISB    LINE,(R1)       ;SELECT A LINE NO.
005164  004567  017436                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
005170  030244                          LINE
005172  031077                          EM7+47

005174  112761  000000  000016  2$:     MOVB    #0,SSR(R1)      ;SCOPE SYNC
005202  010412                          MOV     R4,(R2)         ;LOAD A CAR WITH ALL ONES
005204  011203                          MOV     (R2),R3         ;GET THE WAS DATA FROM THE CAR
005206  020403                          CMP     R4,R3           ;DID IT CONTAIN ALL ONES ??
```

```
005210  001406                          BEQ     3$              ;;BR IF ALL 1'S

005212  004767  017164                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
005216  012767  005174  173664          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
005224  104007                          ERROR   7               ;FAILED TO SET ALL 1'S IN SELECTED CAR

005226  005004                  3$:     CLR     R4              ;RESULT IN CAR S/B = 000000
005230  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
005236  010412                          MOV     R4,(R2)         ;CLEAR SELECTED CAR
005240  011203                          MOV     (R2),R3         ;GET THE WAS DATA
005242  001741                          BEQ     1$              ;BR IF CAR GOT CLEARED

005244  004767  017132                  JSR     PC,SUER2        ;GO SET UP FOR ERROR CALL
005250  012767  005226  173632          MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN
005256  104007                          ERROR   7               ;FAILED TO CLR ALL BITS IN SELECTED CAR
005260  000732                          BR      1$              ;GO TEST NEXT LINE
```

```
                                ;;*****************************************************************
                                ;*TEST 17        "BCR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
                                ;;*****************************************************************
                                ;;
005262  000004                  TST17:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THE ABILITY TO SET AND CLEAR ALL BITS IN
                                ALL THE SELECTED LOCATIONS (LINES) OF THE BYTE COUNT MEMORY.
                                IT USES THE CONFIGURATION PARAMETER (LINSEL:) TO DEFINE WHICH LINES TO
                                TEST. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SELECT A LINE # TO TEST
                                        2. LOAD THE SELECTED LOCATION WITH 177777
                                        3. READ IT BACK TO VERIFY ALL BITS SET
                                        4. LOAD THE SELECTED LOCATION WITH 000000
                                        5. READ IT BACK TO VERIFY ALL BITS CLEARED
                                        6. REPEAT STEPS 1 THRU 5 UNTIL ALL SELECTED LINES ARE TESTED.

                                ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH THE NEXT
                                LINE # IN SEQUENCE AS DEFINED BY "LINSEL".

                                ERRORS:
                                ********

                                1.) [ERROR  10] IS CALLED TO REPORT ALL DATA COMPARE ERRORS

                                SYNC:
                                *****

                                1.) WRITE 1'S:  M7277   SH4     LOAD SSR LOW BYTE H     CR1

                                2.) WRITE 0'S:  M7277   SH4     LOAD SSR HIGH BYTE H    CP2

                                DEBUG:  (REFER TO TEST 15)
                                ******

                                KEY LOGIC:      (REFER TO TEST 15)
                                **********

                                %
005264  010102                          MOV     R1,R2           ;COPY IT INTO R2
005266  062702  000010                  ADD     #BCR,R2         ;R2 GETS BCR ADDRESS
005272  004767  017236          1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
005276  000443                          BR      TST20           ;;BR IF DONE ALL SELECTED LINES
005300  012704  177777                  MOV     #-1,R4          ;RESULT IN BCR S/B = 177777
005304  156711  022734                  BISB    LINE,(R1)       ;SELECT A LINE NO.
005310  004567  017312                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
005314  030244                          LINE
005316  031146                          EM10+44

005320  112761  000000  000016  2$:     MOVB    #0,SSR(R1)      ;SCOPE SYNC
005326  010412                          MOV     R4,(R2)         ;LOAD A BCR WITH ALL ONES
005330  011203                          MOV     (R2),R3         ;GET THE WAS DATA FROM THE BCR
005332  020403                          CMP     R4,R3           ;DID IT CONTAIN ALL ONES ??
```

```
005334  001406                          BEQ     3$              ;;BR IF ALL 1'S

005336  004767  017040                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
005342  012767  005320  173540          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
005350  104010                          ERROR   10              ;FAILED TO SET ALL 1'S IN SELECTED BCR

005352  005004                  3$:     CLR     R4              ;RESULT IN BCR S/B = 000000
005354  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
005362  010412                          MOV     R4,(R2)         ;CLEAR SELECTED BCR
005364  011203                          MOV     (R2),R3         ;GET THE WAS DATA
005366  001741                          BEQ     1$              ;BR IF BCR GOT CLEARED

005370  004767  017006                  JSR     PC,SUER2        ;GO SET UP FOR ERROR CALL
005374  012767  005352  173506          MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN
005402  104010                          ERROR   10              ;FAILED TO CLR ALL BITS IN SELECTED BCR
005404  000732                          BR      1$              ;GO TEST NEXT LINE
```

```
                                        ;;*****************************************************************
                                        ;*TEST 20        "CAR" MEMORY PATTERNS TEST / D'S DISTURB
                                        ;;*****************************************************************
005406  000004                          TST20:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        **************

                                                THIS TEST VERIFIES THAT WHEN A TEST PATTERN IS WRITTEN INTO
                                        LOCATION "N" OF THE "CAR" MEMORY, IT DOES NOT DISTURB ANY BITS IN
                                        ANY OTHER LOCATIONS. THERE ARE THREE TEST PATTERNS USED* (177777,
                                        125252, 052525) FOR EACH LOCATION SELECTED BY THE CONFIGURATION PAR-
                                        AMETER "LINSEL". THE TEST SEQUENCE IS AS FOLLOWS:

                                                1. SELECT A TEST PATTERN
                                                2. SELECT A LINE # TO TEST
                                                3. CLEAR ALL 16. LOCATIONS IN THE MEMORY
                                                4. WRITE THE TEST PATTERN INTO THE SELECTED LOCATION
                                                5. VERIFY THAT THE PATTERN WAS WRITTEN CORRECTLY AND THAT
                                                   NO OTHER LOCATIONS WERE DISTURBED.
                                                6. REPEAT 2 THRU 5 UNTIL ALL SELECTED LINES TESTED
                                                7. REPEAT 1 THRU 6 UNTIL ALL THREE PATTERNS TESTED

                                        ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH CHECKING THE
                                        NEXT LINE IN SEQUENCE.

                                        ERRORS:
                                        *******

                                        1.) [ERROR  46] IS CALLED TO REPORT ANY ERROR DETECTED. THE INFORM-
                                            ATION PRINTED INCLUDES THE LINE # WRITTEN, THE LINE # BEING CHECKED,
                                            AND THE PATTERN USED.

                                        SYNC*
                                        ******

                                        1.) WRITE LINE: M7277   SH4     LOAD SSR LOW BYTE H     CR1

                                        2.) READ CHECK: M7277   SH4     LOAD SSR HIGH BYTE H    CP2

                                        DEBUG:  (REFER TO TEST 14)
                                        ******

                                        KEY LOGIC:      (REFER TO TEST 14)
                                        **********
                                        %
005410  010102                                  MOV     R1,R2           ;SET UP REGADR
005412  062702  000006                          ADD     #CAR,R2
005416  012705  027252                          MOV     #PATRNA,R5      ;SET UP POINTER TO DATA PATTERNS
005422  012567  173556                  1$:     MOV     (R5)+,$TMP1     ;GET A DATA TEST PATTERN
005426  001472                                  BEQ     TST21           ;;BR IF DONE THREE PATTERNS
005430  004767  017100                  11$:    JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
005434  000772                                  BR      1$              ;BR IF DONE ALL SELECTED LINES
005436  116767  022602  022602                  MOVB    LINE,LINEA      ;SAVE THE LINE NO. FOR ERROR LOOPING

005444  105067  173540                  2$:     CLRB    $TMP3           ;INIT LINE COUNTER
```

```
005450  116711  173534          3$:   MOVB   $TMP3,(R1)       ;SELECT A LINE TO CLEAR
005454  005012                        CLR    (R2)             ;CLR CAR FOR THAT LINE
005456  105267  173526                INCB   $TMP3            ;GENERATE NEW LINE NO.
005462  126727  173522  000020        CMPB   $TMP3,#20        ;DONE CLEARING ALL LINES ?
005470  001367                        BNE    3$               ;BR IF NOT

005472  116711  022550                MOVB   LINEA,(R1)       ;SET LINE SELECT BITS
005476  112761  000000  000016        MOVB   #0,SSR(R1)       ;SCOPE SYNC
005504  016712  173474                MOV    $TMP1,(R2)       ;LOAD CAR WITH TEST PATTERN

005510  105067  173472                CLRB   $TMP2            ;INIT A LINE COUNTER
005514  016704  173464          4$:   MOV    $TMP1,R4         ;SET UP S/B DATA
005520  116711  173462                MOVB   $TMP2,(R1)       ;SET LINE SELECT IN SCR
005524  112761  000000  000017        MOVB   #0,SSR+1(R1)     ;SCOPE SYNC
005532  011203                        MOV    (R2),R3          ;GET WAS DATA
005534  126767  173446  022502        CMPB   $TMP2,LINE       ;IS THIS THE LINE WITH THE TEST PATTERN
005542  001401                        BEQ    5$               ;BR IF IT IS
005544  005004                        CLR    R4               ;MAKE S/B DATA = 000000
005546  020304                  5$:   CMP    R3,R4            ;CORRECT DATA IN CAR ?
005550  001412                        BEQ    6$               ;BR IF YES

005552  004767  016712                JSR    PC,SUER4         ;GO SET UP ERROR IN F0
005556  004567  017044                JSR    R5,SUNUM         ;GO SET UP LINE NO. IN MSG BUFFER
005562  001206                        $TMP2
005564  033720                        EM46+63
005566  012767  005444  173314        MOV    #2$,$LPERR       ;SET UP ERROR LOOP RETURN
005574  104046                        ERROR  46               ;INCORRECT DATA READ FROM CAR

005576  105267  173404          6$:   INCB   $TMP2            ;GENERATE NEXT LINE NO.
005602  122767  000020  173376        CMPB   #20,$TMP2        ;DONE ALL LINES ?
005610  001707                        BEQ    11$              ;BR IF YES
005612  000740                        BR     4$               ;GO CHECK NEXT LINE
```

```
                                ;;*****************************************************************
                                ;*TEST 21        "BCR" MEMORY PATTERNS TEST / 0'S DISTURB
                                ;;*****************************************************************
005614  000004                  TST21:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT WHEN A TEST PATTERN IS WRITTEN INTO
                                LOCATION "N" OF THE "BCR" MEMORY, IT DOES NOT DISTURB ANY BITS IN
                                ANY OTHER LOCATIONS. THERE ARE THREE TEST PATTERNS USED* (177777,
                                125252, 052525) FOR EACH LOCATION SELECTED BY THE CONFIGURATION PAR-
                                AMETER "LINSEL". THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SELECT A TEST PATTERN
                                        2. SELECT A LINE # TO TEST
                                        3. CLEAR ALL 16. LOCATIONS IN THE MEMORY
                                        4. WRITE THE TEST PATTERN INTO THE SELECTED LOCATION
                                        5. VERIFY THAT THE PATTERN WAS WRITTEN CORRECTLY AND THAT
                                           NO OTHER LOCATIONS WERE DISTURBED.
                                        6. REPEAT 2 THRU 5 UNTIL ALL SELECTED LINES TESTED
                                        7. REPEAT 1 THRU 6 UNTIL ALL THREE PATTERNS TESTED

                                ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH CHECKING THE
                                NEXT LINE IN SEQUENCE.

                                ERRORS:
                                *******

                                1.) [ERROR  47] IS CALLED TO REPORT ANY ERROR DETECTED. THE INFORM-
                                    ATION PRINTED INCLUDES THE LINE # WRITTEN, THE LINE # BEING CHECKED,
                                    AND THE PATTERN USED.

                                SYNC*
                                ******

                                1.) WRITE LINE: M7277   SH4     LOAD SSR LOW BYTE H     CR1

                                2.) READ CHECK: M7277   SH4     LOAD SSR HIGH BYTE H    CP2

                                DEBUG:  (REFER TO TEST 15)
                                ******

                                KEY LOGIC:      (REFER TO TEST 15)
                                **********
                                %
005616  010102                          MOV     R1,R2           ;SET UP REGADR
005620  062702  000010                  ADD     #BCR,R2
005624  012705  027252                  MOV     #PATRNA,R5      ;SET UP POINTER TO DATA PATTERNS
005630  012567  173350          1$:     MOV     (R5)+,$TMP1     ;GET A DATA TEST PATTERN
005634  001472                          BEQ     TST22           ;;BR IF DONE THREE PATTERNS
005636  004767  016672          11$:    JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
005642  000772                          BR      1$              ;BR IF SELECTED ALL LINES
005644  116767  022374  022374          MOVB    LINE,LINEA      ;SAVE THE LINE NO. FOR ERROR LOOP

005652  105067  173332          2$:     CLRB    $TMP3           ;INIT LINE COUNTER
```

```
005656  116711  173326          3$:     MOVB    $TMP3,(R1)      ;SELECT A LINE TO CLEAR
005662  005012                          CLR     (R2)            ;CLR BCR FOR THAT LINE
005664  105267  173320                  INCB    $TMP3           ;GENERATE NEW LINE NO.
005670  126727  173314  000020          CMPB    $TMP3,#20       ;DONE CLEARING ALL LINES ?
005676  001367                          BNE     3$              ;BR IF NOT

005700  116711  022342                  MOVB    LINEA,(R1)      ;SET LINE SELECT BITS
005704  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
005712  016712  173266                  MOV     $TMP1,(R2)      ;LOAD BCR WITH TEST PATTERN

005716  105067  173264                  CLRB    $TMP2           ;INIT A LINE COUNTER
005722  016704  173256          4$:     MOV     $TMP1,R4        ;SET UP S/B DATA
005726  116711  173254                  MOVB    $TMP2,(R1)      ;SELECT A LINE TO CHECK
005732  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
005740  011203                          MOV     (R2),R3         ;GET WAS DATA
005742  126767  173240  022274          CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
005750  001401                          BEQ     5$              ;BR IF IT IS
005752  005004                          CLR     R4              ;MAKE S/B DATA = 000000
005754  020304          5$:             CMP     R3,R4           ;CORRECT DATA IN BCR ?
005756  001412                          BEQ     6$              ;BR IF YES

005760  004767  016504                  JSR     PC,SUER4        ;GO SET UP ERROR IN F0
005764  004567  016636                  JSR     R5,SUNUM        ;GO SET UP LINE NO.IN MSG BUFFER
005770  001206                          $TMP2
005772  034120                          EM47+56
005774  012767  005652  173106          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
006002  104047                          ERROR   47              ;INCORRECT DATA READ FROM BCR

006004  105267  173176          6$:     INCB    $TMP2           ;GENERATE NEXT LINE NO.
006010  122767  000020  173170          CMPB    #20,$TMP2       ;DONE ALL LINES ?
006016  00  07                          BEQ     11$             ;BR IF YES
006020  000740                          BR      4$              ;GO CHECK NEXT LINE
```

```
                                        ;;*****************************************************************
                                        ;*TEST 22        "CAR" MEMORY PATTERNS TEST / 1'S DISTURB
                                        ;;*****************************************************************
006022  000004                          TST22:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        ***************

                                                THIS TEST VERIFIES THAT WHEN ALL ZEROS ARE WRITTEN INTO LINE "N"
                                        IN THE "CAR" MEMORY, IT DOES NOT CLEAR ANY BITS IN ANY OTHER LOCATIONS.
                                        ONLY THE LINES SELECTED BY "LINSEL" ARE TESTED. THE TEST SEQUENCE IS
                                        AS FOLLOWS:

                                                1. SELECT A LINE TO TEST
                                                2. SET ALL ONES (177777) INTO ALL MEMORY LOCATIONS
                                                3. CLEAR THE SELECTED LINE
                                                4. VERIFY THAT ONLY THE SELECTED LINE WAS CLEARED
                                                   AND ALL OTHER LINES STILL CONTAIN 177777
                                                5. REPEAT STEPS 1 THRU 4 UNTIL ALL SELECTED LINES ARE TESTED

                                        ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES CHECKING THE NEXT
                                        LINE IN SEQUENCE.

                                        ERRORS:
                                        *******

                                        1.) [ERROR  46] IS CALLED TO REPORT ALL ERRORS. THE INFORMATION PRINTED
                                            INCLUDES THE LINE # WRITTEN, THE LINE # BEING CHECKED, AND THE
                                            PATTERN USED.

                                        SYNC:
                                        *****

                                        1.) WRITE LINE: M7277   SH4     LOAD SSR LOW BYTE H     CR1

                                        2.) CHECK LINE: M7277   SH4     LOAD SSR HIGH BYTE H    CP2

                                        DEBUG:  (REFER TO TEST 14)
                                        ******

                                        KEY LOGIC:      (REFER TO TEST 14)
                                        **********
                                        %
006024  010102                                  MOV     R1,R2           ;SET UP REGADR
006026  062702  000006                          ADD     #CAR,R2
006032  012705  177777                          MOV     #-1,R5          ;TEST PATERRN IN R5 = 177777
006036  010567  173142                          MOV     R5,$TMP1        ;SAVE FOR ERROR REPORTING
006042  004767  016466                  1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
006046  000465                                  BR      TST23           ;;BR IF DONE ALL LINES
006050  116767  022170  022170                  MOVB    LINE,LINEA      ;SAVE THE LINE NO. FOR ERROR LOOP

006056  105067  173126                  2$:     CLRB    $TMP3            INIT LINE COUNTER
006062  116711  173122                  3$:     MOVB    $TMP3,(R1)      ;SELECT A LINE TO CLEAR
006066  010512                                  MOV     R5,(R2)         ;LOAD CAR WITH 177777
006070  105267  173114                          INCB    $TMP3           ;GENERATE NEW LINE NO.
006074  126727  173110  000020                  CMPB    $TMP3,#20       ;DONE SETTING ALL LINES TO 177777 ?
```

```
006102  001367                          BNE     3$              ;BR IF NOT

006104  116711  022136                  MOVB    LINEA,(R1)      ;SET LINE SELECT IN SCR
006110  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
006116  005012                          CLR     (R2)            ;CLEAR THE CAR UNDER TEST

006120  105067  173062                  CLRB    $TMP2           ;INIT A LINE COUNTER
006124  005004                  4$:     CLR     R4              ;MAKE S/B DATA = 000000
006126  116711  173054                  MOVB    $TMP2,(R1)      ;SELECT A LINE TO CHECK
006132  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
006140  011203                          MOV     (R2),R3         ;GET WAS DATA
006142  126767  173040  022074          CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
006150  001401                          BEQ     5$              ;BR IF IT IS
006152  010504                          MOV     R5,R4           ;MAKE S/B DATA = 177777
006154  020304                  5$:     CMP     R3,R4           ;CORRECT DATA IN CAR ?
006156  001412                          BEQ     6$              ;BR IF YES

006160  004767  016304                  JSR     PC,SUER4        ;GO SET UP ERROR IN FO
006164  004567  016436                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
006170  001206                          $TMP2
006172  033720                          EM46+63
006174  012767  006056  172706          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
006202  104046                          ERROR   46              ;INCORRECT DATA READ FROM CAR

006204  105267  172776          6$:     INCB    $TMP2           ;GENERATE NEXT LINE NO.
006210  122767  000020  172770          CMPB    #20,$TMP2       ;DONE ALL LINES ?
006216  001711                          BEQ     1$              ;BR IF YES
006220  000741                          BR      4$              ;GO CHECK NEXT LINE
```

```
                                ;;*****************************************************************
                                ;;*TEST 23        "BCR" MEMORY PATTERNS TEST / 1'S DISTURB
                                ;;*****************************************************************
006222  000004                  TST23:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST VERIFIES THAT WHEN ALL ZEROS ARE WRITTEN INTO LINE "N"
                                IN THE "BCR" MEMORY, IT DOES NOT CLEAR ANY BITS IN ANY OTHER LOCATIONS.
                                ONLY THE LINES SELECTED BY "LINSEL" ARE TESTED. THE TEST SEQUENCE IS
                                AS FOLLOWS:

                                        1. SELECT A LINE TO TEST
                                        2. SET ALL ONES (177777) INTO ALL MEMORY LOCATIONS
                                        3. CLEAR THE SELECTED LINE
                                        4. VERIFY THAT ONLY THE SELECTED LINE WAS CLEARED
                                           AND ALL OTHER LINES STILL CONTAIN 177777
                                        5. REPEAT STEPS 1 THRU 4 UNTIL ALL SELECTED LINES ARE TESTED

                                ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES CHECKING THE NEXT
                                LINE IN SEQUENCE.

                                ERRORS:
                                *******

                                1.) [ERROR  47] IS CALLED TO REPORT ALL ERRORS. THE INFORMATION PRINTED
                                    INCLUDES THE LINE # WRITTEN, THE LINE # BEING CHECKED, AND THE
                                    PATTERN USED.

                                SYNC:
                                *****

                                1.) WRITE LINE: M7277   SH4     LOAD SSR LOW BYTE H     CR1

                                2.) CHECK LINE: M7277   SH4     LOAD SSR HIGH BYTE H    CP2

                                DEBUG:  (REFER TO TEST 15)
                                ******

                                KEY LOGIC:      (REFER TO TEST 15)
                                **********
                                %
006224  010102                          MOV     R1,R2           ;SET UP REGADR
006226  062702  000010                  ADD     #BCR,R2
006232  012705  177777                  MOV     #-1,R5          ;TEST PATERRN IN R5 = 177777
006236  010567  172742                  MOV     R5,$TMP1        ;SAVE IT FOR ERROR REPORTING
006242  004767  016266          1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
006246  000465                          BR      TST24           ;;BR IF DONE ALL LINES
006250  116767  021770  021770          MOVB    LINE,LINEA      ;SAVE THE LINE NO.

006256  105067  172726          2$:     CLRB    $TMP3           ;INIT LINE COUNTER
006262  116711  172722          3$:     MOVB    $TMP3,(R1)      ;SELECT A LINE TO INIT
006266  010512                          MOV     R5,(R2)         ;LOAD BCR WITH 177777
006270  105267  172714                  INCB    $TMP3           ;GENERATE NEW LINE NO.
006274  126727  172710  000020          CMPB    $TMP3,#20       ;DONE SETTING ALL LINES TO 177777 ?
```

```
006302  001367                          BNE     3$              ;BR IF NOT

006304  116711  021736                  MOVB    LINEA,(R1)      ;SET LINE SELECT BITS
006310  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
006316  005012                          CLR     (R2)            ;CLEAR THE BCR UNDER TEST

006320  105067  172662                  CLRB    $TMP2           ;INIT A LINE COUNTER
006324  005004                  4$:     CLR     R4              ;MAKE S/B DATA = 000000
006326  116711  172654                  MOVB    $TMP2,(R1)      ;SELECT A LINE TO CHECK
006332  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
006340  011203                          MOV     (R2),R3         ;GET WAS DATA
006342  126767  172640  021674          CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
006350  001401                          BEQ     5$              ;BR IF IT IS
006352  010504                          MOV     R5,R4           ;MAKE S/B DATA = 177777
006354  020304                  5$:     CMP     R3,R4           ;CORRECT DATA IN BCR ?
006356  001412                          BEQ     6$              ;BR IF YES

006360  004767  016104                  JSR     PC,SUER4        ;GO SET UP ERROR IN F0
006364  004567  016236                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
006370  001206                          $TMP2
006372  034120                          EM47+56
006374  012767  006256  172506          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
006402  104047                          ERROR   47              ;INCORRECT DATA READ FROM BCR

006404  105267  172576          6$:     INCB    $TMP2           ;GENERATE NEXT LINE NO.
006410  122767  000020  172570          CMPB    #20,$TMP2       ;DONE ALL LINES ?
006416  001711                          BEQ     1$              ;BR IF YES
006420  000741                          BR      4$              ;GO CHECK NEXT LINE
```

```
                                ;;*****************************************************************
                                ;*TEST 24        TEST THAT "CAR" MEMORY EXT BITS SET/CLR PROPERLY
                                ;;*****************************************************************
006422  000004                  TST24:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT THE "EXT MEM" BITS (CAR<17:16> CAN
                                BE SET AND CLEARED IN ALL "CAR" MEMORY LOCATIONS. IT WRITES THE
                                BINARY TEST PATTERNS (11, 01, AND 10) INTO BITS<17:16> TO CHECK
                                EVERY MEMORY LOCATION. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SELECT A TEST PATTERN TO USE
                                        2. CLEAR ALL 18 BITS IN ALL 16 LOCATIONS
                                        3. SELECT A LINE TO TEST
                                        4. WRITE THE TEST PATTERN INTO <17:16> OF THE
                                           SELECTED LOCATION
                                        5. READ CHECK ALL LOCATIONS TO VERIFY THAT ONLY THE
                                           SELECTED LOCATION CONTAINS THE PATTERN
                                        6. REPEAT STEPS 3 THRU 5 UNTIL ALL SELECTED LINES TESTED
                                        7. REPEAT STEPS 1 THRU 6 UNTIL ALL PATTERNS USED

                                ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES CHECKING THE NEXT
                                LINE IN SEQUENCE.

                                        NOTES:  1.) BITS<05:04> IN THE "SCR" ARE USED TO WRITE T E
                                                    EXT MEM BITS
                                                2.) BITS<07:06> IN THE "SSR" ARE USED TO CHECK BITS<17:16>

                                ERRORS:
                                *******

                                1.) [ERROR  7] IS CALLED TO REPORT ALL ERRORS

                                SYNC:
                                ******

                                1.) WRITE CAR:  M7277   SH4     LOAD LPR H      EP2

                                2.) READ CAR:   M7277   SH4     LOAD BCR H      FU2

                                DEBUG:
                                ******

                                1.) ASSUMING THAT THE PREVIOUS "CAR" MEMORY TESTS RAN ERROR FREE, THE
                                    PROBLEM IS EITHER THE M7277 OR THE M7278
                                2.) SET UP SCOPE ERROR LOOP AND START BACKTRACKING THROUGH THE LOGIC
                                    STARTING WITH THE KEY SIGNALS BELOW.

                                KEY LOGIC:
                                **********

                                        M7277   SH5     SCR05 H         CD2
                                                        SCR04 H         CE1
                                                        SSR07 H         CF1
```

```
                                                        SSR06 H          CH1

                                        M7278   SH7     74151 MUX CHIPS E66 AND E58 (INPUT PIN 12)

                                        NOTE:   THER MAY BE A PRINT ERROR ON SH7 OF THE M7278. THE
                                                SIGNALS INTO THE MUX CHIPS E66 AND E58 COME FROM THE
                                                M7277 SH5 RATHER FROM M7279 SH3.

                                %
006424  010102                          MOV     R1,R2           ;SET UP REGADR
006426  062702  000016                  ADD     #SSR,R2
006432  012705  027262                  MOV     #PATRNB,R5      ;SET UP POINTER TO DATA PATTERNS
006436  012567  172542          1$:     MOV     (R5)+,$TMP1     ;GET THE PATTERNS
006442  012567  172540                  MOV     (R5)+,$TMP2
006446  001505                          BEQ     TST25           ;;BR IF DONE ALL PATTERNS

006450  105067  172534          2$:     CLRB    $TMP3           ;INIT A LINE COUNTER
006454  142711  000017          3$:     BICB    #17,(R1)        ;INIT LINE SELECT BITS IN "SCR"
006460  156711  172524                  BISB    $TMP3,(R1)      ;SELECT A LINE IN SCR
006464  142711  000060                  BICB    #60,(R1)        ;SO WE CLEAR ALL THE MEM EXT BITS
006470  005061  000006                  CLR     CAR(R1) ;CLEAR A CAR
006474  105267  172510                  INCB    $TMP3           ;GENERATE NXT LINE NO.
006500  122767  000020  172502          CMPB    #20,$TMP3       ;CLEARED THE WHOLE THING ?
006506  001362                          BNE     3$              ;BR IF NOT

006510  004767  016020          4$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
006514  000750                          BR      1$              ;BR IF DONE ALL LINES
006516  156711  021522                  BISB    LINE,(R1)       ;SET UP LINE SELECT BITS
006522  156711  172456                  BISB    $TMP1,(R1)      ;SET UP MEM EXT BIT PATTERN
006526  012761  000000  000004          MOV     #0,LPR(R1)      ;SCOPE SYNC
006534  012761  000000  000006          MOV     #0,CAR(R1)      ;WRITE EXT BITS IN THIS LOCATION

006542  105067  172444          5$:     CLRB    $TMP4           ;INIT A LINE COUNTER
006546  016704  172434                  MOV     $TMP2,R4        ;SET UP S/B DATA
006552  142711  000017                  BICB    #17,(R1)        ;INIT SELECT BITS IN "SCR"
006556  156711  172430                  BISB    $TMP4,(R1)      ;SET SELECT BITS IN SCR
006562  012761  000000  000014          MOV     #0,BKR(R1)      ;SCOPE SYNC
006570  016103  000006                  MOV     CAR(R1),R3      ;READ THE SELECTED "CAR"
006574  011203                          MOV     (R2),R3         ;GET THE WAS DATA
006576  042703  177477                  BIC     #177477,R3      ;CLEAR JUNK BITS
006602  126767  021436  172402          CMPB    LINE,$TMP4      ;LINE UNDER TEST ??
006610  001401                          BEQ     6$              ;BR IF YES
006612  005004                          CLR     R4              ;MAKE S/B DATA = 000000
006614  020304                  6$:     CMP     R3,R4           ;WERE MEM EXT BITS CORRECT ?
006616  001412                          BEQ     7$              ;BR IF YES

006620  004767  015556                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
006624  004567  015776                  JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG BUFFER
006630  001212                          $TMP4
006632  031077                          EM7+47
006634  012767  006450  172246          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
006642  104007                          ERROR   7               ;MEM EXT BITS READ INCORRECTLY

006644  10526   172342          7$:     INCB    $TMP4           ;GENERATE NXT LINE NO.
006650  122767  000020  172334          CMPB    #20,$TMP4       ;DONE ALL LINES
006656  001674                          BEQ     2$              ;BR IF YES
```

```
 4727  006660  000732                    BR      5$              ;GO CHECK NEXT LINE
```

```
                                ;;*****************************************************************
                                ;*TEST 25        TEST INTR. ENAB. BITS - INTR. CONDITION DISABLED
                                ;;*****************************************************************
006662  000004                  TST25:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT NO TRANSMITTER OR RECEIVER INTERRUPT
                                OCCURS WHEN THE ENABLE BIT IS SET WITH OUT THE INTERRUPTING CONDITION
                                ACTIVE. A BIT MASK (INTMSK:  030100) IS UESED TO DEFINE THE I.E. BITS.
                                IN THE "SCR" (BITS 13, 12, AND 06). THE TEST SEQUENCE IS AS
                                FOLLOWS:

                                        1. SET UP THE XMIT AND RCVR VECTORS
                                        2. SELECT AN I.E. BIT TO TEST
                                        3. INIT THE SP AND LOCK OUT INTERRUPTS
                                        4. SET THE SELECTED BIT IN THE "SCR"
                                        5. CLEAR THE PSW TO ALLOW INTRS
                                        6. IF NO INTR:  REPEAT 2 THRU 5 UNTIL ALL BITS TESTED
                                        7. IF INTR:     REPORT ERROR AND CONTINUE WITH NEXT BIT TO TEST

                                ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH THF NEXT BIT
                                IN SEQUENCE .

                                ERRORS:
                                *******

                                1.) [ERROR  11] IS CALLED TO REPORT RCVR INTR FAULTS

                                2.) [ERROR  12] IS CALLED TO REPORT XMITTR INTR FAULTS

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                ******

                                1.) PROBLEM IS MOST LIKELY THE M7289 MODULE IF THIS IS THE
                                    FIRST TEST TO FAIL.
                                2.) SET UP SCOPE ERROR LOOP AND BACKTRACK THROUGH THE LOGIC
                                    STARTING WITH THE KEY LOGIC BELOW.

                                KEY LOGIC:
                                **********

                                        M7289   SH6     XMIT INT REQ H  FM1
                                                        RCV INT REQ H   DP1
                                %
006664  012767  006742  172216          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
006672  010102                          MOV     R1,R2           ;MAKE IT REGADR TOO
006674  016703  020326                  MOV     DHVCT,R3        ;GET FIRST VECTOR ADDRESS
006700  012723  006772                  MOV     #4$,(R3)+       ;GO TO 3$ IF RCVR INTRS
006704  116723  021330                  MOVB    DHRLVL,(R3)+
006710  105723                          TSTB    (R3)+           ;UPDATE POINTER
006712  012723  007014                  MOV     #5$,(R3)+       ;GO TO 5$ IF XMITTR INTRS
```

```
006716  116713  021317                  MOVB    DHTLVL,(R3)
006722  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MARKER
006726  030567  020666          1$:     BIT     R5,INTMSK       ;TEST THIS BIT ??
006732  001003                          BNE     3$              ;BR IF YES
006734  006305                  2$:     ASL     R5              ;SHIFT THE MARKER
006736  001437                          BEQ     6$              ;BR IF TESTED ALL REQUIRED BITS
006740  000772                          BR      1$              ;GO TEST FOR THIS ONE

006742  012706  001100          3$:     MOV     #STACK,SP       ;RESET SP FOR ERROR LOOPING
006746  004767  020172                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
006752  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE DH11 INTERFACE
006756  010504                          MOV     R5,R4           ;SET UP S/B DATA
006760  050511                          BIS     R5,(R1)         ;SET THE TEST I.E. BIT
006762  004767  020142                  JSR     PC,CHPS1        ;GO CLEAR PSW
006766  000240                          NOP                     ;WAIT A BIT TO ALLOW INTR
006770  000761                          BR      2$              ;OK - GO DO NEXT I.E. BIT

006772  004767  020162          4$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
006776  011103                          MOV     (R1),R3         ;GET THE WAS DATA
007000  004767  015402                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
007004  104011                          ERROR   11              ;DH11 RCVR SHOULD NOT HAVE INTERRUPTED
007006  012716  006734                  MOV     #2$,(SP)        ;SET UP TO RETURN
007012  000002                          RTI                     ;RETURN TO TEST NEXT BIT

007014  004767  020140          5$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
007020  011103                          MOV     (R1),R3         ;GET THE WAS DATA
007022  004767  015360                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
007026  104012                          ERROR   12              ;XMITTER SHOULD NOT HAVE INTERRUPTED
007030  012716  006734                  MOV     #2$,(SP)        ;SET UP TO RETURN
007034  000002                          RTI                     ;RETURN TO TEST NEXT BIT

007036  012706  001100          6$:     MOV     #STACK,SP       ;RESET THE SP JUST IN CASE
007042  004767  017702                  JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER IN VECTOR
```

```
                                ;;*****************************************************************
                                ;;*TEST 26        TEST CHAR. AVAIL. I.E.  WITH INTR. CONDITION ACTIVE
                                ;;*****************************************************************
007046  000004                  TST26:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST USES MAINT. MODE (SCR09=1) TO SET THE CHAR AVAIL
                                BIT (SCR7) TO GENERATE A RCVR INTERRUPT THROUGH THE PROPER VECTOR.
                                THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SET UP THE VECTORS
                                        2. ISSUE DH11 "MASTER CLR", RESET THE SP, AND LOCK OUT INTRS
                                        3. PRIMT THE DH11 TO GENERATE A RCVR INTR
                                        4. CLEAR THE PSW TO ALLOW INTRS
                                        5. REPORT NO RCVR INTR OR FALSE XMITTR INTR.

                                ALL ERRORS ARE REPORTED AND THEN THE TEST RESETS THE VECTOR AND SP
                                AND CONTINUES TO THE NEXT TEST IN THE PROGRAM.

                                ERRORS:
                                ********

                                1.) [ERROR  13] IS CALLED TO REPORT RCVR INTR FAULTS
                                2.) [ERROR  12] IS CALLED TO REPORT XMITTR INTR FAULTS

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                *******

                                1.) IF NO RCVR INTR OCCURRED THE PROBLEM IS EITHER SECTION "A" OF THE
                                    M7821 OR THE M7289 - SH6.
                                2.) IF A FALSE XMIT INTR OCCURRED THE PROBLEM IS MOST LIKELY THE
                                    M7821 GENERATING AN INCORRECT VECTOR ADDRESS.

                                KEY LOGIC:
                                **********

                                        M7289   SH6     E31-12

                                        M7821   SEC "A" BUS A BR L      U2
                                                        BUS SACK L      T2
                                                        BUS BG IN H     B1
                                                        "A" MASTER L    N1
                                                        VECTOR BIT 02 H D2

                                        NOTE:   REMEMBER THAT PROBLEMS IN THIS AREA COULD BE CAUSED
                                                BY ANY DEVICE IN THE SYSTEM INCLUDING THE "CPU".
                                                SYSTEM RE-CONFIGURATION TO ISOLATE THE FAULTY SUB-
                                                SYSTEM MAY BE REQUIRED.
                                %
007050  012767  007106  172032           MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
007056  010102                          MOV     R1,R2           ;MAKE IT REGADR TOO
```

```
007060  016703  020142                MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
007064  012723  007216                MOV     #3$,(R3)+       ;GO TO 3$ IF RCVR INTRS
007070  116723  021144                MOVB    DHRLVL,(R3)+
007074  105723                        TSTB    (R3)+           ;UPDATE POINTER
007076  012723  007172                MOV     #2$,(R3)+       ;GO TO 3$ ON XMITTR INTRS
007102  116713  021133                MOVB    DHTLVL,(R3)
007106  012711  004000        1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
007112  012706  001100                MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
007116  004767  020022                JSR     PC,CHPS2        ;GO LOCK OUT INTRS
007122  012711  001000                MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
007126  052711  000100                BIS     #BIT06,(R1)     ;SET CHAR AVAILABLE I.E. BIT
007132  052711  000200                BIS     #BIT07,(R1)     ;SET THE CHAR AVAIL BIT TO FORCE INTR
007136  004767  017766                JSR     PC,CHPS1        ;GO CLEAR PSW
007142  000240                        NOP                     ;GIVE IT A LITTLE TIME

007144  004767  020010                JSR     PC,SAPS         ;SAVE THE ERROR PSW
007150  011103                        MOV     (R1),R3         ;GET THE WAS DATA
007152  005011                        CLR     (R1)            ;CLEAR OUT THE SCR
007154  005011                        CLR     (R1)
007156  012704  001300                MOV     #1300,R4        ;SET UP S/B DATA
007162  004767  015220                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
007166  104013                        ERROR   13              ;TIMED OUT AWAITING CHAR AVAIL INTR
007170  000412                        BR      3$              ;GO EXIT TEST

007172  004767  017762        2$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
007176  011103                        MOV     (R1),R3         ;GET WAS DATA
007200  012704  001300                MOV     #1300,R4        ;SET UP S/B DATA
007204  005011                        CLR     (R1)            ;CLR OUT SCR REG
007206  005011                        CLR     (R1)
007210  004767  015172                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
007214  104012                        ERROR   12              ;UNEXPECTED XMITTR INTR

007216  012706  001100        3$:     MOV     #STACK,SP       ;RESET THE SP
007222  004767  017522                JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER
```

```
                                        ;;*****************************************************************
                                        ;*TEST 27        TEST SILO OVFLW.  I.E.  WITH INTR. CONDITION ACTIVE
                                        ;;*****************************************************************
007226  000004                          TST27:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT
                                        *************

                                                THIS TEST USES MAINT. MODE (SCR09=1) TO ENABLE SILO FULL
                                        INTERRUPT. THE TEST SEQUENCE IS AS FOLLOWS:

                                                1.  SET UP XMIT AND RCVR VECTORS
                                                2.  RESET THE DH11 AND S.P. - THE LOCK OUT INTRS.
                                                3.  PRIME DH11 TO GENERATE SILO FULL INTR. - ALLOW INTRS.
                                                4.  REPORT ERROR IF NO RCVR. INTR OCCURS OR
                                                    A FALSE XMIT INTR. DOES OCCUR
                                                5.  AFTER REPORTING ANY ERRORS DETECTED RESET
                                                    THE SP AND VECTORS THEN GO TO TEST 30

                                        ERRORS:
                                        *******

                                        1.  [ERROR 43] IS CALLED TO REPORT NO RCVR INTR OCCURRED
                                        2.  [ERROR 12] IS CALLED TO REPORT FALSE TRANSMITTER INTR.

                                        SYNC:   M7277   SH3     INIT A H        EF2
                                        *****

                                        DEBUG:
                                        ******

                                        1.  IF THE RECEIVER INTR FAILED TO INTERRUPT PROBLEM IS MOST LIKELY
                                            THE M7289 MODULE
                                        2.  IF A FALSE XMITTR INTR OCCURRED PROBLEM IS MOST LIKELY THE M7281 MODULE

                                        KEY LOGIC:
                                        **********

                                                M7289   SHG     E35,E50, OR E31
                                                                SCR 14 H (STORAGE) DS1

                                                M7821   "B" SECTION

                                        %
007230  012767  007266  171652                  MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
007236  010102                                  MOV     R1,R2           ;MAKE IT REGADR TOO
007240  016703  017762                          MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
007244  012723  007376                          MOV     #3$,(R3)+       ;GO TO 3$ IF RCVR INTRS
007250  116723  020764                          MOVB    DHRLVL,(R3)+
007254  105723                                  TSTB    (R3)+           ;UPDATE POINTER
007256  012723  007352                          MOV     #2$,(R3)+       ;GO TO 2$ ON XMITTR INTRS
007262  116713  020753                          MOVB    DHTLVL,(R3)
007266  012711  004000                  1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
007272  012706  001100                          MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
007276  004767  017642                          JSR     PC,CHPS2        ;GO LOCK OUT INTRS
007302  012711  001000                          MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
```

```
007306  052711  040000                  BIS     #BIT14,(R1)     ;SET SILO OVFLW I.E. BIT
007312  052711  010000                  BIS     #BIT12,(R1)     ;SET THE SILO FULL BIT TO FORCE INTR
007316  004767  017606                  JSR     PC,CHPS1        ;GO CLEAR PSW
007322  000240                          NOP                     ;GIVE IT A LITTLE TIME

007324  004767  017630                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
007330  011103                          MOV     (R1),R3         ;GET THE WAS DATA
007332  005011                          CLR     (R1)            ;CLEAR OUT THE SCR
007334  005011                          CLR     (R1)
007336  012704  051000                  MOV     #51000,R4       ;SET UP S/B DATA
007342  004767  015040                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
007346  104043                          ERROR   43              ;TIMED OUT AWAITING SILO OVFLW INTR
007350  000412                          BR      3$              ;GO EXIT TEST

007352  004767  017602          2$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
007356  011103                          MOV     (R1),R3         ;GET WAS DATA
007360  012704  051000                  MOV     #51000,R4       ;SET UP S/B DATA
007364  005011                          CLR     (R1)            ;CLR OUT SCR REG
007366  005011                          CLR     (R1)
007370  004767  015012                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
007374  104012                          ERROR   12              ;UNEXPECTED XMITTR INTR

007376  012706  001100          3$:     MOV     #STACK,SP       ;RESET THE SP
007402  004767  017342                  JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER
```

```
                                ;;*****************************************************************
                                ;*TEST 30        TEST NON EX MEM I.E.  WITH INTR. CONDITION ACTIVE
                                ;;*****************************************************************
007406  000004                  TST30:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT THE NON-EX-MEM BIT (SCR10) CAN CAUSE A
                                TRANSMITTER INTERRUPT VIA THE PROPER VECTOR. THE TEST SEQUENCE IS AS
                                FOLLOWS:

                                        1.  SET UP XMIT AND RCVR VECTORS
                                        2.  CLEAR THE DH11, RESET SP, AND LOCK OUT INTRS
                                        3.  PRIME DH11 TO GENERATE XMIT INTR IN MAINT. MODE
                                        4.  ALLOW INTRS.
                                        5.  REPORT ERROR IF NO XMIT INTR OCCURS OR IF
                                            A FALSE RCVR INTR OCCURS
                                        6.  REST SP AND VECTORS THEN GO TO TEST 31

                                ERRORS:
                                *******

                                1.  [ERROR 44] IS CALLED IF NON-EX-MEM FAILS TO GENERATE XMIT INTR
                                2.  [ERROR 11] IS CALLED IF FALSE RCVR INTR OCCURS

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                ******

                                1.  IF THE NON-EX-MEM INTERRUPT FAILS TO OCCUR PROBLEM IS MOST LIKELY
                                    THE M7289 MODULE
                                2.  IF A FALSE RCVR INTR OCCURS PROBLEM IS MOST LIKELY THE M7289 OR
                                    THE M7281 MODULES.

                                KEY LOGIC:
                                **********

                                        M7289   SH6     SCR 10 H (NO EX MEM)    FL1
                                                        E35, E41, OR E48

                                %
007410  012767  007446  171472          MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
007416  010102                          MOV     R1,R2           ;MAKE IT REGADR TOO
007420  016703  017602                  MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
007424  012723  007532                  MOV     #2$,(R3)+       ;GO TO 2$ IF RCVR INTRS
007430  116723  020604                  MOVB    DHRLVL,(R3)+
007434  105723                          TSTB    (R3)+           ;UPDATE POINTER
007436  012723  007556                  MOV     #3$,(R3)+       ;GO TO 3$ ON XMITTR INTRS
007442  116713  020573                  MOVB    DHTLVL,(R3)
007446  012711  004000          1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
007452  012706  001100                  MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
007456  004767  017462                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
007462  012711  001000                  MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
```

```
007466  052711  020000                BIS     #BIT13,(R1)     ;SET XMITTR I.E. BIT
007472  052711  002000                BIS     #BIT10,(R1)     ;SET THE NON EX MEM BIT TO FORCE INTR
007476  004767  017426                JSR     PC,CHPS1        ;GO CLEAR PSW
007502  000240                        NOP                     ;GIVE IT A LITTLE TIME

007504  004767  017450                JSR     PC,SAPS         ;SAVE THE ERROR PSW
007510  011103                        MOV     (R1),R3         ;GET THE WAS DATA
007512  005011                        CLR     (R1)            ;CLEAR OUT THE SCR
007514  005011                        CLR     (R1)
007516  012704  023000                MOV     #23000,R4       ;SET UP S/B DATA
007522  004767  014660                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
007526  104044                        ERROR   44              ;TIMED OUT AWAITING NON EX MEM INTR
007530  000412                        BR      3$              ;GO EXIT TEST

007532  004767  017422          2$:   JSR     PC,SAPS         ;SAVE THE ERROR PSW
007536  011103                        MOV     (R1),R3         ;GET WAS DATA
007540  012704  023000                MOV     #23000,R4       ;SET UP S/B DATA
007544  005011                        CLR     (R1)            ;CLR OUT SCR REG
007546  005011                        CLR     (R1)
007550  004767  014632                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
007554  104011                        ERROR   11              ;UNEXPECTED RCVR INTR

007556  012706  001100          3$:   MOV     #STACK,SP       ;RESET THE SP
007562  004767  017162                JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER
```

```
                                        ;;*****************************************************************
                                        ;*TEST 31        TEST XMITTR DONE I.E.  WITH INTR. CONDITION ACTIVE
                                        ;;*****************************************************************
 007566  000004                         TST31:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        ***************

                                                THIS TEST VERIFIES THAT XMIT DONE (SCR15) CAN BE SET IN MAINT.
                                        MODE TO CAUSE A XMITTR INTR VIA THE PROPER VECTOR. THE TEST SEQUENCE
                                        IS AS FOLLOWS:

                                                1.  SET UP XMIT AND RCVR VECTORS
                                                2.  CLEAR THE DH11, RESET SP, AND LOCK OUT INTRS
                                                3.  PRIME DH11 TO GENERATE "XMIT DONE" INTR
                                                4.  CLEAR PSW TO ALLOW INTRS
                                                5.  REPOT ERROR IF XMITTR FAILS TO INTR OR A
                                                    FALSE RCVR INTR OCCURS

                                        ERRORS:
                                        *******

                                        1.  [ERROR 45] IS CALLED TO REPORT "XMIT DONE" INTR FAILED TO OCCUR
                                        2.  [ERROR 11] IS CALLED TO REPORT FALSE RCVR INTRS

                                        SYNC:   M7277   SH3     INIT A H        EF2
                                        *****

                                        DEBUG:
                                        ******

                                        1.  IF NO XMIT INTR OCCURS PROBLEM IS MOST LIKELY THE M7289 MODULE
                                        2.  IF A FALSE RCVR INTR. OCCURS PROBLEM IS MOST LIKELY THE M7821 MODULE.


                                        KEY LOGIC:
                                        **********

                                                M7289   SH6     SCR 15 H (XMIT) FR2
                                                                E48, E50

                                        %
 007570  012767  007626  171312                 MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
 007576  010102                                 MOV     R1,R2           ;MAKE IT REGADR TOO
 007600  016703  017422                         MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
 007604  012723  007712                         MOV     #2$,(R3)+       ;GO TO 2$ IF RCVR INTRS
 007610  116723  020424                         MOVB    DHRLVL,(R3)+
 007614  105723                                 TSTB    (R3)+           ;UPDATE POINTER
 007616  012723  007736                         MOV     #3$,(R3)+       ;GO TO 3$ ON XMITTR INTRS
 007622  116713  020413                         MOVB    DHTLVL,(R3)
 007626  012711  004000                 1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
 007632  012706  001100                         MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
 007636  004767  017302                         JSR     PC,CHPS2        ;GO LOCK OUT INTRS
 007642  012711  001000                         MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
 007646  052711  020000                         BIS     #BIT13,(R1)     ;SET XMIT DONE I.E. BIT
 007652  052711  100000                         BIS     #BIT15,(R1)     ;SET THE XMITTR DONE BIT TO FORCE INTR
```

```
5123  007656  004767  017246                JSR     PC,CHPS1        ;GO CLEAR PSW
5124  007662  000240                        NOP                     ;GIVE IT A LITTLE TIME
5125
5126  007664  004767  017270                JSR     PC,SAPS         ;SAVE THE ERROR PSW
5127  007670  011103                        MOV     (R1),R3         ;GET THE WAS DATA
5128  007672  005011                        CLR     (R1)            ;CLEAR OUT THE SCR
5129  007674  005011                        CLR     (R1)
5130  007676  012704  121000                MOV     #121000,R4      ;SET UP S/B DATA
5131  007702  004767  014500                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5132  007706  104045                        ERROR   45              ;TIMED OUT AWAITING XMIT DONE INTR
5133  007710  000412                        BR      3$              ;GO EXIT TEST
5134
5135  007712  004767  017242        2$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
5136  007716  011103                        MOV     (R1),R3         ;GET WAS DATA
5137  007720  012704  121000                MOV     #121000,R4      ;SET UP S/B DATA
5138  007724  005011                        CLR     (R1)            ;CLR OUT SCR REG
5139  007726  005011                        CLR     (R1)
5140  007730  004767  014452                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5141  007734  104011                        ERROR   11              ;UNEXPECTED RCVR INTR
5142
5143  007736  012706  001100        3$:     MOV     #STACK,SP       ;RESET THE SP
5144  007742  004767  017002                JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER
```

```
                                ;;*************************************************************
                                ;*TEST 32        BASIC TRANSMITTER "NPR" LOGIC TEST 1
                                ;;*************************************************************
007746  000004                  TST32:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST TRANSMITS A SINGLE BYTE FROM LOCATION 0 ON ALL SELECTED
                                LINES (AS SELECTED BY THE CONFIGURATION PARAMETER "LINSEL:") ONE AT A TIME.
                                THE TEST SEQUENCE IS AS FOLLOWS:

                                        1.  SET UP THE XMITTR VECTOR
                                        2.  SELECT A LINE # TO TEST
                                        3.  RESET THE SP
                                        4.  CLEAR ALL LOCATIONS IN "CAR" AND "BCR" MEMORIES
                                        5.  LOCK OUT INTRS AND CLEAR THE DH11
                                        6.  PRIME DH11 TO XMIT ONE CHAR FROM LOCATION 0
                                            [9600 BAUD, 5-BITS, 1 STOP BIT]
                                        7.  ACTIVATE SELECTED XMITTR AND ENABLE XMIT DONE INTR
                                        8.  CLEAR PSW TO ALLOW INTR.
                                        9.  ACTIVATE TIMER TO WAIT FOR XMIT DONE INTR
                                        10. IF TIMEOUT OCCURS REPORT ERROR AND RESTART AT STEP 2
                                        11. IF XMIT INTERRUPT OCCURS CHECK THE FOLLOWING CONDITIONS
                                            AND REPORT ANY ERRORS:

                                            A.  XMIT DONE (SCR15=1) SET
                                            B.  "BAR" BIT GOT CLEARED
                                            C.  "CAR" REGISTER GOT INCREMENTED TO +1
                                            D.  "BCR" REGISTER GO INCREMENTED TO 0

                                        12. REPEAT STEP 2 THRU 11 UNTIL ALL SELECTED LINES TESTED
                                        13. AFTER TESTING ALL LINES CLEAR THE DH11, RESET THE VECTOR
                                            CLEAR PSW, RESET SP, AND GO TO TEST 33.

                                ERRORS:
                                *******

                                1.  [ERROR 15] IS CALLED IF XMIT DONE FAILS TO INTR ON TIME
                                2.  [ERROR 14] IS CALLED IF XMIT DONE NOT SET
                                3.  [ERROR 14] IS CALLED IF "BAR" BIT FAILED TO CLEAR
                                4.  [ERROR 14] IS CALLED IF "CAR" NOT INCREMENTED PROPERLY
                                5.  [ERROR 14] IS CALLED IF "BCR" NOT INCREMENTED PROPERLY

                                    ALL ERROR MESSAGE HEADERS INCLUDE THE LINE NO. OF THE FAILING
                                    LINE.

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****
                                        (NOTE: USE SR09=1 TO LOCK ON FAILING LINE AND SR13=1 TO
                                         INHIBIT ERROR PRINTOUT TO MINIMIZE SCOPE LOOP.)

                                DEBUG:
                                ******

                                1.  IF ALL LINES FAIL TO INTERRUPT ON TIME, SUSPECT LOSS OF 9600 BAUD
```

```
                                        CLOCK SIGNAL OR BYTE COUNT DECODER ON M7278 SH3 (XMIT FINISHED PULSE L)
                                    2.  IF GROUP OF 8 LINES <15:08> OR <07:00> FAIL TO INTERRUPT ON TIME, SUSPECT
                                        THE BUFFERED CLOCK SIGNALS (TOP AND BOT) ON THE M7288 SH3
                                    3.  IF ONLY ONE LINE FAILS TO INTERRUPT ON TIME, SUSPECT LOSS OF CLOCK IN
                                        LINE MULTIPLEXORS M7288 SH4-SH11.
                                    4.  IF LINE INTERRUPTED OK BUT "SCR", "BAR", "CAR", OR "BCR" WAS INCORRECT
                                        REFER TO KEY LOGIC SIGNALS BELOW.


                                    KEY LOGIC:
                                    **********

                                    XMIT DONE FAILED TO SET:

                                            M7289   SH6     SCR15 H FR2
                                            M7278   SH3     XMIT FINISHED PULSE L   AR1

                                    "BAR" BIT FAILED TO CLEAR:

                                            M7278   SH3     CLR BAR <15:00> L
                                                    SH5     BAR <15:12> H   E54, E55 (7474)
                                                    SH6     BAR <11:08> H   E62, E63 (7474)
                                                    SH7     BAR <07:04> H   E70, E71 (7474)
                                                    SH8     BAR <03:00> H   E78, E79 (7474)

                                    "CAR" REG NOT INCREMENTED:

                                            M7277   SH6     END CYCLE PULSE DLY H   FL2
                                                    SH5 AND SH6 "CAR" MEMORY LOGIC
                                            M796            END CYCLE H P2

                                    "BCR" REG NOT INCREMENTED:

                                            M796    END CYCLE L     N2
                                            M7278   SH3 AND SH4 "BCR" MEMORY LOGIC

                                    %
007750  012767  010004  171132              MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
007756  016703  017244                      MOV     DHVCT,R3        ;GET THE FIRST VECTOR ADDRESS
007762  062703  000004                      ADD     #4,R3           ;POINT TO XMITTR ENTRY
007766  012723  010150                      MOV     #4$,(R3)+       ;GO TO 4$ ON XMITTR INTR
007772  116713  020243                      MOVB    DHTLVL,(R3)
007776  004767  014532          1$:         JSR     PC,SELINE       ;GO SELECT A LINE NO. TO TEST
010002  000552                              BR      8$              ;BR IF TESTED ALL SELECTED LINES

010004  012706  001100          2$:         MOV     #STACK,SP       ;RESET SP FOR ERROR LOOPS
010010  010102                              MOV     R1,R2           ;SET UP REGADR
010012  012704  120000                      MOV     #120000,R4      ;SET UP S/B DATA
010016  156704  020222                      BISB    LINE,R4
010022  004767  014660                      JSR     PC,CLCABC       ;GO CLEAR CAR AND BCR MEMORIES
010026  004767  017112                      JSR     PC,CHPS2        ;GO LOCK OUT INTRS
010032  012711  004000                      MOV     #BIT11,(R1)     ;CLEAR THE DH11 INTERFACE
010036  156711  020202                      BISB    LINE,(R1)       ;SELECT A LINE NO.
010042  012761  177777  000010              MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO -1
010050  012761  033500  000004              MOV     #33500,LPR(R1)  ;SET UP LINE PARAMETERS
010056  056761  017154  000012              BIS     LINMSK,BAR(R1)  ;ACTIVATE SELECTED LINE
```

```
010064  052711  020000                  BIS     #BIT13,(R1)     ;ENABLE INTERRUPT ON XMIT DONE
010070  004767  017034                  JSR     PC,CHPS1        ;GO CLEAR PSW

010074  012767  000001  020170          MOV     #1,TIMEA        ;INIT TIMER A
010102  005067  020166                  CLR     TIMEB           ;INIT TIMER B
010106  000240                  3$:     NOP                     ;DO NOTHING WAIT
010110  004767  016662                  JSR     PC,TIMEIT       ;CALL TIMER
010114  000774                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC AROUND
                                                                ;THIS BRANCH IF TIMEOUT OCCURS

010116  004767  017036                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
010122  011103                          MOV     (R1),R3         ;GET THE WAS DATA
010124  042703  000200                  BIC     #BIT07,R3       ;WE'RE NOT INTERESTED IN THIS BIT
010130  004767  014252                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
010134  004567  014466                  JSR     R5,SUNUM        ;GO SET LINE NO. IN ERROR MSG
010140  030244                          LINE
010142  031444                          EM15+43
010144  104015                          ERROR   15              ;TIMEOUT WHILE AWAITING XMIT INTR
010146  000713                          BR      1$              ;GO TEST NEXT LINE

010150  005711                  4$:     TST     (R1)            ;DID XMIT DONE SET ??
010152  100411                          BMI     5$              ;BR IF YES

010154  004767  017000                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
010160  011103                          MOV     (R1),R3         ;GET THE WAS DATA
010162  004767  014220                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
010166  004767  000174                  JSR     PC,9$           ;GO SET UP( SOME ERROR STUFF
010172  104014                          ERROR   14              ;XMIT DONE FAILED TO SET
010174  000700                          BR      1$              ;GO TEST NEXT LINE

010176  016103  000012          5$:     MOV     BAR(R1),R3      ;GET WAS DATA FROM "BAR"
010202  001413                          BEQ     6$              ;BR IF BAR BIT GOT CLEARED

010204  004767  016750                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
010210  062702  000012                  ADD     #BAR,R2         ;SET UP REGADR
010214  005004                          CLR     R4              ;SET UP S/B DATA
010216  004767  014164                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
010222  004767  000140                  JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
010226  104014                          ERROR   14              ;BAR BIT FAILED TO CLEAR
010230  000662                          BR      1$              ;GO TEST NEXT LINE

010232  016103  000006          6$:     MOV     CAR(R1),R3      ;GET THE WAS DATA FROM CAR
010236  022703  000001                  CMP     #1,R3           ;DID IT GET INCREMENTED ?
010242  001414                          BEQ     7$              ;BR IF YES

010244  004767  016710                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
010250  012704  000001                  MOV     #1,R4           ;SET UP S/B DATA
010254  062702  000006                  ADD     #CAR,R2         ;SET UP REGADR
010260  004767  014122                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
010264  004767  000076                  JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
010270  104014                          ERROR   14              ;CAR REG NOT  INCREMENTED PROPERLY
010272  000641                          BR      1$              ;GO TEST NEXT LINE

010274  016103  000010          7$:     MOV     BCR(R1),R3      ;GET WAS DATA FROM BCR
010300  001636                          BEQ     1$              ;BR IF BCR GOT INCREMENTED TO 000000
```

```
010302  004767  016652                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
010306  005004                          CLR     R4              ;SET UP S/B DATA
010310  062702  000010                  ADD     #BCR,R2         ;SET UP REGADR
010314  004767  014066                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
010320  004767  000042                  JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
010324  104014                          ERROR   14              ;BCR REG NOT INCREMENTED PROPERLY
010326  000623                          BR      1$              ;GO TEST NEXT LINE

010330  012711  004000          8$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
010334  016703  016666                  MOV     DHVCT,R3        ;GET THE VECTOR ADDR
010340  062703  000004                  ADD     #4,R3           ;POINT TO XMIT VECTOR
010344  010313                          MOV     R3,(R3)         ;RESTORE TRAP CATCHER
010346  062723  000002                  ADD     #2,(R3)+
010352  005013                          CLR     (R3)
010354  004767  016550                  JSR     PC,CHPS1        ;GO CLEAR PSW
010360  012706  001100                  MOV     #STACK,SP       ;RESET THE STACK POINTER
010364  000405                          BR      TST33           ;;GO TO NEXT TEST

010366  004567  014234          9$:     JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG.
010372  030244                          LINE
010374  031376                          EM14+44
010376  000207                          RTS     PC              ;RETURN TO REPORT ERROR
```

```
                                        ;;*************************************************************
                                        ;*TEST 33        TRANSMITTR NPR LOGIC TEST 2
                                        ;;*************************************************************
010400  000004                          TST33:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        **************

                                                THIS TEST IS SIMILAR TO TEST 32 EXCEPT THAT ALL LOCATIONS IN THE "BCR"
                                        AND "CAR" MEMORIES ARE TESTED TO VERIFY THAT TRANSMISSION ON THE SELECTED
                                        LINE DID NOT DISTURB ANY UNSELECTED LOCATIONS IN THE MEMORIES. IF ALSO OPERATES
                                        IN "FLAG" MODE RATHER THAN USING INTERRUPTS. THE TEST SEQUENCE IS AS
                                        FOLLOWS:

                                                1.  SELECT A LINE # TO TEST (AS DEFINED BY "LINSEL:")
                                                2.  CLEAR BOTH THE "CAR" AND "BCR" MEMORIES
                                                3.  LOAD THE "BCR" MEMORY WITH ALL ONES (BYTE COUNT = -1)
                                                4.  ACTIVATE THE XMITTER ON THE SELECTED LINE
                                                5.  ACTIVATE TIMER TO WAIT FOR "XMIT DONE"
                                                6.  IF "XMIT DONE" FAILS TO SET ON TIME - REPORT ERROR AND
                                                    REPEAT 1 THRU 5 UNTIL ALL SELECTED LINES TESTED
                                                7.  IF "XMIT DONE" SETS CHECK ALL LOCATIONS IN THE "BCR" MEMORY
                                                    REPORT ANY UNSELECTED LINES NOT CONTAINING -1 AND THE SELECTED
                                                    LINE IF IT DOES NOT CONTAIN 0
                                                8.  CHECK ALL LOCATIONS IN THE "CAR" MEMORY AND REPORT ANY UNSELECTED
                                                    LOCATIONS NOT CONTAINING 0 AND THE SELECTED LINE IF IT DOES NOT
                                                    CONTAIN +1.
                                                9.  REPEAT STEPS 1 THRU 8 UNTIL ALL SELECTED LINES TESTED.

                                        ERRORS:
                                        *******

                                        1.  [ERROR 50] CALLED IF XMIT DONE TIMEOUT ERROR DETECTED.
                                        2.  [ERROR 51] CALLED IF "BCR" MEMORY ERROR DETECTED
                                        3.  [ERROR 51] CALLED IF "CAR" MEMORY ERROR DETECTED

                                        SYNC:   M7277   SH3     INIT A H        EF2
                                        *****

                                        DEBUG:
                                        ******

                                        1.  ASSUMING TEST 32 RAN ERROR FREE THE PROBLEM IS MOST LIKELY THE:
                                                M7278 MODULE IF "BCR" ERRORS
                                                M7277 MODULE IF "CAR" ERRORS

                                        KEY LOGIC:      (SAME AS TEST 32)
                                        **********

                                        %
010402  012767  010416  170500                  MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
010410  004767  014120                  1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
010414  000544                                  BR      TST34           ;;BR IF DONE ALL SELECTED LINES
010416  052711  004000                  2$:     BIS     #BIT11,(R1)     ;CLEAR THE DH11
010422  004767  014260                          JSR     PC,CLCABC       ;GO CLEAR "CAR" AND "BCR" MEMORIES
010426  004767  014316                          JSR     PC,LDBCR        ;GO LOAD "BCR" MEMORY WITH ALL ONES
```

```
010432  156711  017606                  BISB    LINE,(R1)          ;SELECT THE LINE
010436  012761  033500  000004          MOV     #33500,LPR(R1)     ;SET UP PARAMETERS
010444  016761  016566  000012          MOV     LINMSK,BAR(R1)     ;ACTIVATE XMIT ON SELECTED LINE

010452  012767  000001  017612          MOV     #1,TIMEA           ;INIT TIMER A
010460  005067  017610                  CLR     TIMEB              ;INIT TIMER B
010464  005711                  3$:     TST     (R1)               ;XMITTR DONE YET
010466  100423                          BMI     4$                 ;BR IF YES
010470  004767  016302                  JSR     PC,TIMEIT          ;CALL THE TIMER
010474  000773                          BR      3$                 ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                   ;AROUND THIS BRANCH IF TIME OUT OCCURS

010476  004767  016456                  JSR     PC,SAPS            ;SAVE THE ERROR PSW
010502  011103                          MOV     (R1),R3            ;GET THE WAS DATA
010504  012704  100000                  MOV     #BIT15,R4          ;SET UP S/B DATA
010510  156704  017530                  BISB    LINE,R4
010514  010102                          MOV     R1,R2              ;MAKE REGADR = DEVADR
010516  004767  013664                  JSR     PC,SUER2A          ;GO SET UP ERROR INFO
010522  004567  014100                  JSR     R5,SUNUM           ;SET LINE NO. IN MSG
010526  030244                          LINE
010530  034176                          EM50+53
010532  104050                          ERROR   50                 ;TIMED OUT AWAITING XMIT DONE ON SEL LINE
010534  000725                          BR      1$                 ;GO TRY THE NEXT LINE

010536  005067  170456          4$:     CLR     $TMP7              ;INIT A LINE COUNTER
010542  116711  170452          5$:     MOVB    $TMP7,(R1)         ;SELECT LINE NO. IN "SCR"
010546  012704  177777                  MOV     #-1,R4             ;SET UP S/B DATA
010552  016103  000010                  MOV     BCR(R1),R3         ;GET THE WAS BYTE COUNT
010556  126767  017462  170434          CMPB    LINE,$TMP7         ;WAS THIS THE ACTIVE LINE ??
010564  001001                          BNE     6$                 ;BR IF NOT
010566  005004                          CLR     R4                 ;CHANGE S/B DATA TO 000000
010570  020304                  6$:     CMP     R3,R4              ;WAS BYTE COUNT CORRECT ??
010572  001416                          BEQ     7$                 ;BR IF YES

010574  005067  170402                  CLR     $TMP0              ;SAVE THE ACTIVE LINE NO.
010600  116767  017440  170374          MOVB    LINE,$TMP0
010606  016767  170406  170370          MOV     $TMP7,$TMP1        ;SAVE THE LINE NO. BEING CHECKED
010614  010102                          MOV     R1,R2              ;SET UP REGADR = BCR REG ADDR
010616  062702  000010                  ADD     #BCR,R2
010622  004767  013642                  JSR     PC,SUER4           ;GO SET UP ERROR INFO
010626  104051                          ERROR   51                 ;BYTE COUNT INCORRECT

010630  005004                  7$:     CLR     R4                 ;SET UP S/B DATA
010632  016103  000006                  MOV     CAR(R1),R3         ;GET THE WAS DATA
010636  126767  017402  170354          CMPB    LINE,$TMP7         ;IS THIS THE ACTIVE LINE
010644  001001                          BNE     8$                 ;BR IF NOT
010646  005204                          INC     R4                 ;BUMP THE CAR ADDRESS FOR ACTIVE LINE
010650  020304                  8$:     CMP     R3,R4              ;CAR CONTENTS CORRECT ??
010652  001416                          BEQ     9$                 ;BR IF YES

010654  005067  170322                  CLR     $TMP0              ;SET UP ACT LINE NO.
010660  116767  017360  170314          MOVB    LINE,$TMP0
010666  016767  170326  170310          MOV     $TMP7,$TMP1        ;SAVE THE LINE NO. BEING CHECKED
010674  010102                          MOV     R1,R2              ;SET UP REGADR
010676  062702  000006                  ADD     #CAR,R2
010702  004767  013562                  JSR     PC,SUER4           ;SET UP THE ERROR INFO
```

```
010706  104051                          ERROR   51              ;CAR REG INCORRECT

010710  005267  170304          9$:     INC     $TMP7           ;GENERATE NEW LINE NO.
010714  022767  000020  170276          CMP     #20,$TMP7       ;TESTED ALL LINES
010722  001707                          BEQ     5$              ;BR IF NOT
010724  000631                          BR      1$              ;GO SELECT NEXT ACTIVE LINE
```

```
                                ;;*****************************************************************
                                ;*TEST 34        TEST THAT CHARACTER AVAILABLE CAN CAUSE RCVR INTERRUPT
                                ;;*****************************************************************
010726  000034                  T$T34:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT WHEN "CHAR AVAIL" (BIT07 IN "SCR") SETS
                                AS A RESULT OF XMITTING AND RECEIVING ONE CHARACTER (USING SILO MAINT MODE)
                                IT CAUSES A RCVR INTR VIA THE PROPER VECTOR. THE TEST SEQUENCE IS AS
                                FOLLOWS:

                                        1.  SET UP THE RCVR VECTOR
                                        2.  LOCK OUT INTRS, RESET SP, AND CLEAR DH11
                                        3.  USE MAINT MODE TO LOAD A CHAR INTO THE SILO (DATA=125252)
                                        4.  CLEAR PSW TO ALLOW INTERRUPTS
                                        5.  ACTIVATE TIMER TO WAIT FOR INTR TO OCCUR
                                        6.  IF NO RCVR INTR OCCURS REPORT ERROR AND GO TO STEP 9
                                        7.  WHEN RCVR INTRS - CHECK SILO DATA FOR 125252 - IF NOT
                                            CORRECT REPORT ERROR AND GO TO STEP 9
                                        8.  CHECK THAT SILO FILL LEVEL=1 - IF NOT REPORT ERROR
                                        9.  RESET SP, CLEAR PSW, RESET VECTOR, AND GO TO TEST 35

                                ERRORS:
                                *******

                                1.  [ERROR 13] IS CALLED TO REPORT RCVR TIMEOUT ERROR
                                2.  [ERROR 52] IS CALLED TO REPORT SILO DATA INCORRECT
                                3.  [ERROR  6] IS CALLED TO REPORT INCORRECT SILO FILL COUNT

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                ******

                                1.  IF NOT RCVR INTR OCCURS SUSPECT THE M7277, M7281, OR M7279 MODULES
                                2.  IF SILO DATA OR FILL-ERRORS SUSPECT THE M7279 MODULE

                                KEY LOGIC:
                                **********

                                        M7279   SH1     SILO DATA MUX'S (74157'S)
                                                        SSR15 H         CR1

                                                SH2     DATA READY L            DV1
                                                        NRC 15 H                DL1
                                                        SILO MEMORY             (E13,E17,E8,E3) [3341'S]
                                                        LOAD SILO L             DJ2
                                                        5.068 MHZ (CLOCK)       DN1
                                                        SSR <13:00> H           (E20,E24)

                                        M7289   SH6     RCV INT REQ H           DP1

                                        M7278   SH8     SSR <03:00> H
```

```
                                                        SH7     SSR <07:00> H

                                        %

010730  012767  010752  170152           MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
010736  016703  016264                   MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
010742  012723  011036                   MOV     #3$,(R3)+       ;GO TO 3$ ON RCVR INTERRUPT
010746  116713  017266                   MOVB    DHRLVL,(R3)
010752  004767  016166          1$:      JSR     PC,CHPS2        ;GO LOCK OUT INTRS
010756  012706  001100                   MOV     #STACK,SP       ;RESET SP FOR ERROR LOOPS
010762  012711  004000                   MOV     #BIT11,(R1)     ;CLEAR THE DH11
010766  052761  100000  000016           BIS     #BIT15,SSR(R1)  ;SET SILO MAINT. BIT TO LOAD SILO
010774  012711  000100                   MOV     #BIT06,(R1)     ;ENABLE CHAR. AVAIL INTERRUPT
011000  004767  016124                   JSR     PC,CHPS1        ;GO CLEAR PSW
011004  012703  001000                   MOV     #1000,R3        ;INIT TIMER
011010  005303                  2$:      DEC     R3              ;DEC TIMER
011012  001376                           BNE     2$              ;BR IF NO TIMEOUT

011014  004767  016140                   JSR     PC,SAPS         ;SAVE THE ERROR PSW
011020  011103                           MOV     (R1),R3         ;GET THE WAS DATA
011022  012704  000300                   MOV     #300,R4         ;SET UP S/B DATA
011026  004767  013354                   JSR     PC,SUER2A       ;GO SET UP ERROR INFO
011032  104013                           ERROR   13              ;CHAR AVAIL FAILED TO SET ON TIME
011034  000436                           BR      5$              ;ESCAPE FROM THIS TEST - CATASTROPHIC ERROR

011036  016105  000016          3$:      MOV     SSR(R1),R5      ;SAVE THE SILO STATUS REG.
011042  016103  000002                   MOV     NRC(R1),R3      ;GET THE WAS DATA
011046  012704  125252                   MOV     #125252,R4      ;SET UP S/B DATA
011052  020304                           CMP     R3,R4           ;WAS = S/B = 125252 ??
011054  001410                           BEQ     4$              ;BR IF IT IS

011056  004767  016076                   JSR     PC,SAPS         ;SAVE THE ERROR PSW
011062  062702  000002                   ADD     #NRC,R2         ;SET  UP REGADR
011066  004767  013314                   JSR     PC,SUER2A       ;GO SET UP  ERROR INFO
011072  104052                           ERROR   52              ;DATA COMPARE ERROR
011074  000416                           BR      5$              ;GET OUT

011076  010503                  4$:      MOV     R5,R3           ;NOW GET THW SILO STATUS REG AGAIN
011100  042703  140377                   BIC     #140377,R3      ;CLR OUT JUNK
011104  012704  000400                   MOV     #400,R4         ;SET UP S/B DATA
011110  020304                           CMP     R3,R4           ;SSR CHAR COUNT = 1 ??
011112  001407                           BEQ     5$              ;BR IF IT IS

011114  004767  016040                   JSR     PC SAPS         ;SAVE THE ERROR PSW
011120  062702  000016                   ADD     #SSR,R2         ;SET UP REGADR
011124  004767  013256                   JSR     PC,SUER2A       ;SET UP ERROR INFO
011130  104006                           ERROR   6               ;SSR COUNT NOT CORRECT

011132  012706  001100          5$:      MOV     #STACK,SP       ;RESET THE STACK POINTER
011136  004767  015766                   JSR     PC,CHPS1        ;GO CLEAR PSW
011142  005011                           CLR     (R1)            ;RESET I.E. BIT
011144  016703  016056                   MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
011150  010313                           MOV     R3,(R3)         ;RESTORE TRAP CATCHER
011152  062723  000002                   ADD     #2,(R3)+
011156  005013                           CLR     (R3)
```

```
                                ;;*****************************************************************
                                ;*TEST 35        TEST THAT THE SILO STATUS REG COUNTS UP CORRECTLY
                                ;;*****************************************************************
011160  000004                  TST35:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT THE SILO FILL LEVEL COUNTS UP CORRECTLY
                                WHEN ALL COUNTS (0-77) ARE TESTED BY LOADING THE SILO USING
                                MAINT MODE. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1.  INIT "$TMP7" TO START WITH A COUNT=01
                                        2.  CLEAR THE DH11
                                        3.  LOAD THE SILO WITH 125252'S IN MAINT MODE
                                            UNTIL # OF WORDS INDICATED BY THE COUNT ARE LOADED
                                        4.  AFTER LOADING REQUIRED COUNT CHECK THAT FILL LEVEL
                                            BITS (SSR<13:08>) EQUAL COUNT - IF NOT REPORT ERROR
                                        5.  INCREMENT COUNT IN "$TMP7" AND REPEAT 1 THRU 4
                                            UNTIL ALL COUNTS (01-77) HAVE BEEN TESED

                                ERRORS:
                                *******

                                1.  [ERROR 6] IS CALLED TO REPORT SILO FILL LEVEL ERRORS

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                ******

                                1.  FAILURES IN THIS TEST MOST LIKELY INDICATE A BAD M7279 MODULE

                                KEY LOGIC:
                                **********

                                        M7279   SH2     SSR <13:08>
                                                        LOAD SILO L DJ2
                                                        5.068 MHZ (CLOCK) DN1
                                                        DATA READY L DV1

                                %
011162  010102                          MOV     R1,R2           ;MAKE REGADR = SSR
011164  062702  000016                  ADD     #SSR,R2
011170  012767  000001  170022          MOV     #1,$TMP7        ;START WITH COUNT OF 1
011176  012711  004000          1$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
011202  016705  170012                  MOV     $TMP7,R5        ;SAVE CHARACTER COUNT BEING TESTED
011206  005004                          CLR     R4              ;INIT A CHAR COUNTER
011210  052712  100000          2$:     BIS     #BIT15,(R2)     ;SET THE SILO MAINT BIT
011214  012703  001000                  MOV     #1000,R3        ;INIT A TIMER
011220  005303                  3$:     DEC     R3              ;STALL TO ALLOW TIME TO LOAD SILO
011222  001376                          BNE     3$
011224  042712  100000                  BIC     #BIT15,(R2)     ;CLEAR SILO MAINT. BIT
011230  005204                          INC     R4              ;COUNT A CHAR LOADED
011232  005305                          DEC     R5              ;DECREMENT TEST COUNT
```

```
011234  001365                          BNE     2$              ;BR UNTIL WE'VE LOADED THE TEST COUNT

011236  011203                          MOV     (R2),R3         ;SET THE WAS COUNT
011240  042703  140377                  BIC     #140377,R3      ;CLR JUNK BITS
011244  000304                          SWAB    R4              ;SET UP S/B DATA
011246  020304                          CMP     R3,R4           ;TEST COUNT = SILO COUNTER ?
011250  001406                          BEQ     4$              ;BR IF YES

011252  004767  013124          JSR     PC,SUER2                ;GO SET UP ERROR INFO
011256  012767  011176  167624          MOV     #1$,$LPERR      ;SET UP ERROR LOOP RETURN
011264  104006                          ERROR   6               ;SSR FAILED TO UP-COUNT CORRECTLY

011266  005267  167726          4$:     INC     $TMP7           ;INCREMENT TO NEXT COUNT TO TEST
011272  022767  000100  167720          CMP     #100,$TMP7      ;MAXIMUM COUNT ??
011300  001336                          BNE     1$              ;BR IF NOT
```

```
                                ;;*****************************************************************
                                ;*TEST 36        TEST THAT SILO STATUS REGISTER DOWN COUNTS CORRECTLY
                                ;;*****************************************************************
011302  000004                  T36:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT THE SILO FILL LEVEL COUNTS DOWN PROPERLY
                                WHEN WORDS ARE READ FROM THE SILO. ALL COUNTS FROM 77-00 ARE
                                TESTED. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1.  INIT "$TMP7" TO START WITH A COUNT OF 1
                                        2.  CLEAR THE DH11 AND FILL SILO WITH 64. WORDS
                                        3.  READ THE NO. OF WORDS SPECIFIED BY COUNT
                                        4.  CHECK THAT FILL LEVEL=64. MINUS [COUNT] - REPORT ERRORS
                                        5.  INCREMENT "$TMP7" AND REPEAT 2 THRU 4 UNTIL ALL
                                            COUNTS TESTED.

                                ERRORS:
                                *******

                                1.  [ERROR 6] IS CALLED TO REPORT SILO FILL LEVEL ERRORS

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:          (REFER TO TEST 35)
                                ******

                                KEY LOGIC:      (REFER TO TEST 35)
                                **********

                                %
011304  010102                          MOV     R1,R2           ;SET UP REGADR
011306  062702  000016                  ADD     #SSR,R2
011312  012767  000001  167700          MOV     #1,$TMP7        ;START WITH COUNT = 1
011320  012711  004000          1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
011324  012705  000100                  MOV     #100,R5         ;TEST COUNT SHOULD BE 64(10) MINUS
011330  166705  167664                  SUB     $TMP7,R5        ;THE NO. OF CHARS READ
011334  012703  000100                  MOV     #100,R3         ;COUNTER USED TO FILL SILO
011340  012704  001000          2$:     MOV     #1000,R4        ;INIT TIMER
011344  052712  100000                  BIS     #BIT15,(R2)     ;SET SILO MAINT. BIT
011350  005304                  3$:     DEC     R4              ;STALL TO ALLOW SILO TO LOAD
011352  001376                          BNE     3$
011354  042712  100000                  BIC     #BIT15,(R2)     ;CLEAR THE SILO MAINT BIT
011360  005303                          DEC     R3              ;COUNT ONE CHAR LOADED
011362  001366                          BNE     2$              ;BR UNTIL ALL LOADED

011364  016703  167630                  MOV     $TMP7,R3        ;INIT COUNTER FOR READING SILO
011370  012704  001000          4$:     MOV     #1000,R4        ;INIT TIMER
011374  005761  000002                  TST     NRC(R1)         ;READ THE SILO
011400  005304                  5$:     DEC     R4              ;GIVE IT TIME TO SETTLE
011402  001376                          BNE     5$
011404  005303                          DEC     R3              ;COUNT ONE READ
011406  001370                          BNE     4$              ;BR UNTIL WE'VE READ TEST COUNT
```

```
011410  011203                          MOV     (R2),R3         ;GET THE WAS DATA
011412  042703  140377                  BIC     #140377,R3      ;CLR JUNK BITS
011416  010504                          MOV     R5,R4           ;SET UP S/B DATA
011420  000304                          SWAB    R4
011422  020304                          CMP     R3,R4           ;DID IT DOWN COUNT OK ??
011424  001406                          BEQ     6$              ;BR IF YES

011426  004767  012750                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
011432  012767  011320  167450          MOV     #1$,$LPERR      ;SET UP ERROR LOOP RETURN
011440  104006                          ERROR   6               ;SILO STATUS REG. DOWN-COUNTED INCORRECTLY

011442  005267  167552          6$:     INC     $TMP7           ;UPDATE COUNT
011446  022767  000101  167544          CMP     #101,$TMP7      ;TESTED ALL COUNTS ??
011454  001321                          BNE     1$              ;BR IF NOT
```

```
                                        ;;*************************************************************
                                        ;*TEST 37       TEST SILO ALARM LEVEL FOR  COUNTS 0,1,2,4,8,16, AND 32
                                        ;;*************************************************************
011456  000004                          TST37:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        **************

                                                THIS TEST VERIFIES THAT THE SILO ALARM LEVEL WORKS PROPERLY FOR
                                        INTEGRAL POWER OF 2 COUNTS (0, 1, 2, 4, 8, 16, AND 32). THE TEST SEQUENCE
                                        IS AS FOLLOWS:

                                                1.  INIT "$TMP7" TO START WITH ALARM LEVEL OF 000
                                                2.  CLEAR THE DH11 AND LOAD THEN SILO WITH THAT NO. OF WORDS
                                                    THAT IS ONE GREATER THAN THE ALARM LEVEL.
                                                3.  VERIFY THAT "DATA READY" DOES NOT SET UNTIL THE FILL
                                                    LEVEL EXCEEDS THE ALARM LEVEL.
                                                4.  REPORT ERRORS IF:
                                                    A.  "READY" SETS TOO SOON
                                                    B.  "READY" SETS TOO LATE
                                                5.  SHIFT "$TMP7" LEFT TO GENERATE NEXT POWER OF 2 LEVEL
                                                6.  REPEAT 2 THRU 5 UNTIL ALL 7 TEST LEVELS CHECKED

                                                NOTE:   FOR (A) ABOVE IF "READY" SETS JUST ONE WORD TOO
                                                        SOON, IT IS ALLOWED, BUT ANYTHING GREATER RESULTS IN
                                                        AN ERROR MESSAGE.

                                        ERRORS:
                                        *******

                                        1.  [ERROR 6] IS CALLED TO REPORT BOTH TYPES OF ERRORS OUTLINED
                                            IN 4(A,B) ABOVE

                                        SYNC:   M7277   SH3     INIT A H        EF2
                                        *****

                                        DEBUG:
                                        ******

                                        1.  ERRORS IN THIS TEST ONLY INDICATE BAD COMPARATOR CHIP (E23 OR E19)
                                            ON THE M7279 - SH2

                                        KEY LOGIC:
                                        **********

                                                M7279   SH2     E19 - PIN 5 (COMPARATOR)
                                                ALSO SAME LOGIC AS TEST 35

                                        %
011460  012767  011500  167422                  MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
011466  010102                                  MOV     R1,R2           ;SET UP REGADR
011470  062702  000016                          ADD     #SSR,R2
011474  005067  167520                          CLR     $TMP7           ;START WITH LEVEL 00
011500  012711  004000                  1$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
011504  016705  167510                          MOV     $TMP7,R5        ;SAVE IT IN R5
011510  010512                                  MOV     R5,(R2)         ;SET ALARM LEVEL IN SSR
```

```
011512  005205                          INC     R5              ;LOAD ONE MORE THAN FILL LEVEL
011514  052712  100000          2$:     BIS     #BIT15,(R2)     ;SET SILO MAINT. TO LOAD A CHAR
011520  012703  001000                  MOV     #1000,R3        ;INIT STALL TIMER
011524  005303                  3$:     DEC     R3              ;WAIT FOR SILO TO SETTLE
011526  001376                          BNE     3$              ;BR TIL R3 GOES TO 000000
011530  042712  100000                  BIC     #BIT15,(R2)     ;CLR THE SILO MAINT BIT
011534  005305                          DEC     R5              ;COUNT ONE LOADED
011536  105711                          TSTB    (R1)            ;CHAR AVAIL SET YET
011540  100412                          BMI     4$              ;BR IF IT IS
011542  005705                          TST     R5              ;SHOULD IT BE ??
011544  001363                          BNE     2$              ;BR IF NOT

011546  004767  015406                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
011552  004767  000042                  JSR     PC,5$           ;GO SET UP S/B DATA
011556  004767  012624                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
011562  104006                          ERROR   6               ;SILO ALARM LEVEL FAILED AT SELECTED COUNT
011564  000426                          BR      6$              ;GO CHECK NEXT COUNT

011566  005705                  4$:     TST     R5              ;SHOULD IT HAVE BEEN SET (CHAR AVAIL)
011570  001424                          BEQ     6$              ;BR IF YES

011572  004767  015362                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
011576  022705  000001                  CMP     #1,R5           ;IS IT OFF BY ONLY ONE ??
011602  001417                          BEQ     6$              ;BR IF YES - WE'LL ALLOW HIM THIS
011604  004767  000010                  JSR     PC,5$           ;GO SET UP S/B DATA
011610  004767  012572                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
011614  104006                          ERROR   6               ;SILO ALARM LEVEL FAILED
011616  000411                          BR      6$              ;GO CHECK NEXT COUNT

011620  011203                  5$:     MOV     (R2),R3         ;GET WAS DATA
011622  016704  167372                  MOV     $TMP7,R4        ;SET UP THE S/B DATA
011626  005204                          INC     R4
011630  000304                          SWAB    R4
011632  105004                          CLRB    R4
011634  156704  167360                  BISB    $TMP7,R4
011640  000207                          RTS     PC              ;RETURN TO SET UP AND REPORT ERROR

011642  005767  167352          6$:     TST     $TMP7           ;COUNT AT ZERO
011646  001002                          BNE     7$              ;BR IF NOT
011650  000261                          SEC                     ;SET THE "C" BIT
011652  000401                          BR      8$              ;GO SET UP COUNT
011654  000241                  7$:     CLC                     ;CLEAR THE "C" BIT
011656  006167  167336          8$:     ROL     $TMP7           ;SHIFT POWER OF TWO BIT
011662  032767  000100  167330          BIT     #BIT6,$TMP7     ;DONE ALL POWERS ??
011670  001703                          BEQ     1$              ;BR IF NOT
```

```
                                ;;*****************************************************************
                                ;*TEST 40        TRANSMITTER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS
                                ;;*****************************************************************
011672  000004                  TST40:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST PERFORMA A "RELATIVE" TIMING TEST FOR ALL BAUD RATES
                                ON ALL SELECTED LINES. IT DOES NOT MEASURE ABSOLUTE TIMES BUT
                                SIMPLY VERIFIES THAT EACH SUCCESSIVE SPEED FROM 50 TO 9600
                                BAUD IS FASTER THAN THE PREVIOUS SPEED. THE TEST SEQUENCE IS AS
                                FOLLOWS:

                                        1.  SELECT A LINE # TO TEST (AS DEFINED BY "LINSEL:")
                                        2.  INIT "STMP7" TO START WITH 50 BAUD AND A RELATIVE
                                            TIMER "TIMEC" TO -1 (177777)
                                        3.  CLEAR THE DH11 AND ACTIVATE SELECTED LINE TO TRANSMIT
                                            THREE CHARS.
                                        4.  ACTIVATE TIMER TO UPDATE "TIMEB" THE LINE SPEED TIMER
                                        5.  IF "XMIT DONE" FAILS TO SET ON TIME - REPORT ERROR
                                            AND REPEAT 3 THRU 4 UNTIL ALL SPEEDS CHECKED - THEN
                                            REPEAT 1 THRU 5 UNTIL ALL LINES CHECKED
                                        6.  IF "XMIT DONE" SETS VERIFY [TIMEB] LESS THAN [TIMEC]
                                            IF NOT REPORT ERROR - MAKE [TIMEC]=[TIMEB] AND
                                            REPEAT 3 THRU 5 UNTIL ALL SPEEDS CHECKED
                                        7.  REPEAT 1 THRU 6 FOR ALL SELECTED LINES.

                                ERRORS:
                                *******

                                1.  [ERROR 53] IS CALLED TO REPORT XMIT TIMEOUT ERRORS
                                2.  [ERROR 17] IS CALLED TO REPORT TIMING ERRORS

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                ******

                                1.  IF ALL LINES FAIL ON ALL SPEEDS SUSPECT THE CLOCK MODULE M4540
                                2.  IF ALL LINES FAIL ON JUST ONE SPEED (THE SAME ONE) SUPECT EITHER THE
                                    CLOCK MODULE OR THE M7288 MODULE (TIMING SELECT MUXES)
                                3.  IF JUST ONE LINE FAILS SUSPECT EITHER THE UART MODULE (M7280)
                                    EITHER FOR LINES <15:08> OR <07:00> OR THE M7288 MODULE

                                KEY LOGIC:
                                **********

                                        M4540   SH2             <9600:50> BAUD SIGNALS
                                        M7288   SH3             BOT AND TOP BUF CLOCK SIGNALS
                                                SH4,6,8,OR 10   TX CLOCK NN L SIGNALS
                                        M7280   TBMT LINE "N" SIGNALS ON UART PIN 22
                                                TX CLOCK LINE "N" SIGNALS ON UART PIN 40

                                %
```

```
011674  012767  011724  167206          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
011702  004767  012626          1$:     JSR     PC,$ELINE       ;GO SELECT A LINE TO TEST
011706  000534                          BR      TST41           ;;BR IF TESTED ALL SELECTED LINES
011710  012767  002100  167302          MOV     #2100,$TMP7     ;INIT TI START WITH LOWEST SPEED
011716  012767  177777  016352          MOV     #-1,TIMEC       ;INIT RELATIVE TIME CHECKER
011724  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
011730  156711  016310          3$:     BISB    LINE,(R1)       ;SELECT IT IN THE SCR
011734  012761  177775  000010          MOV     #-3,BCR(R1)     ;SET BYTE COUNT TO XFER 3 CHARS
011742  005061  000006                  CLR     CAR(R1)         ;GET TEST DATA STARTING AT LOC. 0
011746  016761  167246  000004          MOV     $TMP7,LPR(R1)   ;SELECT A XMIT SPEED
011754  016761  015256  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE TRANSMITTER

011762  012767  000001  016302          MOV     #1,TIMEA        ;INIT TIMER A
011770  005067  016300                  CLR     TIMEB           ;INIT TIMER B
011774  005711                  4$:     TST     (R1)            ;XMITTR DONE SET YET ?
011776  100437                          BMI     5$              ;BR IF YES
012000  004767  014772                  JSR     PC,TIMEIT       ;CALL THE TIMER
012004  000773                          BR      4$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS

012006  016767  167206  167166          MOV     $TMP7,$TMP0     ;SAVE AND SET UP THE SPEED CODE
012014  000367  167162                  SWAB    $TMP0
012020  006267  167156                  ASR     $TMP0
012024  006267  167152                  ASR     $TMP0
012030  042767  177760  167144          BIC     #177760,$TMP0
012036  011103                          MOV     (R1),R3         ;GET THE WAS DATA
012040  042703  000200                  BIC     #BIT07,R3       ;CLEAR UNINTERESTING BITS
012044  010102                          MOV     R1,R2           ;MAKE REGADR = DEVADR
012046  012704  100000                  MOV     #BIT15,R4       ;SET UP S/B DATA
012052  156704  016166                  BISB    LINE,R4
012056  004767  012324                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
012062  004567  012540                  JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
012066  030244                          LINE
012070  034176                          EM50+53
012072  104053                          ERROR   53              ;TIMED OUT WAITING FOR XMIT DONE
012074  000426                          BR      8$              ;GO TEST NEXT SPEED

012076  016703  016172          5$:     MOV     TIMEB,R3        ;GET THE WAS COUNT
012102  016704  016170                  MOV     TIMEC,R4        ;GET LASTR CHECK COUNT
012106  020304                          CMP     R3,R4           ;COMPARE RELATIVE TIMES
012110  103420                          BLO     8$              ;BR IF THIS SPEED FASTER THAN LAST
                                                                ;SPEED TESTED

012112  004767  015042          7$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
012116  016702  167076                  MOV     $TMP7,R2        ;GET SPEED CODE AND RIGHT JUSTIFY
012122  000302                          SWAB    R2
012124  006202                          ASR     R2
012126  006202                          ASR     R2
012130  042702  177760                  BIC     #177760,R2      ;STRIP AWAY ALL JUNK
012134  004767  012246                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
012140  004567  012462                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
012144  030244                          LINE
012146  031541                          EM17+41
012150  104017                          ERROR   17              ;TRANSMITTER SPEED INCORRECT

012152  016767  016116  016116  8$:     MOV     TIMEB,TIMEC     ;SET UP NEW CHECK TIMER COUNT
```

```
012160  062767  002100  167032          ADD     #2100,$TMP7     ;GENERATE NEXT SPEED
012166  022767  035600  167024          CMP     #35600,$TMP7    ;DONE ALL SPEEDS ?
012174  001253                          BNE     2$              ;BR IF NOT
012176  000641                          BR      1$              ;GO TEST NEXT LINE
```

```
                                        ;;*****************************************************************
                                        ;*TEST 41       RECEIVER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS
                                        ;;*****************************************************************
012200  000004                          TST41:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        **************

                                                THIS TEST IS IDENTICAL TO TEST 40 EXCEPT IT WAITS FOR "DATA READY"
                                        TO CHECK RECEIVER TIMING. THE SEQUENCE IS SIMILAR AND THE SAME TIMERS
                                        ARE USED FOR ERROR CHECKING.

                                        ERRORS:
                                        *******

                                        1.  [ERROR 54] IS CALLED TO REPORT RCVR TIMEOUT ERRORS
                                        2.  [ERROR 20] IS CALLED TO REPORT RCVR TIMING ERRORS

                                        SYNC:   M7277   SH3     INIT A H        EF2
                                        *****

                                        DEBUG:          (SAME AS TEST 42)
                                        ******

                                        KEY LOGIC:      (SAME AS TEST 42 PLUS)
                                        **********

                                                M7288   SH5,7,9,11      RX CLOCK NN L SIGNALS
                                                M7280   BUF DA LINE "N" UART PIN 19
                                                        RX CLOCK LINE "N" UART PIN 17

                                        %
012202  012767  012232  166700                  MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
012210  004767  012320          1$:             JSR     PC,$ELINE       ;GO SELECT A LINE TO TEST
012214  000534                                  BR      TST42           ;;BR IF TESTED ALL SELECTED LINES
012216  012767  002100  166774                  MOV     #2100,$TMP7     ;INIT TO START WITH LOWEST SPEED
012224  012767  177777  016044                  MOV     #-1,TIMEC       ;INIT RELATIVE TIME CHECKER
012232  012711  004000          2$:             MOV     #BIT11,(R1)     ;CLEAR THE DH11
012236  012711  001000                          MOV     #BIT9,(R1)      ;SET MAINTENANCE MODE ENABLE
012242  156711  015776          3$:             BISB    LINE,(R1)       ;SELECT IT IN THE SCR
012246  012761  177777  000010                  MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO XFER 1 CHAR
012254  005061  000006                          CLR     CAR(R1)         ;GET TEST DATA STARTING AT LOC. 0
012260  016761  166734  000004                  MOV     $TMP7,LPR(R1)   ;SELECT A XMIT SPEED
012266  016761  014744  000012                  MOV     LINMSK,BAR(R1)  ;ACTIVATE THE TRANSMITTER

012274  012767  000001  015770                  MOV     #1,TIMEA        ;INIT TIMER A
012302  005067  015766                          CLR     TIMEB           ;INIT TIMER B
012306  105711                  4$:             TSTB    (R1)            ;RCVR DONE YET ??
012310  100435                                  BMI     5$              ;BR IF YES
012312  004767  014460                          JSR     PC,TIMEIT       ;CALL THE TIMER
012316  000773                                  BR      4$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                        ;AROUND THIS BRANCH IF TIME OUT OCCURS

012320  016767  166674  166654                  MOV     $TMP7,$TMP0     ;SAVE AND SET UP THE SPEED CODE
012326  006367  166650                          ASL     $TMP0
012332  006367  166644                          ASL     $TMP0
```

```
012336  000367  166640                  SWAB    $TMP0
012342  042767  177760  166632          BIC     #177760,$TMP0
012350  011103                          MOV     (R1),R3          ;GET THE WAS DATA
012352  010102                          MOV     R1,R2            ;MAKE REGADR = DEVADR
012354  012704  100200                  MOV     #BIT15+BIT07,R4  ;SET UP S/B DATA
012360  156704  015660                  BISB    LINE,R4
012364  004767  012016                  JSR     PC,SUER2A        ;GO SET UP ERROR INFO
012370  004567  012232                  JSR     R5,SUNUM         ;GO SET LINE NO. IN MSG
012374  030244                          LINE
012376  032020                          EM22+51
012400  104054                          ERROR   54               ;TIMED OUT WAITING FOR CHAR AVAIL
012402  000426                          BR      8$               ;GO TEST NEXT SPEED

012404  016703  015664          5$:     MOV     TIMEB,R3         ;GET THE WAS COUNT
012410  016704  015662                  MOV     TIMEC,R4         ;GET THE CHECK COUNT
012414  020304                          CMP     R3,R4            ;COMPARE RELATIVE TIMES
012416  103420                          BLO     8$               ;BR IF TIME INDICATES THIS SPEED FASTER
                                                                 ;THAN LAST SPEED

012420  004767  014534          7$:     JSR     PC,SAPS          ;SAVE THE ERROR PSW
012424  016702  166570                  MOV     $TMP7,R2         ;GET SPEED CODE AND RIGHT JUSTIFY
012430  006302                          ASL     R2
012432  006302                          ASL     R2
012434  000302                          SWAB    R2
012436  042702  177760                  BIC     #177760,R2       ;STRIP AWAY ALL JUNK
012442  004767  011740                  JSR     PC,SUER2A        ;GO SET UP ERROR INFO
012446  004567  012154                  JSR     R5,SUNUM         ;GO PUT LINE NO. IN MSG
012452  030244                          LINE
012454  031700                          EM20+36
012456  104020                          ERROR   20               ;RECEIVER SPEED INCORRECT

012460  016767  015610  015610   8$:    MOV     TIMEB,TIMEC      ;SET UP NEW CHECK TIMER COUNT
012466  062767  002100  166524          ADD     #2100,$TMP7      ;GENERATE NEXT SPEED
012474  022767  035600  166516          CMP     #35600,$TMP7     ;DONE ALL SPEEDS ?
012502  001253                          BNE     2$               ;BR IF NOT
012504  000641                          BR      1$               ;GO TEST NEXT LINE
```

6021

```
                                ;;*****************************************************************
                                ;*TEST 42        VERIFY STORAGE OVERFLOW - NON MAINT. MODE - ALL SELECTED LINES
                                ;;*****************************************************************
012506  000004                  TST42:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT THE STORAGE OVERFLOW BIT (SCR14) SETS
                                AND CLEARS PROPERLY FOR ALL SELECTED LINES. THE TEST SEQUENCE IS AS
                                FOLLOWS:

                                        1  SET UP THE ERROR RETURN
                                        2. SELECT A LINE NUMBER TO TEST - GO TO NEXT TEST IF ALL
                                           SELECTED LINES HAVE BEEN TESTED.
                                        3. PRIME THE SELECTED LINE TO XMIT 65(10) CHARS.
                                        4. ACTIVATE THE SELECTED LINE AND WAIT FOR STORAGE OVERFLOW
                                           TO SET (SCR14=1)
                                        5. IF SCR14 FAILS TO SET ON TIME - REPORT ERROR AND THEN
                                           CONTINUE WITH THE NEXT LINE (STEP 2).
                                        6. IF IT SETS OK - READ THE "NRC" REG TWICE TO EMPTY TWO
                                           WORDS FROM THE SILO.
                                        7. AFTER A BRIEF STALL, VERIFY THAT SCR14 HAS CLEARED -
                                           IF NOT REPORT ERROR AND CONTINUE WITH NEXT LINE (STEP2)
                                        8. IF IT CLEARS OK, VERIFY THAT THE FILL COUNT (SSR<15:08>)
                                           CONTAINS A 77(8) - IF NOT REPORT ERROR AND CONTINUE
                                           WITH NEXT LINE  (STEP 2).
                                        9. REPEAT STEPS 2 THRU 8 UNTIL ALL SELECTED LINES HAVE BEEN
                                           TESTED.

                                ERRORS:
                                *******

                                1. [ERROR  57] IS CALLED TO REPORT ALL ERRORS DETECTED.


                                SYNC:   M7277   SH3     INIT A  H       EF2
                                *****

                                DEBUG:
                                ******

                                1. PROBLEM CAN VERY LIKELY BE THAT NO TEST CONNECTOR IS PRESENT.
                                   BE SURE YOU HAVE ONE ON FOR THIS TEST.

                                2. PROBLEM IS POSSIBLY ON THE M7289 MODULE (SH4) OR SOME
                                   SIGNAL FEEDING THIS LOGIC.

                                KEY LOGIC:
                                **********

                                        M7289   SH4     STORAGE OVERFLOW L      E43-12
                                                        READY IN PULSE H        E40-11
                                                        UC1 MASTER DA H         BH2
                                                        UC2 MASTER DA H         BD2
```

```
                                        %
012510  012767  012524  166372           MOV     #2$,$LPERR      ;SET UP ERROR RETURN
012516  004767  012012          1$:     JSR     PC,$ELINE       ;GO SELECT A LINE NO. TO TEST
012522  000535                          BR      TST43           ;;BR IF DONE ALL SELECTED LINES
012524  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
012530  010102                          MOV     R1,R2           ;MAKE REGADR = DEVADR
012532  116711  015506                  MOVB    LINE,(R1)       ;SELECT A LINE
012536  005061  000006                  CLR     CAR(R1)         ;SET UP CURRENT ADDRESS
012542  012761  177677  000010          MOV     #-65.,BCR(R1)   ;SET UP BYTE COUNT
012550  012761  033500  000004          MOV     #33500,LPR(R1)  ;SET UP LPR: 9600 BAUD,5 BIT CHARS
012556  016761  014454  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE

012564  012767  000001  015500          MOV     #1,TIMEA        ;INIT TIMERS
012572  005067  015476                  CLR     TIMEB
012576  032711  040000          3$:     BIT     #BIT14,(R1)     ;STOREGE OVERFLOW YET ??
012602  001024                          BNE     4$              ;BR IF YES YOU SHOULD GET IT
012604  004767  014166                  JSR     PC,TIMEIT       ;CALL TIMER
012610  000772                          BR      3$              ;BR IF NO TIME OUT

012612  004767  014342                  JSR     PC,SAPS         ;GO SAVE PSW
012616  012704  040000                  MOV     #BIT14,R4       ;SET UP S/B DATA
012622  156704  015416                  BISB    LINE,R4
012626  011103                          MOV     (R1),R3         ;SET UP WAS DATA
012630  042703  137760                  BIC     #137760,R3      ;CLEAR UNINTERESTING BITS
012634  004767  011546                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
012640  004567  011762                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
012644  030244                          LINE
012646  034663                          EM57+44
012650  104057                          ERROR   57              ;STORAGE OVERFLOW FAILED TO SET
012652  000721                          BR      1$              ;GO TRY NEXT LINE

012654  016167  000002  166336  4$:     MOV     NRC(R1),$TMP7   ;READ THE SILO
012662  016167  000002  166330          MOV     NRC(R1),$TMP7   ;READ IT AGAIN
012670  012705  001000                  MOV     #1000,R5        ;INIT STALL COUNTER
012674  005305                  41$:    DEC     R5              ;COUNT TIMER
012676  001376                          BNE     41$             ;BR IF NO TIMEOUT
012700  032711  040000                  BIT     #BIT14,(R1)     ;DID OVERFLOW GO AWAY ?
012704  001420                          BEQ     5$              ;BR IF YES

012706  004767  014246                  JSR     PC,SAPS         ;GO SAVE THE PSW
012712  005004                          CLR     R4              ;SET UP S/B DATA
012714  156704  015324                  BISB    LINE,R4
012720  011103                          MOV     (R1),R3         ;SET UP WAS DATA
012722  042703  137760                  BIC     #137760,R3      ;CLEAR UNINTERESTING BITS
012726  004767  011454                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
012732  004567  011670                  JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
012736  030244                          LINE
012740  034663                          EM57+44
012742  104057                          ERROR   57              ;STORAGE BIT FAILED TO CLEAR
012744  000664                          BR      1$              ;GO TRY NEXT LINE

012746  122761  000077  000017  5$:     CMPB    #77,SSR+1(R1)   ;WAS IT REALLY 65. ??
012754  001660                          BEQ     1$

012756  004767  014176                  JSR     PC,SAPS         ;GO SAVE PSW
012762  012704  037400                  MOV     #37400,R4       ;SET UP S/B DATA
```

```
012766  062702  000016          ADD     #SSR,R2         ;SET UP REGADR
012772  016103  000016          MOV     SSR(R1),R3      ;SAVE WAS DATA
012776  004767  011404          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
013002  004567  011620          JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
013006  030244                  LINE
013010  034663                  EM57+44
013012  104057                  ERROR   57              ;READING SILO FAILED TO DEC SSR OR
                                                        ;STORAGE OVFL SET AT WRONG COUNT
013014  000640                  BR      1$              ;GO TRY NEXT LINE
```

```
                                ;;*****************************************************************
                                ;;*TEST 43        BASIC DATA TEST - ALL SELECTED LINES/ALL CHAR LENGTHS
                                ;;*****************************************************************
                                ;;
013016  000004                  TST43:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT A SINGLE ALL ONES CHAR. CAN BE TRANS-
                                MITTED AND RECEIVED ON ALL SELECTED LINES AT ALL FOUR CHAR LENGTHS
                                (5, 6, 7, AND 8 BITS). THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SET UP THE ERROR LOOP RETURN
                                        2. SELECT A LINE NO. TO TEST - GO TO THE NEXT TEST IF DONE ALL
                                           SELECTED LINES.
                                        3. GET A TEST CHARACTER FROM THE DATA TABLE AND UPDATE THE
                                           TABLE POINTER.
                                        4. CLEAR THE DH11
                                        5. PRIME THE SELECTED LINE TO XMIT ONE CHAR AT 9600 BAUD
                                        6. WAIT FOR "CHAR AVAIL" TO SET - IF TIMEOUT REPORT ERROR AND
                                           RESTART AT STEP 8.
                                        7. IF NO TIMEOUT - CHECK DATA AND REPORT ANY ERRORS
                                        8. INCREMENT "SCR" REG TO CHANGE CHAR LENGTH - IF DONE
                                           ALL FOUR GO TO STEP 2 - IF NOT THEN STEP 4.

                                ERRORS:
                                ******

                                1. [ERROR  55] IS CALLED TO REPORT RCVR TIMEOUT.
                                2. [ERROR  23] IS CALLED TO REPORT DATA COMPARE ERRORS

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                ******

                                1. IF FAULT AFFECTS ONLY ONE LINE AT ALL CHAR LENGTHS, SUSPECT A
                                   BAD UART MODULE M7280.

                                2. IF FAULT AFFECTS ONLY ONE BIT ON ALL LINES, SUSPECT THE
                                    THE M7279 MODULE.
                                3. IF FAULT AFFECTS ONLY CERTAIN CHAR LENGTHS, SUSPECT EITHER THE M7278
                                   OR THE UART MODULE M7280.

                                KEY LOGIC:
                                **********

                                        M7280'S UART CHIPS PINS <12:05>

                                        M7279   SH1     E1,E2,E6, OR E7

                                        M7278   SH8     NB2 LPR 01 H    FH1
                                                        NB1 LPR 00 H    FH2
                                %
013020  012767  013046  166062          MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN
```

```
013026  004767  011502          1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
013032  000511                          BR      TST44           ;;BR IF DONE ALL SELECTED LINES
013034  012705  030260                  MOV     #TDATA2,R5      ;GET POINTER TO DATA TABLE
013040  005002                          CLR     R2              ;INIT R2 TO START AT CHAR LENGTH OF 5 BITS
013042  012567  166152          2$:     MOV     (R5)+,$TMP7     ;PUT TEST CHAR IN XMIT BUFFER
013046  012711  004000          3$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
013052  156711  015166                  BISB    LINE,(R1)       ;SELECT THE LINE
013056  012761  177777  000010          MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO -1
013064  012761  001220  000006          MOV     #$TMP7,CAR(R1)  ;SET CURRENT ADDRESS REG
013072  012761  033500  000004          MOV     #33500,LPR(R1)  ;SET BAUD RATE TO 9600
013100  050261  000004                  BIS     R2,LPR(R1)      ;SELECT CHAR LENGTH
013104  056761  014126  000012          BIS     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE

013112  012767  000001  015152          MOV     #1,TIMEA        ;INIT TIMER A
013120  005067  015150                  CLR     TIMEB           ;INIT TIMER B
013124  105711                  4$:     TSTB    (R1)            ;RCVR DONE YET ??
013126  100424                          BMI     5$              ;BR IF YES
013130  004767  013642                  JSR     PC,TIMEIT       ;CALL THE TIMER
013134  000773                          BR      4$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS

013136  004767  014016                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
013142  011103                          MOV     (R1),R3         ;GET THE SCR
013144  042703  177560                  BIC     #177560,R3      ;CLEAR UNINTERESTING BITS
013150  012704  000200                  MOV     #200,R4         ;SET UP S/B DATA
013154  156704  015064                  BISB    LINE,R4
013160  004767  011222                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
013164  004567  011436                  JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
013170  030244                          LINE
013172  032020                          EM22+51
013174  104055                          ERROR   55              ;CHAR AVAIL FAILED TO SET ON TIME
013176  000422                          BR      6$              ;GO TEST NEXT CHAR LENGTH

013200  016103  000002          5$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
013204  012704  000200                  MOV     #200,R4         ;SET UP THE S/B DATA IN R4
013210  156704  015030                  BISB    LINE,R4
013214  000304                          SWAB    R4
013216  156704  165776                  BISB    $TMP7,R4
013222  020304                          CMP     R3,R4           ;WAS THE RCVD DATA CORRECT ??
013224  001407                          BEQ     6$              ;BR IF YES

013226  004767  011150                  JSR     PC,SUER2        ;GO SET UP THE ERROR INFO
013232  004567  011370                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
013236  030244                          LINE
013240  032061                          EM23+36
013242  104023                          ERROR   23              ;DATA COMPARE ERROR

013244  005202          6$:             INC     R2              ;DO NEXT CHAR LENGTH ON SELECTED LINE
013246  022702  000004                  CMP     #4,R2           ;HAVE WE DONE ALL FOUR CHAR LENGTHS ??
013252  001273                          BNE     2$              ;BR IF NOT
013254  000664                          BR      1$              ;GO DO NEXT LINE
```

```
                                ;;*****************************************************************
                                ;*TEST 44        SINGLE LINE DATA TEST - ALL SELECTED LINES
                                ;;*****************************************************************
013256  000004                  TST44:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST TRANSMITS AND RECEIVES A BINARY COUNT PATTERN
                                (000 - 377) ON ALL SELECTED LINES. THE TEST SEQUENCE IS AS
                                FOLLOWS:

                                        1. SET UP THE ERROR LOOP RETURN
                                        2. GO SELECT A LINE NO. TO TEST - IF DONE ALL SELECTED
                                           LINES THEN GO TO THE NEXT TEST .
                                        3. CLEAR THE DH11 AND PRIME THE SELECTED LINE TO XMIT
                                           TO XMIT 256. CHARS AT 9600. BAUD - 8 BIT CHARS.
                                        4. SET UP R5 TO POINT TO RCVR CORE BUFFER.
                                        5. ACTIVATE THE SELECTED XMITTER.
                                        6. WAIT FOR "CHAR AVAIL" TO SET BEFORE READING THE SILO.
                                           IF RCVR TIMEOUT REPORT ERROR AND RESTART AT STEP 2.
                                        7. IF NO TIMEOUT READ THE SILO AND STORE THE WORD IN THE RCVR
                                           CORE BUFFER - WHEN THE BUFFER IS FULL GO TO STEP8 IF NOT
                                           THEN GO TO STEP 6.
                                        8. COMPARE THE XMIT AND RCVR CORE IMAGE BUFFERS AND REPORT
                                           ALL DATA COMPARE ERRORS.
                                        9. CHECK THE "BAR", "BCR", AND "CAR" REGISTERS FOR CORRECT
                                           CONTENTS - REPORT ALL ERRORS.
                                        10.GO TO STEP 2


                                ERRORS:
                                *******

                                1. [ERROR  22] IS CALLED TO REPORT "DATA AVAIL" TIMEOUT
                                2. [ERROR  37] "        "        " DATA COMPARE ERRORS
                                3. [ERROR  40] "        "        " "BAR" REG NOT CLEARED
                                4. [ERROR  10] "        "        " "BCR" REG NOT ALL ZEROES
                                5. [ERROR   7] "        "        " "CAR" REG NOT UPDATED CORRECTLY

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                *******

                                1. IF THE FAULT AFFECTS ONE OR MORE LINES IN AN 8 LINE GROUP <15:08>
                                   OR <07:00>, SWAP THE M7280 MODULES. IF THE FAULT SHIFTS SO THAT
                                   THE ERROR INDICATES DIFFERENT LINES THE PROBLEM IS MOST LIKELY
                                   THE M7280 THE SYMPTOM SHIFTED TO.

                                2. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS ON
                                   ONE LINE THE FAULT IS MOST LIKELY A "UART" CHIP.

                                3. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS
                                   ON ALL LINES SUSPECT THE DATA PATHS EXTERNAL TO, THE M7280 MODULES.
```

```
                                        4. IF THE FAULT CAUSES NO DATA ERRORS BUT GIVES "BAR", "BCR", OR "CAR" ERRORS
                                           SUSPECT THE M7278 OR M7277 MODULES.

                                        KEY LOGIC       (REFER TO TEST 43)
                                        ***********

                                        %
013260  012767  013302  165622                  MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
013266  004767  011242                  1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
013272  000401                                  BR      11$             ;;BR IF ALL SELECTED LINES DONE
013274  000402                                  BR      2$              ;GO TEST IT
013276  000167  000424                  11$:    JMP     8$              ;EXIT TEST
013302  016701  013716                  2$:     MOV     DHADR,R1        ;RESET DEVADR
013306  012711  004000                          MOV     #BIT11,(R1)     ;CLEAR THE DH11
013312  156711  014726                          BISB    LINE,(R1)       ;SET SELECT BITS IN SCR
013316  012761  037244  000006                  MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDRESS REG
013324  012761  177400  000010                  MOV     #-400,BCR(R1)   ;SET UP BYTE COUNT
013332  012761  033503  000004                  MOV     #33503,LPR(R1)  ;SET LINE PARAMETERS
013340  012705  036244                          MOV     #RBUF,R5        ;SET UP POINTER TO INPUT DATA BUFFER
013344  052711  001000                          BIS     #BIT09,(R1)     ;SET MAINT MODE BIT
013350  016761  013662  000012                  MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE

013356  012767  000002  014706                  MOV     #2,TIMEA        ;INIT TIMER A
013364  005067  014704                          CLR     TIMEB           ;INIT TIMER B
013370  105711                          3$:     TSTB    (R1)            ;RCVR DONE YET ??
013372  100425                                  BMI     4$              ;BR IF YES
013374  004767  013376                          JSR     PC,TIMEIT       ;CALL THE TIMER
013400  000773                                  BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                        ;AROUND THIS BRANCH IF TIME OUT OCCURS

013402  004767  013552                          JSR     PC,SAPS         ;SAVE THE ERROR PSW
013406  010102                                  MOV     R1,R2           ;SET UP REGADR
013410  011203                                  MOV     (R2),R3         ;GET THE WAS DATA
013412  042703  176400                          BIC     #176400,R3      ;CLEAR JUNK BITS
013416  012704  001200                          MOV     #1200,R4        ;SET UP S/B DATA
013422  156704  014616                          BISB    LINE,R4
013426  004767  010754                          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
013432  004567  011170                          JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
013436  030244                                  LINE
013440  032020                                  EM22+51
013442  104022                                  ERROR   22              ;CHAR AVAIL TIMEOUT
013444  000710                                  BR      1$              ;GO TRY NEXT LINE

013446  016125  000002                  4$:     MOV     NRC(R1),(R5)+   ;SAVE THE RECEIVED DATA
013452  022705  037244                          CMP     #RBUF+1000,R5   ;INPUT BUFFER FULL ??
013456  001344                                  BNE     3$              ;BR IF NOT

013460  012702  037244                          MOV     #TBUF,R2        ;SET UP POINTER TO OUTPUT BUFFER
013464  012701  036244                          MOV     #RBUF,R1        ;SET UP POINTER TO INPUT BUFFER
013470  111204                          5$:     MOVB    (R2),R4         ;SET UP S/B DATA IN R4
013472  042704  177400                          BIC     #177400,R4
013476  000304                                  SWAB    R4
013500  156704  014540                          BISB    LINE,R4
013504  152704  000200                          BISB    #200,R4
013510  000304                                  SWAB    R4
```

```
013512  011103                          MOV     (R1),R3         ;GET THE WAS DATA
013514  020304                          CMP     R3,R4           ;DATA CORRECT ??
013516  001407                          BEQ     6$              ;BR IF YES

013520  004767  010656                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
013524  004567  011076                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
013530  030244                          LINE
013532  033276                          EM37+33
013534  104037                          ERROR   37              ;DATA COMPARE ERROR


013536  005202                  6$:     INC     R2              ;UPDATE DATA BUFFER POINTERS
013540  062701  000002                  ADD     #2,R1           ;
013544  022701  037244                  CMP     #RBUF+1000,R1   ;COMPARED ALL 256. CHARS ??
013550  001347                          BNE     5$              ;BR IF NOT

013552  016701  013446                  MOV     DHADR,R1        ;RESET DEVADR
013556  010102                          MOV     R1,R2           ;SET UP REGADR
013560  062702  000012                  ADD     #BAR,R2
013564  005712                          TST     (R2)            ;WAS THE "BAR" ALL ZEROES ??
013566  001413                          BEQ     7$              ;BR IF YES

013570  004767  013364                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
013574  011203                          MOV     (R2),R3         ;GET THE WAS DATA
013576  005004                          CLR     R4              ;SET UP S/B DATA
013600  004767  010602                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
013604  004567  011016                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
013610  030244                          LINE
013612  033341                          EM40+40
013614  104040                          ERROR   40              ;"BAR" REG NOT ALL ZEROES

013616  010102                  7$:     MOV     R1,R2           ;SET UP REGADR
013620  062702  000010                  ADD     #BCR,R2
013624  005712                          TST     (R2)            ;BYTE COUNT REG ALL ZEROES ?
013626  001413                          BEQ     71$             ;;BR IF BYTE COUNT ZERO

013630  004767  013324                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
013634  011203                          MOV     (R2),R3         ;GET THE WAS DATA
013636  005004                          CLR     R4              ;SET UP THE S/B DATA
013640  004767  010542                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
013644  004567  010756                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
013650  030244                          LINE
013652  031146                          EM10+44
013654  104010                          ERROR   10              ;BYTE COUNT NOT ALL ZEROES

013656  010102                  71$:    MOV     R1,R2           ;SET UP REGADR
013660  062702  000006                  ADD     #CAR,R2
013664  022712  037644                  CMP     #TBUF+400,(R2)  ;DID "CAR" INCREMENT PROPERLY ?
013670  001414                          BEQ     72$             ;BR IF YES

013672  004767  013262                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
013676  011203                          MOV     (R2),R3         ;GET THE WAS DATA
013700  012704  037644                  MOV     #TBUF+400,R4    ;SET UP S/B DATA
013704  004767  010476                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
013710  004567  010712                  JSR     R5,SUNUM        ;GO PUT LINE NO IN MESSAGE
013714  030244                          LINE
```

```
013716  031077                          EM7+47
013720  104007                          ERROR   7               ;"CAR" NOT UPDATED CORRECTLY

013722  000167  177340          72$:    JMP     1$              ;GO DO NEXT LINE

013726  000240                  8$:     NOP                     ;EXIT POINT
```

```
                                        ;;*****************************************************************
                                        ;*TEST 45       BASIC PARITY LOGIC TEST - ALL SELECTED LINES - ODD PARITY
                                        ;;*****************************************************************
013730  000004                          TST45:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        ****************

                                                THIS TEST VERIFIES THE ODD PARITY FUNCTION FOR ALL SELECTED
                                        LINES USING THE "BREAK" FUNCTION TO FORCE PARITY ERRORS. REFER TO
                                        THE FLOW CHARTS IN THE PROGRAM DOCUMENTATION FOR TEST SEQUENCES.

                                        ERRORS:
                                        *******

                                        1. [ERROR  22] IS CALLED TO REPORT RCVR TIMEOUT
                                        2. [ERROR  33] IS CALLED TO REPORT DATA/PARITY ERRORS

                                        SYNC:   M7277   SH3     INIT A H        EF2
                                        *****

                                        DEBUG:
                                        ******

                                        1. IF FAULT AFFECTS ALL LINES SUSPECT THE M7278 MODULE.

                                        2. IF IT AFFECTS ONLY ONE LINE SUSPECT THE "UART" MODULE FOR THAT LINE.


                                        KEY LOGIC:
                                        **********

                                                M7278   SH7     PEN LPR04 L     FF2
                                                                PEV LPR05 L     FN1
                                        %
013732  012767  013746  165150                  MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
013740  004767  010570                  1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
013744  000506                                  BR      TST46           ;;BR IF ALL SELECTED LINES DONE
013746  012711  004000                  2$:     MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
013752  012704  000260                          MOV     #260,R4         ;SET UP THE S/B DATA IN R4
013756  156704  014262                          BISB    LINE,R4
013762  000304                                  SWAB    R4
013764  156711  014254                          BISB    LINE,(R1)       ;SET LINE NO. IN SCR
013770  012767  000377  014260                  MOV     #377,TDATA1     ;LOAD XMIT BUFFER WITH TEST CHARACTER
013776  012761  073563  000004                  MOV     #73563,LPR(R1)  ;SET UP THE LINE PARAMETERS
014004  012761  177777  000010                  MOV     #-1,BCR(R1)     ;LOAD THE BYTE COUNT REG
014012  012761  030256  000006                  MOV     #TDATA1,CAR(R1) ;LOAD THE BUS ADDR REG
014020  016761  013212  000014                  MOV     LINMSK,BKR(R1)  ;SET BREAK BIT FOR SELECTED LINE
014026  016761  013204  000012                  MOV     LINMSK,BAR(R1)  ;ACTIVATE THE XMITTR

014034  012767  000001  014230                  MOV     #1,TIMEA        ;INIT TIMER A
014042  005067  014226                          CLR     TIMEB           ;INIT TIMER B
014046  105711                          3$:     TSTB    (R1)            ;RCVR DONE YET ??
014050  100423                                  BMI     4$              ;BR IF YES
014052  004767  012720                          JSR     PC,TIMEIT       ;CALL THE TIMER
014056  000773                                  BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
```

```
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS

014060  004767  013074                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
014064  011103                          MOV     (R1),R3         ;GET THE WAS DATA
014066  012704  100200                  MOV     #100200,R4      ;SET UP THE S/B DATA
014072  156704  014146                  BISB    LINE,R4
014076  010102                          MOV     R1,R2           ;SET UP REGADR
014100  004767  010302                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
014104  004567  010516                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
014110  030244                          LINE
014112  032020                          EM22+51
014114  104022                          ERROR   22              ;TIMED OUT WAITING FOR DATA AVAIL
014116  000710                          BR      1$              ;GO TEST NEXT LINE

014120  016103  000002          1$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
014124  020304                          CMP     R3,R4           ;CORRECT DATA RECEIVED ??
014126  001704                          BEQ     1$              ;BR IF YES

014130  004767  013024                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
014134  010102                          MOV     R1,R2           ;SET UP THE REGADR
014136  062702  000002                  ADD     #NRC,R2
014142  004767  010240                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
014146  004567  010454                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
014152  030244                          LINE
014154  033012                          EM33+40
014156  104033                          ERROR   33              ;INCORRECT DATA OR PARITY ERROR
014160  000667                          BR      1$              ;GO TEST NEXT LINE
```

```
                                        ;;*****************************************************************
                                        ;*TEST 46        MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES
                                        ;;*****************************************************************
014162  000004                          TST46:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        **************

                                                THIS TEST VERIFIES ALL SELECTED LINES CAN TRANSMIT AND RECEIVE
                                        A BINARY COUNT PATTERN WHEN RUN CONCURRENTLY. ALL CHAR LENGTHS (5, 6, 7,
                                        AND 8 BITS) ARE TESTED WITH BOTH EVEN AND ODD PARITY CHECKING SPECIFIED.
                                        THE TEST ACTUALLY INCLUDES EIGHT SUB-TESTS - THE PARAMETERS FOR EACH
                                        SUB-TEST RETRIEVED FROM A TABLE TAGGED "PRTYTB:".  REFER TO THIS TABLE
                                        TO DETERMINE THE TEST SEQUENCE.


                                        ERRORS:
                                        ********

                                        1. [ERROR  41] IS CALLED TO REPORT FALSE RECEIVER INTRS.
                                        2. [ERROR  42] IS CALLED TO REPORT SILO OVERFLOW ERRORS
                                        3. [ERROR  34] IS CALLED TO REPORT PARITY/DATA ERRORS
                                        4. [ERROR  35] IS CALLED TO REPORT TEST TIMEOUTS

                                        SYNC:   (NONE)
                                        *****

                                        DEBUG:  (REFER TO TEST 45)
                                        ******

                                        KEY LOGIC:      (REFER TO TEST 45)
                                        ***********

                                        %
014164  012767  014202  164716                  MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
014172  012705  027306                          MOV     #PRTYTB+4,R5    ;SET UP POINTER TO TEST PARAMETERS
014176  005067  165016                          CLR     $TMP7           ;START WITH SUB TEST #00
014202  162705  000004          1$:             SUB     #4,R5           ;RESET POINTER FOR ERROR LOOPS
014206  005367  165006                          DEC     $TMP7           ;RESET SUB TEST # FOR ERROR LOOP
014212  022705  027342          2$:             CMP     #PRTYTB+40,R5   ;DONE ALL 8. SUB TESTS ??
014216  001456                                  BEQ     21$             ;;BR IF YES
014220  012706  001100                          MOV     #STACK,SP       ;RESET STACK POINTER FOR ERROR LOOPS
014224  016701  012774                          MOV     DHADR,R1        ;RESET DEVADR FOR ERROR LOOPS
014230  012567  164762                          MOV     (R5)+,$TMP6     ;GET THE BYTE COUNT PARAMETER
014234  012567  164754                          MOV     (R5)+,$TMP5     ;GET THE LINE PARAMETERS
014240  005267  164754                          INC     $TMP7           ;GENERATE NEW SUB-TEST NO.
014244  012711  004000                          MOV     #BIT11,(R1)     ;CLEAR THE DH11
014250  004767  010534                          JSR     PC,SUPPAR       ;GO SET UP PARAMETERS
014254  016167  000004  164720                  MOV     LPR(R1),$TMP0   ;SAVE CURRENT LINE PARAMETERS
014262  016767  012746  164720                  MOV     LINSEL,$TMP3    ;SAVE SELECTED LINES PARAMETER
014270  004767  012650                          JSR     PC,CHPS2        ;GO LOCK OUT INTRS
014274  016702  012726                          MOV     DHVCT,R2        ;SET UP THE VECTOR
014300  012722  014372                          MOV     #3$,(R2)+       ;GO TO 3$ ON RCVR INTERRUPT
014304  116712  013730                          MOVB    DHRLVL,(R2)
014310  012711  000100                          MOV     #100,(R1)       ;ENABLE CHAR AVAIL INTERRUPTS
014314  016767  012714  013160                  MOV     LINSEL,LINACT   ;FLAG ALL SELECTED LINES ACTIVE
```

```
014322  016761  012706  000012          MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL SELECTEDLINES
014330  116767  164546  164650          MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
014336  042767  177400  164642          BIC     #177400,$TMP2
014344  004767  012560                  JSR     PC,CHPS1        ;GO CLEAR PSW
014350  000167  000236                  JMP     7$              ;GO WAIT FOR INTERRUPTS

014354  012706  001100          21$:    MOV     #STACK,SP       ;RESTORE THE SP
014360  004767  012544                  JSR     PC,CHPS1        ;GO CLEAR PSW
014364  004767  012360                  JSR     PC,RESTRP       ;RESTORE TRAP CATCHER
014370  000556                          BR      TST47           ;;GO TO NEXT TEST

                                ;RECEIVER INTERRUPT SERVICE ROUTINE

014372  005067  164614          3$:     CLR     $TMP4
014376  105711                          TSTB    (R1)            ;CHAR AVAIL SET
014400  100404                          BMI     4$              ;BR IF YES

014402  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
014406  104041                          ERROR   41              ;RCVR FALSE INTERRUPT - CHAR AVAIL NOT SET
014410  000700                          BR      2$              ;GO TRY NEXT SUB TEST

014412  032711  040000          4$:     BIT     #BIT14,(R1)     ;SILO OVERFLOW ??
014416  001404                          BEQ     5$              ;BR IF NOT

014420  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
014424  104042                          ERROR   42              ;SILO OVERFLOW ERROR
014426  000671                          BR      2$              ;GO TRY NEXT SUB TEST

014430  016103  000002          5$:     MOV     NRC(R.),R3      ;GET THE WAS DATA
014434  010302                          MOV     R3,R2           ;EXTRACT AND SAVE LINE NO.
014436  000302                          SWAB    R2
014440  042702  177760                  BIC     #177760,R2
014444  006302                          ASL     R2              ;GENERATE TABLE OFFSETR
014446  005267  164540                  INC     $TMP4
014452  022767  000101  164532          CMP     #101,$TMP4
014460  001744                          BEQ     3$
014462  036267  030170  012544          BIT     $LNSEL(R2),LINSEL       ;IS THIS ONE OF THE SELECTED LINES?
014470  001002                          BNE     51$             ;IF SO,GO ANALYZE THE CHARACTER
014472  104000                          ERROR                   ;INDICATE SOME KIND OF ERROR
014474  000755                          BR      5$              ;CHECK THE NEXT SILO ENTRY
014476  010267  164510          51$:    MOV     R2,$TMP4
014502  006267  164504                  ASR     $TMP4
014506  026203  036244                  CMP     RBUF(R2),R3     ;CORRECT DATA RECEIVED ??
014512  001426                          BEQ     6$              ;BR IF YES

014514  004767  012440                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
014520  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
014524  016204  036244                  MOV     RBUF(R2),R4     ;SET UP S/B DATA
014530  062701  000002                  ADD     #NRC,R1         ;SET UP WAS ADDRESS
014534  062702  036244                  ADD     #RBUF,R2        ;SET UP S/B ADDRESS
014540  004767  007642                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
014544  004567  010056                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
014550  001212                          $TMP4
014552  033066                          EM34+51
014554  004567  010046                  JSR     R5,SUNUM        ;PUT SUBTEST NO. IN MESSAGE
014560  001220                          $TMP7
```

```
014562  033104                          EM34+67
014564  104034                          ERROR   34              ;PARITY DATA COMPARE ERROR
014566  000611                          BR      2$              ;GO TRY NEXT SUBTEST

014570  105262  036244          6$:     INCB    RBUF(R2)        ;GENERATE NEW RCVD DATA
014574  005262  027504                  INC     MULPTB(R2)      ;COUNT ONE BYTES RECEIVED
014600  001003                          BNE     61$             ;BR IF NOT DONE
014602  046267  027442  012672          BIC     LINBIT(R2),LINACT ;FLAG THIS LINE DONE
014610  000002                  61$:    RTI                     ;RETURN TO WAIT ROUTINE

                                ;WAIT ROUTINE

014612  012767  000002  013452  7$:     MOV     #2,TIMEA        ;INIT TIMER A
014620  005067  013450                  CLR     TIMEB           ;INIT TIMER B
014624  005761  000012          8$:     TST     BAR(R1)         ;ALL LINES DONE XMITTING ??
014630  001413                          BEQ     9$              ;BR IF YES
014632  004767  012140                  JSR     PC,TIMEIT       ;CALL THE TIMER
014636  000772                          BR      8$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS

014640  016167  000012  164342          MOV     BAR(R1),$TMP3   ;SAVE THE ACTIVE LINES FLAG
014646  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
014652  104035                          ERROR   35              ;TIMED OUT WAITING FOR TRANSMITTERS TO FINISH
014654  000167  177332                  JMP     2$              ;GO TRY NEXT SUBTEST


014660  012767  000001  013404  9$:     MOV     #1,TIMEA        ;INIT TIMER A
014666  005067  013402                  CLR     TIMEB           ;INIT TIMER B
014672  005767  012604          10$:    TST     LINACT          ;ALL CHARS RECEIVED ?
014676  001411                          BEQ     11$             ;BR IF YES
014700  004767  012072                  JSR     PC,TIMEIT       ;CALL THE TIMER
014704  000772                          BR      10$             ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS


014706  016767  012570  164274          MOV     LINACT,$TMP3    ;SET UP ACTIVE LINE PARAMETER
014714  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
014720  104035                          ERROR   35              ;SILO EMPTY TIMEOUT
014722  000167  177264          11$:    JMP     2$              ;GO TRY NEXT SUB TEST
```

```
                                ;;*****************************************************************
                                ;*TEST 47        AUTO ECHO TEST 1 - ALL SELECTED LINES
                                ;;*****************************************************************
014726  000004                  TST47:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT ALL SELECTED LINES CAN'TURN AROUND
                                A SINGLE TEST CHARACTER (377) AND (000) IN AUTO ECHO MODE. THE TEST SEQUENCE
                                IS AS FOLLOWS:

                                        1. SET UP THE ERROR LOOP RETURN
                                        2. RETRIEVE THE AUTO-ECHO TEST DATA FROM "AETAB:" AND
                                           UPDATE THE POINTER.
                                        3. GO SELECT A LINE NO. TO TEST - GO TO STEP 10 IF DONE
                                           ALL SELECTED LINES.
                                        4. CLEAR THE "CAR" AND "BCR" MEMORIES.
                                        5. PRIME THE SELECTED LINE TO XMIT ONE CHAR WITH A.E. ENABLED.
                                        6. ACTIVATE THE SELECTED TRANSMITTER.
                                        7. WAIT FOR "CHAR AVAIL" - IF TIMEOUT REPORT ERROR AND RESTART
                                           AT STEP 2.
                                        8. IF NO TIMEOUT - READ SILO AND COMPARE AUTO ECHO DATA RECEIVED
                                           REPORT DATA COMPARE ERRORS AND RESTART AT STEP 2
                                        9. IF NO ERRORS REPEAT STEPS 7 AND 8 SIXTY-FOUR TIMES THEN TURN
                                           OFF A.E. ENABLE AND READ LAST CHAR FROM SILO.. CHECK LAST
                                           CHAR FOR DATA COMPARE ERRORS - REPORT ERRORS IF ANY - AND RE-
                                           START AT  TEP 2.
                                        10.CHANGE A.._. TABLE POINTER TO POINT TO "AETABO:" (0'S DATA)
                                           AND REPEAT STEPS 2 THRU 9.

                                ERRORS:
                                ********

                                1. [ERROR 24] IS CALLED TO REPORT ALL ERRORS

                                SYNC:   M7277   SH4     LOAD BAR LB+HB L        CN2
                                *****

                                DEBUG:
                                ******

                                1. IF ALL LINES FAIL, SUSPECT EITHER THE M7277 OR M7289

                                2. IF ONLY ONE LINE FAILS SUSPECT THE M7288

                                3. LOOP ON THE FAILING LINE AND TRACK BACK THROUGH THE KEY LOGIC.


                                KEY LOGIC:
                                ***********

                                        M7277   SH3     AE GO L         CS2
                                                        7402 "OR" GATE CHIPS E38 OR E41
                                                        74157 MUX CHIPS  E39 OR 342
                                                SH4     E35 - PIN 2 STUCK LOW
```

```
                                                M7289   SH3     AE GO L          EK1
                                                                AE SCAN MUX  E22 PIN 10
                                                        SH4     SAMPLE STATUS H E21-12

                                                M7288   SH5, 7, 9, 11
                                                                AE ENABLE "NN" H CONTROL FLOPS
                                                                74174 CHIPS PIN 15

                                        %
014730  012767  015012  164152                  MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
014736  005067  164246                          CLR     $TMP3           ;INIT I/O DATA F
014742  012711  004000                          MOV     #BIT11,(R1)     ;CLEAR THE DH
014746  012705  027342                          MOV     #AETAB,R5       ;GET POINTER  ECHO DATA TABLE
014752  005067  012262          7$:             CLR     LMSK1           ;INIT BIT TES
014756  000261                                  SEC                     ;SET "C" BIT F  KER
014760  000401                                  BR      13$             ;GO SHIFT MASK
014762  000241                  1$:             CLC                     ;INIT THE "C" BI
014764  006167  012250          13$:            ROL     LMSK1           ;SHIFT BIT MARKER
014770  001407                                  BEQ     12$             ;BR IF DONE ALL LINES
014772  012504                                  MOV     (R5)+,R4        ;SET UP THE S/B DATA
014774  036767  012240  012232                  BIT     LMSK1,LINSEL    ;TEST THIS LINE ?
015002  001767                                  BEQ     1$              ;BR IF NOT
015004  004767  007524          11$:            JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
015010                          12$:
015010  000522                                  BR      6$              ;;BR IF ALL SELECTED LINES TESTED
015012  004767  007670          2$:             JSR     PC,CLCABC       ;GO CLEAR "CAR" AND "BCR" MEMORIES
015016  116711  013222                          MOVB    LINE,(R1)       ;SET SELECT BITS IN SCR REG
015022  012761  177777  000010                  MOV     #-1,BCR(R1)     ;SET UP TO XFER ONE CHAR
015030  010561  000006                          MOV     R5,CAR(R1)      ;SET UP THE BUS ADDRESS REG
015034  162761  000002  000006                  SUB     #2,CAR(R1)      ;CORRECT BUS ADDRESS
015042  012767  000100  164150                  MOV     #100,$TMP7      ;COUNT 64 CHARS TO BE RECEIVED IN AUTO ECHO
015050  005067  164142                          CLR     $TMP6           ;INIT CHAR COUNTER
015054  012761  133503  000004                  MOV     #133503,LPR(R1) ;SET UP LINE PARAMETER REG
015062  016761  012150  000012                  MOV     LINMSK,BAR(R1)  ;ACTIVATE THE LINE

015070  012767  000002  013174                  MOV     #2,TIMEA        ;INIT TIMER A
015076  005067  013172                          CLR     TIMEB           ;INIT TIMERB
015102  105711                  3$:             TSTB    (R1)            ;CHAR AVAIL SET ??
015104  100427                                  BMI     4$              ;BR IF YES
015106  004767  011664                          JSR     PC,TIMEIT       ;CALL THE TIMER
015112  000773                                  BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                        ;AROUND THIS BRANCH IF TIME OUT OCCURS

015114  004767  012040                          JSR     PC,SAPS         ;SAVE THE ERROR PSW
015120  005061  000004                          CLR     LPR(R1)         ;TURN OFF AUTO ECHO MODE
015124  010102                                  MOV     R1,R2           ;MAKE REGADR = DEVADR
015126  011103                                  MOV     (R1),R3         ;GET THE WAS DATA
015130  042703  100000                          BIC     #BIT15,R3       ;CLEAR JUNK BIT
015134  012704  000200                          MOV     #200,R4         ;SET UP S/B DATA
015140  156704  013100                          BISB    LINE,R4
015144  004767  007236                          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
015150  004567  007452                          JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
015154  030244                                  LINE
015156  032216                                  EM24+35
015160  104024                                  ERROR   24              ;DATA AVAIL FAILED TO SET ON TIME
```

```
015162  000677                          BR      1$              ;GO TRY NEXT LINE

015164  005267  164026          4$:     INC     $TMP6           ;COUNT ONE CHAR RECVD
015170  016103  000002                  MOV     NRC(R1),R3      ;GET THE WAS DATA
015174  020304                          CMP     R3,R4           ;WAS CHAR AUTO ECHOED CORRECTLY ?
015176  001417                          BEQ     5$              ;BR IF YES

015200  004767  011754                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
015204  005061  000004                  CLR     LPR(R1)         ;DISABLE AUTO ECHO
015210  010102                          MOV     R1,R2           ;SET UP REGADR
015212  062702  000002                  ADD     #NRC,R2
015216  004767  007164                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
015222  004567  007400                  JSR     R5,SUNUM        ;PUT LINE NO. IN ERROR MSG
015226  030244                          LINE
015230  032216                          EM24+35
015232  104024                          ERROR   24              ;CHAR AUTO ECHOED INCORRECTLY
015234  000652                          BR      1$              ;GO TRY NEXT LINE

015236  005367  163756          5$:     DEC     $TMP7           ;COUNT ONE CHAR READ OUT OF 64
015242  003317                          BGT     3$              ;BR IF NOT LAST ONE
015244  100646                          BMI     1$              ;;BR IF LAST ONE READ
015246  042761  100000  000004          BIC     #BIT15,LPR(R1)  ;DISABLE AUTO ECHO
015254  000712                          BR      3$              ;GO READ LAST CHAR

015256  005167  163726          6$:     COM     $TMP3           ;TOGGLE I/O FLAG
015262  001406                          BEQ     TST50           ;;BR IF DONE BOTH I/O DATA
015264  005067  012754                  CLR     LINE            ;INIT LINE NO TO 00
015270  012705  027402                  MOV     #AETABO,R5      ;SET POINTER TO 0'S TABLE
015274  000167  177452                  JMP     7$              ;REPEAT TEST FOR ZERO PATTERNS
```

```
                                ;;*****************************************************************
                                ;*TEST 50        AUTO ECHO TEST 2 - ALL SELECTED LINES
                                ;;*****************************************************************
015300  000004                  TST50:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST IS SIMILAR TO TEST 47 EXCEPT ALL SELECTED LINES OTHER
                                THAN THE A.E. TEST LINE ARE ACTIVELY TURNING AROUND A BINARY COUNT
                                TEST PATTERN IN NON-AUTO ECHO MODE AND THE A.E. TEST LINE IS TESTED
                                FOR ALL 1'S DATA ONLY.

                                ERRORS:
                                ********

                                1. [ERROR  32] IS CALLED TO REPORT A.E. TEST TIMEOUTS
                                2. [ERROR  31] IS CALLED TO REPORT ALL DATA COMPARE ERRORS

                                SYNC:   M7277   SH4     LOAD BAR LB+HB L        CN2
                                *****

                                DEBUG:
                                ******

                                        REFER TO TEST 47

                                KEY LOGIC:
                                **********

                                        REFER TO TEST 47

                                %
015302  012767  015356  163600           MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
015310  012705  027342                  MOV     #AETAB,R5       ;SET POINTER TO A.E. TEST DATA TABLE
015314  005067  011720                  CLR     LMSK1           ;INIT BIT TEST MASK
015320  000261                          SEC                     ;GENERATER MARKER BIT IN "C"
015322  000401                          BR      12$             ;GO SHIFT MASK
015324  000241                  1$:     CLC                     ;INIT THE "C" BIT
015326  006167  011706          12$:    ROL     LMSK1           ;SHIFT TEST BIT
015332  001410                          BEQ     11$             ;BR IF TESTED ALL LINES
015334  012567  163660                  MOV     (R5)+,$TMP7     ;GET THE A.E. TEST DATA FOR THIS LINE
015340  036767  011674  011666          BIT     LMSK1,LINSEL    ;TEST THIS LINE ?
015346  001766                          BEQ     1$              ;BR IF NOT
015350  004767  007160                  JSR     PC,SELINE       ;GO SELECT A LINE
015354                          11$:
015354  000575                          BR      TST51           ;;BR IF DONE ALL SELECTED LINES

015356  016701  011642          2$:     MOV     DHADR,R1        ;RESET DEVADR IN CASE OF ERROR LOOP
015362  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
015366  004767  011426                  JSR     PC,SETALL       ;GO SET UP FOR BINARY COUNT XFER ON
                                                                ;ALL LINES OTHER THAN THE SELECTED ONE
015372  156711  012646                  BISB    LINE,(R1)       ;SELECT THE LINE FOR A.E. TEST
015376  010561  000006                  MOV     R5,CAR(R1)      ;SET BUS ADDR TO XMIT TEST CHAR
015402  162761  000002  000006          SUB     #2,CAR(R1)      ;CORRECT THE ADDRESS
015410  012761  177777  000010          MOV     #-1,BCR(R1)     ;XMIT ONE CHAR ON THIS LINE
```

```
015416  012761  133503  000004          MOV     #133503,LPR(R1) ;DO IT AT 9600 BAUD/8 BITS
015424  116767  163452  163554          MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
015432  042767  177400  163546          BIC     #177400,$TMP2
015440  046767  011572  012034          BIC     LINMSK,LINACT   ;MAKE THIS LINE APPEAR INACTIVE
015446  016761  011562  000012          MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL SELECTED TRANSMITTERS

015454  012767  000002  012610  21$:    MOV     #2,TIMEA        ;INIT TIMER A
015462  005067  012606                  CLR     TIMEB           ;INIT TIMER B
015466  016103  000002          3$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
015472  100414                          BMI     4$              ;BR IF YES
015474  004767  011276                  JSR     PC,TIMEIT       ;CALL THE TIMER
015500  000772                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS

015502  016167  000004  163472          MOV     LPR(R1),$TMP0   ;SAVE THE CURRENT "LPR"
015510  004567  007112                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
015514  030244                          LINE
015516  032707                          EM32+35
015520  104032                          ERROR   32              ;AUTO ECHO TIMEOUT
015522  000700                          BR      1$              ;GO TRY NEXT LINE

015524  010304                  4$:     MOV     R3,R4           ;EXTRACT LINE NUMBER OF RCVD CHAR
015526  000304                          SWAB    R4
015530  042704  177760                  BIC     #177760,R4
015534  010402                          MOV     R4,R2           ;SAVE IT IN R2
015536  006302                          ASL     R2              ;GENERATE TABLE INDEX IN R2
015540  126704  012500                  CMPB    LINE,R4         ;IS THIS THE A.E. TEST LINE ??
015544  001432                          BEQ     5$              ;BRANCH IF YES.
015546  036267  030170  011460          BIT     $LNSEL(R2),LINSEL
015554  001737                          BEQ     21$

015556  026203  036244                  CMP     RBUF(R2),R3     ;RECVD DATA CORRECT ??
015562  001447                          BEQ     6$              ;BR IF IT WAS

015564  004767  011370                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
015570  010467  163420                  MOV     R4,$TMP5        ;SAVE THE LINE NUMBER
015574  016204  036244                  MOV     RBUF(R2),R4     ;SET UP S/B DATA
015600  062702  036244                  ADD     #RBUF,R2        ;SET UP S/B ADDRESS
015604  012701  177703                  MOV     #177703,R1      ;SET UP THE WAS ADDRESS
015610  004767  006572                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
015614  004567  007006                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
015620  001214                          $TMP5
015622  032552                          EM31+45
015624  104031                          ERROR   31              ;NON-ECHO DATA COMPARE ERROR
015626  000167  177472                  JMP     1$              ;GO TRY NEXT LINE

015632  020367  163362          5$:     CMP     R3,$TMP7        ;CHAR ECHOED OK ??
015636  001427                          BEQ     7$              ;BR IF YES

015640  004767  011314                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
015644  012702  001220                  MOV     #$TMP7,R2       ;SAVE THE S/B ADDRESS
015650  016704  163344                  MOV     $TMP7,R4        ;SAVE THE S/B DATA
015654  012701  177703                  MOV     #177703,R1      ;SAVE THE WAS ADDRESS
015660  004767  006522                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
015664  004567  006736                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MESSAGE
```

```
015670  030244                          LINE
015672  032552                          EM31+45
015674  104031                          ERROR   31              ;AUTO ECHO LINE DATA ERROR
015676  000167  177422                  JMP     1$              ;GO TRY NEXT LINE

015702  105262  036244          6$:     INCB    RBUF(R2)        ;GENERATE NEXT EXPECTED DATA ON THIS LINE
015706  001262                          BNE     21$             ;BR IF ITS NOT BACK TO 000
015710  046267  027442  011564          BIC     LINBIT(R2),LINACT       ;INDICATE THIOS LINE DONE 256 BYTES
015716  005767  011560          7$:     TST     LINACT          ;ALL LINES INACTIVE
015722  001254                          BNE     21$             ;BR IF NOT
015724  042761  100000  000004          BIC     #BIT15,LPR(R1)  ;TURN OFF THE A.E. BIT
015732  105761  000017                  TSTB    SSR+1(R1)       ;SILO EMPTY ??
015736  001002                          BNE     8$              ;BR IF NOT
015740  000167  177360                  JMP     1$              ;GO TEST NEXT LINE
015744  000167  177504          8$:     JMP     21$             ;GO EMPTY IT
```

```
                                ;;*****************************************************************
                                ;*TEST 51        AUTO ECHO TEST 3 - ALL SELECTED LINES
                                ;;*****************************************************************
015750  000004                  TST51:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST IS IDENTICAL TO TEST 47 EXCEPT ALL SELECTED LINES
                                ARE ACTIVATED CONCURRENTLY RATHER THAN ONE AT A TIME AND ONLY
                                THE ALL 1'S DATA IS USED.

                                ERRORS:
                                *******

                                1. [ERROR  36] IS CALLED TO REPORT "DATA AVAIL" TIMEOUTS
                                2. [ERROR  31] IS CALLED TO REPORT A.E. DATA ERRORS

                                SYNC:   M7277   SH4 LOAD BAR LB+HB L    CN2
                                *****

                                DEBUG:
                                ******

                                        REFER TO TEST 47

                                KEY LOGIC:
                                ***********

                                        REFER TO TEST 47

                                %
015752  012767  015760  163130          MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
015760  012711  004000          1$:     MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
015764  012705  000020                  MOV     #20,R5          ;INIT COUNTER TO SET UP 16. LINES
015770  012702  027342                  MOV     #AETAB,R2       ;SET UP POINTER TO AUTO ECXHO TEST DATA
015774  012703  036204                  MOV     #RCNT,R3        ;R3 POINTS TO TABLE OF CHAR COUNTERS
016000  010261  000006          2$:     MOV     R2,CAR(R1)      ;SET UP BUS ADDRESS REG
016004  012761  177777  000010          MOV     #-1,BCR(R1)     ;SET UP BYTE COUNT REG
016012  012761  131403  000004          MOV     #131403,LPR(R1) ;SET UP LINE PARAMETERS
016020  005023                          CLR     (R3)+           ;CLEAR A COUNTER
016022  062702  000002                  ADD     #2,R2           ;UPDATE POINTERS
016026  005211                          INC     (R1)            ;SELECT NEXT LINE
016030  005305                          DEC     R5              ;COUNT ONE DONE
016032  001362                          BNE     2$              ;BR TILL 16. DONE
016034  116767  163042  163144          MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
016042  042767  177400  163136          BIC     #177400,$TMP2
016050  016767  011160  011424          MOV     LINSEL,LINACT   ;SET FLAG TO INDICATE ALL 16. ACTIVE
016056  016761  011152  000012          MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL XMITTERS

016064  012767  000002  012200          MOV     #2,TIMEA        ;INIT TIMER A
016072  005067  012176                  CLR     TIMEB           ;INIT TIMERB
016076  005067  163110          3$:     CLR     $TMP4
016102  105711                          TSTB    (R1)            ;CHAR AVAIL SET YET ?
016104  100410                          BMI     4$              ;BR IF YES
016106  004767  010664                  JSR     PC,TIMEIT       ;CALL THE TIMER
```

```
016112  000771                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS
016114  016167  000004  163060          MOV     LPR(R1),$TMP0   ;SAVE THE "LPR" REG
016122  104036                          ERROR   36              ;DATA AVAILABLE TIMEOUT
016124  000467                          BR      TST52           ;;EXIT TEST ON ERROR

016126  016103  000002          4$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
016132  010302                          MOV     R3,R2           ;BUILD AND SAVE LINE NO.
016134  000302                          SWAB    R2
016136  042702  177760                  BIC     #177760,R2
016142  010267  163050                  MOV     R2,$TMP6        ;SAVE THE LINE NO.
016146  006302                          ASL     R2              ;GENERATE TABLE OFFSET
016150  005267  163036                  INC     $TMP4
016154  022767  000101  163030          CMP     #101,$TMP4
016162  001745                          BEQ     3$
016164  036267  030170  011042          BIT     $LNSEL(R2),LINSEL  ;IS THIS ONE OF THE SELECTED LINES?
016172  001002                          BNE     41$             ;IF SO,GO ANALYZE THE CHARACTER
016174  104000                          ERROR                   ;INDICATE SOME KIND OF  ERROR
016176  000753                          BR      4$              ;GO GET THE NEXT CHARACTER
016200  005262  036204          41$:    INC     RCNT(R2)        ;COUNT THE CHARACTER
016204  020362  027342                  CMP     R3,AETAB(R2)    ;IS THE DATA CORRECT ??
016210  001420                          BEQ     5$              ;BR IF YES

016212  004767  010742                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
016216  016204  027342                  MOV     AETAB(R2),R4    ;GET THE S/B DATA
016222  062702  027342                  ADD     #AETAB,R2       ;GENERATE S/B ADDRESS
016226  062701  000002                  ADD     #NRC,R1         ;GENERATE THE WAS ADDRESS
016232  004767  006150                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
016236  004567  006364                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
016242  001216                          $TMP6
016244  032552                          EM31+45
016246  104031                          ERROR   31              ;DATA COMPARE ERROR
016250  000415                          BR      TST52           ;;EXIT TEST ON ERROR

016252  022762  000100  036204  5$:     CMP     #100,RCNT(R2)   ;DONE 64. CHARS ON THIS LINE ?
016260  001306                          BNE     3$              ;BR IF NOT
016262  016711  162730                  MOV     $TMP6,(R1)      ;SELECT LINE IN SCR REG
016266  042761  100000  000004          BIC     #BIT15,LPR(R1)  ;TURN OFF A.E. BIT
016274  046267  027442  011200          BIC     LINBIT(R2),LINACT       ;ALL LINES INACTIVE ??
016302  001275                          BNE     3$              ;BR IF NOT
```

```
                                ;;*****************************************************************
                                ;*TEST 52        BREAK BIT TEST - ALL SELECTED LINES
                                ;;*****************************************************************
016304  000004                  TST52:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT THE "BREAK" FEATURE WORKS PROPERLY
                                FOR ALL SELECTED LINES. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SET UP THE ERROR LOOP RETURN
                                        2. RETRIEVE THE CORRECT S/B DATA FROM THE "BREAK" DATA
                                           TABLE AND UPDATE THE POINTER.
                                        3. GO SELECT A LINE TO TEST - GO TO THE NEXT TEST IF DONE ALL
                                           SELECTED LINES
                                        4. RESET THE DH11 AND CLEAR THE "CAR" AND "BCR" MEMORIES.
                                        5. PRIME SELECTED LINE TO OUTPUT TWO "NULL" CHARS TO CLEAR UART
                                        6. ACTIVATE THE SELECTED LINE
                                        7. WAIT FOR SILO TO RECEIVE TWO NULLS - IF TIMEOUT REPORT ERROR
                                           AND RESTART AT STEP 2
                                        8. IF NO TIMEOUT CLEAR THE SELECTED DH11 AND RESELECT LINE NO.
                                        9. PRIME SELECTED LINE TO OUT PUT 256. CHARS.
                                        10.SET THE SELECTED LINE'S BREAK BIT
                                        11.ACTIVATE THE SELECTED LINE
                                        12.WAIT FOR "BAR" REG TO CLEAR -F TIMEOUT REPORT ERROR AND
                                           RESTART AT STEP 2
                                        13.IF NO TIMEOUT VERIFY THAT THE SILO RECEIVED ONLY ONE CHAR-
                                           IF NOT REPORT ERROR AND RESTART AT STEP 2
                                        14.IF SILO RECEIVED ONLY ONE CHAR VERIFY THAT IT WAS A "BREAK"
                                           CHAR - IF NOT REPORT ERROR - AND RESTART AT STEP 2

                                ERRORS:
                                ********

                                1. [ERROR  25] IS CALLED TO REPORT ALL ERRORS

                                SYNC:   M7277   SH4     LOAD BCR H      FU1
                                *****

                                DEBUG:
                                *******

                                1. IF ALL LINES FAILED SUSPECT THAT THE M7277 IS NOT GENERATING THE
                                   BREAK CONTROL REG LOAD SIGNAL.

                                2. IF ONLY ONE LINE FAILS SUSPECT THE BREAK CONTROL LOGIC ON THE M7278

                                KEY SIGNALS:
                                ************

                                        M7277   SH4     LOAD BCR H      FU1

                                        M7278   SH5 THRU SH8

                                                        74175 REGISTER CHIPS E51, E38, E67, E60
```

```
                                                                    7400 DRIVERS    E45, E46, E75, E76

                                        %
016306  012767  016366  162574           MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
016314  012705  027544                  MOV     #BRKTAB,R5      ;SET UP POINTER TO BREAK DATA TABLE
016320  005067  010714                  CLR     LMSK1           ;INIT BIT TEST MASK
016324  000261                          SEC                     ;SET BIT MARKER IN "C"
016326  000401                          BR      12$             ;GO SHIFT MASK
016330  000241                  1$:     CLC                     ;INIT THE "C" BIT
016332  006167  010702          12$:    ROL     LMSK1           ;SHIFT TEST MARKER
016336  001411                          BEQ     11$             ;BR IF ALL LINES DONE
016340  012504                          MOV     (R5)+,R4        ;GET TEST DATA FOR THIS LINE
016342  036767  010672  010664          BIT     LMSK1,LINSEL    ;LINE SELECTED ?
016350  001767                          BEQ     1$              ;BR IF NOT
016352  004767  006156                  JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
016356  000401                          BR      11$             ;BR IF DONE ALL SELECTED LINES
016360  000402                          BR      2$              ;GO TEST THE SELECTED LINE
016362  000167  000454          11$:    JMP     9$              ;GO EXIT TEST
016366  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
016372  004767  006310                  JSR     PC,CLCABC       ;GO CLR THE "CAR" AND "BCR" MEMORIES
016376  116711  011642                  MOVB    LINE,(R1)       ;SELECT THE LINE

016402  012761  030300  000006          MOV     #TNULL,CAR(R1)  ;SET UP TO OUTPUT TWO NULL CHARS
016410  012761  177776  000010          MOV     #-2,BCR(R1)     ;SET BYTE COUNT TO 2
016416  012761  033503  000004          MOV     #33503,LPR(R1)  ;SET UP LINE PARAMETERS
016424  016761  010606  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE SELECTED LINE

016432  012767  000001  011632          MOV     #1,TIMEA        ;INIT TIMER A
016440  005067  011630                  CLR     TIMEB           ;INIT TIMER B
016444  122761  000002  000017  3$:     CMPB    #2,SSR+1(R1)    ;TWO CHARS RECEIVED ??
016452  001432                          BEQ     4$              ;BR IF YES
016454  004767  010316                  JSR     PC,TIMEIT       ;CALL THE TIMER
016460  000771                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS

016462  004767  010472                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
016466  010467  162512                  MOV     R4,$TMP1        ;SAVE S/B DATA
016472  010102                          MOV     R1,R2           ;SET UP REGADR
016474  062702  000016                  ADD     #SSR,R2
016500  011203                          MOV     (R2),R3         ;GET THE WAS DATA
016502  042703  100377                  BIC     #100377,R3      ;CLEAR JUNK
016506  012704  000002                  MOV     #2,R4           ;SET UP S/B DATA
016512  000304                          SWAB    R4
016514  004767  005666                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
016520  004567  006102                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MESSAGE
016524  030244                          LINE
016526  032255                          EM25+34
016530  016704  162450                  MOV     $TMP1,R4        ;RESTORE S/B DATA
016534  104025                          ERROR   25              ;TIMED OUT WAITING FOR TWO NULLS
016536  000674                          BR      1$              ;GO TRY NEXT LINE

016540  012711  004000          4$:     MOV     #BIT11,(R1)     ;CLEAR THE INTERFACE
016544  116711  011474                  MOVB    LINE,(R1)       ;SELECT THE LINE
016550  012761  037244  000006          MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDRESS REG FOR XMITTR
016556  012761  177400  000010          MOV     #-400,BCR(R1)   ;SET BYTE COUNT TO XMIT 256(10) CHARS
016564  012761  033503  000004          MOV     #33503,LPR(R1)  ;SET UP LINE PARAMETERS
```

```
016572  016761  010440  000014          MOV     LINMSK,BKR(R1)  ;SET BREAK BIT FOR ACTIVE LINE
016600  016761  010432  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE

016606  012767  000005  011456          MOV     #5,TIMEA        ;INIT TIMER A
016614  005067  011454                  CLR     TIMEB           ;INIT TIMER B
016620  005761  000012          5$:     TST     BAR(R1)         ;BAR BIT CLEARED ??
016624  001426                          BEQ     6$              ;BR IFD YES
016626  004767  010144                  JSR     PC,TIMEIT       ;CALL THE TIMER
016632  000772                          BR      5$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS

016634  004767  010320                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
016640  010467  162340                  MOV     R4,$TMP1        ;SAVE THE S/B DATA
016644  010102                          MOV     R1,R2           ;SET UP REGADR
016646  062702  000012                  ADD     #BAR,R2
016652  011203                          MOV     (R2),R3         ;GET THE WAS DATA
016654  005004                          CLR     R4              ;SET UP S/B DATA
016656  004767  005524                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
016662  004567  005740                  JSR     R5,SUNUM        ;PUT LINE NO IN MESSAGE
016666  030244                          LINE
016670  032255                          EM25+34
016672  016704  162306                  MOV     $TMP1,R4        ;RESTORE THE S/B DATA
016676  104025                          ERROR   25              ;BAR BIT FAILED TO CLEAR
016700  000613                          BR      1$              ;GO TRY NEXT LINE

016702  122761  000001  000017  6$:     CMPB    #1,SSR+1(R1)    ;ONE CHAR RECEIVED ?
016710  001430                          BEQ     7$              ;BR IF YES

016712  004767  010242                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
016716  010467  162262                  MOV     R4,$TMP1        ;SAVE THE S/B DATA
016722  010102                          MOV     R1,R2           ;SET UP REGADR
016724  062702  000016                  ADD     #SSR,R2
016730  011203                          MOV     (R2),R3         ;GET THE WAS DATA
016732  042703  100377                  BIC     #100377,R3      ;CLEAR JUNK
016736  012704  000001                  MOV     #1,R4           ;SET UP S/B DATA
016742  000304                          SWAB    R4
016744  004767  005436                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
016750  004567  005652                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MESSAGE
016754  030244                          LINE
016756  032255                          EM25+34
016760  016704  162220                  MOV     $TMP1,R4        ;RESTORE THE S/B DATA
016764  104025                          ERROR   25              ;FAILED TO RECEIVE THE ONE CHAR
016766  000167  177336                  JMP     1$              ;GO TRY NEXT LINE


016772  016103  000002          7$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
016776  020304                          CMP     R3,R4           ;WAS IT A BREAK CHAR ?
017000  001002                          BNE     8$              ;BR IF NOT CORRECT
017002  000167  177322                  JMP     1$              ;GO TEST NEXT LINE

017006  004767  010146          8$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
017012  010102                          MOV     R1,R2           ;SET UP REGADR
017014  062702  000002                  ADD     #NRC,R2
017020  004767  005362                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
017024  004567  005576                  JSR     R5,SUNUM        ;PUT LINE NO IN MESSAGE
017030  030244                          LINE
```

```
017032  032255                          EM25+34
017034  104025                          ERROR   25              ;INCORRECT DATA RECEIVED
017036  000167  177266                  JMP     1$              ;GO TRY NEXT LINE
017042  000240                  9$:     NOP                     ;EXIT THIS TEST
```

```
                                ;;*****************************************************************
                                ;*TEST 53        HALF DUPLEX TEST - ALL SELECTED LINES
                                ;;*****************************************************************
017044  000004                  TST53:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                *************

                                        THIS TEST VERIFIES THAT THE RECEIVERS ON ALL SELECTED LINES ARE
                                "BLINDED" WHEN THE HALF-DUPLEX MODE IS ENABLED. THE TEST
                                SEQUENCE IS AS FOLLOWS:

                                        1. SET UP THE ERROR LOOP RETURN
                                        2. GO SELECT A LINE NO. TO TEST
                                        3. IF DONE ALL SELECTED LINES - GO TO THE NEXT TEST
                                        4. RESET THE DH11 AND CLEAR THE "CAR" AND "BCR" MEMORIES
                                        5. PRIME THE SELECTED DH11 TO XMIT 256. CHARS IN HALF-
                                           DUPLEX MODE.
                                        6. ACTIVATE THE SELECTED LINE AND WAIT FOR THE "BAR" REG TO CLEAR
                                        7. IF TIMEOUT - REPORT ERROR AND GO TO STEP 2
                                        8. IF NO TIMEOUT VERIFY THE "CHAR AVAIL" DID NOT SET (RECEIVER
                                           BLINDED) - IF ERROR REPORT IT AND GO TO STEP2 - IF NO ERROR
                                            GO TO STEP 2

                                ERRORS:
                                *******

                                1. [ERROR  26] IS CALLED TO REPORT ALL ERRORS

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                ******

                                1. SUSPECT EITHER THE M7289 OR THE M7288 MODULES

                                KEY LOGIC:
                                ***********

                                        M7289   SH5     HALF DUPLEX <15:00> H SIGNALS
                                                        END OF CHAR <15:00> SIGNALS

                                        M7288   SH5     HALF DUPLEX <03:00> H SIGNALS
                                                SH7      "       "  <07:00> H SIGNALS
                                                SH9      "       "  <11:08> H SIGNALS
                                                SH11     "       "  <15:12> H SIGNALS

                                %
017046  012767  017062  162034          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
017054  004767  005454          1$:     JSR     PC,SELINE
017060  000477                          BR      TST54           ;;BR IF ALL LINES TESTED
017062  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR THE INTERFACE
017066  004767  005614                  JSR     PC,CLCABC       ;GO CLR THE "CAR" AND "BCR" MEMORIES
017072  156711  011146                  BISB    LINE,(R1)       ;SELECT THE LINE
017076  012761  037244  000006          MOV     #TBUF,CAR(R1)   ;POINT TO XMIT BUFFER
```

```
017104  012761  177400  000010          MOV     #-400,BCR(R1)     ;XMIT 256(10) CHARS
017112  012761  073503  000004          MOV     #73503,LPR(R1)    ;SET UP THE LINE PARAMETERS
017120  016761  010112  000012          MOV     LINMSK,BAR(R1)    ;ACTIVATE THE SELECTED LINE

017126  012767  000001  011136          MOV     #1,TIMEA          ;INIT TIME A
017134  005067  011134                  CLR     TIMEB             ;INIT TIME B
017140  005761  000012          3$:     TST     BAR(R1)           ;WAIT FOR XMITTR TO FINISH
017144  001423                          BEQ     4$                ;BR IF XMITTR FINISHED
017146  004767  007624                  JSR     PC,TIMEIT         ;CALL TIMER
017152  000772                          BR      3$                ;TIMER WILL MOVE RETURN PC AROUND
                                                                  ;THIS BRANCH IF TIMEOUT OCCURS

017154  004767  010000                  JSR     PC,SAPS           ;SAVE THE ERROR PSW
017160  016103  000012                  MOV     BAR(R1),R3        ;GET THE WAS DATA
017164  010102                          MOV     R1,R2             ;SET UP REGADR
017166  062702  000012                  ADD     #BAR,R2
017172  005004                          CLR     R4                ;SET UP NEW S/B DATA
017174  004767  005206                  JSR     PC,SUER2A         ;GO SET UP THE ERROR INFO
017200  004567  005422                  JSR     R5,SUNUM          ;PUT LINE NO. IN MESSAGE
017204  030244                          LINE
017206  032320                          EM26+37
017210  104026                          ERROR   26                ;BAR BIT FAILED TO CLEAR ON TIME
017212  000720                          BR      1$                ;GO TRY NEXT LINE

017214  105711                  4$:     TSTB    (R1)              ;CHAR AVAIL SET ??
017216  100316                          BPL     1$                ;BR IF NOT IT SHOULDN'T BE

017220  004767  007734                  JSR     PC,SAPS           ;SAVE THE ERROR PSW
017224  010102                          MOV     R1,R2             ;SET UP REGADR
017226  011103                          MOV     (R1),R3           ;GET WAS DATA
017230  042703  100000                  BIC     #BIT15,R3         ;CLEAR JUNK BIT
017234  116704  011004                  MOVB    LINE,R4           ;SET UP S/B DATA
017240  004767  005142                  JSR     PC,SUER2A         ;GO SETUP ERROR INFO
017244  004567  005356                  JSR     R5,SUNUM          ;PUT LINE NO. IN MSG
017250  030244                          LINE
017252  032320                          EM26+37
017254  104026                          ERROR   26                ;HALF DUPLEX FAILED TO BLIND RECVR
017256  000676                          BR      1$                ;GO SELECT NEXT LINE
```

```
                                ;;*****************************************************************
                                ;;*TEST 54       VERIFY THAT OVERRUN CAN SET PROPERLY - ALL SELECTED LINES
                                ;;*****************************************************************
017260  000004                  TST54:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST VERIFIES THAT "OVERRUN" SETS PROPERLY FOR ALL LINES
                                THAT ARE SELECTED FOR TEST WHEN THE OVERRUN CONDITION IS FORCED BY THE
                                PROGRAM. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SET UP THE ERROR LOOP RETURN
                                        2. SELECT A LINE NO. TO TEST - IF DONE ALL LINES GO TO
                                           THE NEXT TEST.
                                        3. PRIME THE SELECTED LINE TO XMIT 68. CHARS
                                        4. ACTIVATE THE SELECTED LINE
                                        5. WAIT FOR "XMIT DONE" TO SET - IF TIMEOUT REPORT ERROR
                                           AND RESTART AT STEP 2
                                        6. IF NO TIMEOUT READ 65. CHARS FROM THE SILO AND VERIFY THAT
                                           "OVERRUN" IS SET ON THE LAST WORD READ
                                        7. IF NOT REPORT ERROR AND RESTART AT STEP 2

                                ERRORS:
                                ********

                                1. [ERROR  50] IS CALLED TO REPORT "XMIT DONE " TIMEOUTS
                                2. [ERROR  56] IS CALLEDT TO REPORT "OVERRUN" ERROR

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                ******

                                1. IF FAULT APPEARS ON ONLY ONE LINE SUSPECT UART MODULE
                                   FOR THE APPROPRIATE LINE IN QUESTION.

                                2. IF FAULT APPEARS ON ALL LINES SUSPECT THE M7279 MODULE

                                KEY LOGIC:
                                ***********

                                        M7279   SH1     MASTER OR H     E12-9
                                                SH2     MEMORY CHIP (3341) E13-11

                                        M7280   SH2     UC1 OR 2 MASTER OR      EN2
                                                SH2-5   UART PIN 15 (BUF OR LINE NN)

                                %

017262  012767  017276  161620          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
017270  004767  005240          1$:     JSR     PC,SELINE       ;GO SE ECT A LINE # TO TEST
017274  000512                          BR      TST55           ;;BR IF DONE ALL SELECTED LINES
017276  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
017302  116711  010736                  MOVB    LINE,(R1)       ;SELECT THE LINE TO TEST
```

F14


```
017306  012761  037244  000006          MOV     #TBUF,CAR(R1)    ;SET UP CURRENT ADDRESS
017314  012761  177674  000010          MOV     #-68.,BCR(R1)    ;SET UP BYTE COUNT REG
017322  012761  033503  000004          MOV     #33503,LPR(R1)   ;DO IT AT 9600 BAUD - 8 BITS
017330  016761  007702  000012          MOV     LINMSK,BAR(R1)   ;ACTIVATE THE SELECTED LINE

017336  012767  000001  010726          MOV     #1,TIMEA         ;INIT TIMERS A AND B
017344  005067  010724                  CLR     TIMEB
017350  005711                  3$:     TST     (R1)             ;TRANSMITTER DONE ??
017352  100425                          BMI     4$               ;BR IF YES
017354  004767  007416                  JSR     PC,TIMEIT        ;CALL TIMER
017360  000773                          BR      3$               ;BR IF NO TIMEOUT

017362  004767  007572                  JSR     PC,SAPS          ;GO SAVE PSW
017366  011103                          MOV     (R1),R3          ;GET THE WAS DATA
017370  042703  077760                  BIC     #77760,R3        ;CLEAR UNINTERESTING BITS
017374  116704  010644                  MOVB    LINE,R4          ;SET UP S/B DATA
017400  052704  100000                  BIS     #BIT15,R4
017404  010102                          MOV     R1,R2            ;SET UP REGADR
017406  004767  004774                  JSR     PC,SUER2A        ;GO SET UP ERROR INFO
017412  004567  005210                  JSR     R5,SUNUM         ;PUT LINE NO. IN MESSAGE HEADER
017416  030244                          LINE
017420  034176                          EM50+53
017422  104050                          ERROR   50               ;REPORT XMIT DONE TIME OUT
017424  000721                          BR      1$               ;GO TRY NEXT LINE

017426  012767  000101  161550  4$:     MOV     #65.,$TMP1       ;SET UP TO READ 65. WORDS FROM SILO
017434  116704  010604                  MOVB    LINE,R4          ;SET UP S/B DATA
017440  000304                          SWAB    R4
017442  152704  000101                  BISB    #65.,R4
017446  052704  140000                  BIS     #BIT15+BIT14,R4  ;PUT IN OVERRUN AND VALID DATA BITS
017452  016103  000002          5$:     MOV     NRC(R1),R3       ;GET WAS DATA FROM SILO
017456  005367  161522                  DEC     $TMP1            ;COUNT ONE WORD READ
017462  001373                          BNE     5$               ;BR TIL 65. READ
017464  020304                          CMP     R3,R4            ;WAS DATA AND OVERRUN CORRECT ??
017466  001700                          BEQ     1$               ;BR IF YES TRY NEXT SELECTED LINE

017470  004767  007464                  JSR     PC,SAPS          ;GO SAVE PSW
017474  010102                          MOV     R1,R2            ;SET UP REGADR
017476  062702  000002                  ADD     #NRC,R2
017502  004767  004700                  JSR     PC,SUER2A        ;GO SET UP ERROR INFO
017506  004567  005114                  JSR     R5,SUNUM         ;GO PUT LINE NO. IN MSG HDR
017512  030244                          LINE
017514  034614                          EM56+42
017516  104056                          ERROR   56               ;OVERRUN OR DATA INCORRECT
017520  000663                          BR      1$               ;GO TEST NEXT SELECTED LINE
```

```
                                        ;;*****************************************************************
                                        ;;*TEST 55        ABBREVIATED MODEM CONTROL DIAGNOSTIC
                                        ;;*****************************************************************
017522  000004                          TST55:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        ***************

                                                THE FOLLOWING 4 TESTS ARE EXTRACTED FROM DZDHK DIAGNOSTIC
                                        AND ARE INSERTED HERE SO THAT ALL LEVEL CONVERTERS AND CABLES CAN
                                        BE CHECKED WITH JUST ONE PROGRAM (RATHER THAN TWO) USING THE H315
                                        TURNAROUND CONNECTOR.

                                                THIS TEST VERIFIES THAT THE LINE ENABLE FUNCTION FLIP-FLOP
                                        CAN BE SET AND CLEARED FOR THE SELECTED LINE.

                                        ERRORS
                                        ******

                                                IF ANY ERRORS OCCUR IN THE FOLLOWING TESTS, THEN DZDHK,
                                        THE  MODEM CONTROL DIAGNOSTIC SHOULD BE RUN IN
                                        ITS ENTIRETY FOR A MORE COMPLETE MODEM CONTROL CHECKOUT.


                                        %

017524  032777  002000  161406                  BIT     #BIT10,@SWR             ;CHECK MODEM CONTROL?
017532  001402                                  BEQ     2$                      ;BRANCH IF YES.
017534  000167  001032                          JMP     ENDA                    ;OTHERWISE, GET OUT.
017540  012700  030066          2$:             MOV     #DMADRS,R0              ;R0 POINTS TO BEGINNING OF DM ADDRESS TABLE.
017544  016767  010472  161430                  MOV     DHNUM,$TMP0
017552  006367  161424                          ASL     $TMP0                   ;DOUBLE TMP0
017556  066700  161420                          ADD     $TMP0,R0                ;CREATE AN OFFSET.
017562  011067  010444                          MOV     (R0),DHMCSR             ;MOVE THE DM ADDRESS INTO DHMCSR.
017566  016767  010440  010440                  MOV     DHMCSR,DHMLSR
017574  062767  000002  010432                  ADD     #2,DHMLSR               ;SAVE LINE STATUS REGISTER ADDRESS.
017602  005767  010424                          TST     DHMCSR                  ;IS THERE A MODEM CONTROL HERE?
017606  001004                                  BNE     DOIT11                  ;BRANCH IF YES.
017610  104401                                  TYPE
017612  036151                                  MSG5                            ;NO MODEM CONTROL FOUND.
017614  000167  000752                          JMP     ENDA                    ;GET OUTTA HERE.
017620  004767  004710          DOIT11:         JSR     PC,SELINE               ;GO SELECT A LINE.
017624  000473                                  BR      TST56           ;;BR IF DONE ALL SELECTED LINES

017626  005077  010400          MUX11:          CLR     @DHMCSR                 ;CLEAR CONTROL STATUS REGISTER
017632  016767  010374  161342                  MOV     DHMCSR,$TMP0            ;SAVE DEVICE REGISTER FOR ERROR MESSAGE
017640  042767  000340  160130                  BIC     #340,PS                 ;ENABLE INTERRUPTS
017646  116701  010372                          MOVB    LINE,R1
017652  010167  161330                          MOV     R1,$TMP2                ;SAVE LINE NUMBER FOR ERROR MESSAGE
017656  012777  002000  010346  MUX11A:         MOV     #CLRMUX,@DHMCSR
017664  012702  000020                          MOV     #16.,R2
017670  010177  010336                          MOV     R1,@DHMCSR              ;SELECT LINE TO BE TESTED
017674  012777  000001  010332                  MOV     #LINENA,@DHMLSR         ;SET LINE ENABLE FUNCTION FLIP-FLOP
017702  005077  010324                          CLR     @DHMCSR                 ;THE STEP BIT WILL BE USED TO FIND RIGHT LINE
017706  005005                  MUX11B:         CLR     R5
017710  017704  010320                          MOV     @DHMLSR,R4              ;READ LINE STATUS REGISTER
```

```
017714  117703  010312                  MOVB    @DHMCSR,R3          ;READ CONTROL STATUS REGISTER
017720  042703  177760                  BIC     #177760,R3          ;CLEAR UNWANTED BITS
017724  020103                          CMP     R1,R3               ;IF LINE NUMBER=SELECTED LINE NUMBER,
017726  001002                          BNE     MUX11C              ;EXCEPT LINE ENABLE FUNCTION FLIP FLOP
017730  012705  000001                  MOV     #LINENA,R5
                                                                    ;TO BE SET
017734  020504                  MUX11C: CMP     R5,R4               ;COMPARE EXPECTED AND RECEIVED
017736  001403                          BEQ     MUX11D              ;RESULTS
017740  030504                          BIT     R5,R4               ;TEST TO SEE IF THE LINE ENABLE BIT WAS SET
017742  001001                          BNE     MUX11D              ;IF SO, EVERYTHING IS OK
017744  104060                          ERROR   60                  ;INDICATE A MODEM CONTROL ERROR
017746  052777  000400  010256  MUX11D: BIS     #STEP,@DHMCSR       ;EXAMINE NEXT LINE
017754  005302                          DEC     R2
017756  001353                          BNE     MUX11B
017760  005005                          CLR     R5
017762  010177  010244          MUX11E: MOV     R1,@DHMCSR
017766  010103                          MOV     R1,R3               ;SET LINE COUNTER TO SELECTED LINE
017770  005077  010240                  CLR     @DHMLSR             ;CLEAR LINE ENABLE FLIP FLOP
017774  105227  000000                  INCB    #0                  ;DELAY FOR CABLE
020000  001375                          BNE     .-4                 ;DITTO
020002  017704  010226                  MOV     @DHMLSR,R4          ;READ LINE STATUS REGISTER
                                                                    ;WAS LINE ENABLE FUNCTION FLIP FLOP
020006  001704                          BEQ     DOIT11              ;CLEARED
020010  104060                          ERROR   60                  ;MODEM CONRTROL ERROR
020012  000702                          BR      DOIT11              ;DO THE NEXT LINE THAT IS SELECTED
```

```
                                        ;;*****************************************************
                                        ;*TEST 56        MODEM CONTROL DIAGNOSTIC CONTINUED
                                        ;;*****************************************************
020014  000004                          TST56:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        ***************

                                                THIS TEST VERIFIES THAT CLEAR TO SEND AND CARRIER ARE SET
                                        IF  "LINE ENABLE" AND TERMINAL ARE SET FOR THE SELECTED LINE.

                                        ERRORS:
                                        ********

                                                IF ANY ERRORS OCCUR, RUN THE  MODEM CONTROL DIAGNOSTIC,
                                         DZDHK.

                                        %
020016  005077  010212                  DOIT15: CLR     @DHMLSR                         ;CLEAR LINE STATUS REGISTER
020022  004767  004506                          JSR     PC,SELINE                       ;GO SELECT A LINE.
020026  000471                                  BR      TST57           ;;BR IF DONE ALL SELECTED LINES
020030  005077  010176                  MUX15:  CLR     @DHMCSR                         ;CLEAR CONTROL REGISTER
020034  042767  000340  157734                  BIC     #340,PS                         ;ENABLE INTERRUPTS
020042  116701  010176                          MOVB    LINE,R1
020046  012777  002000  010156                  MOV     #CLRMUX,@DHMCSR                 ;RESET ALL THE LINE STATUS REGISTERS
020054  010167  161126                          MOV     R1,STMP2                        ;SAVE LINE NBUMBER FOR ERROR MESSAGE
020060  012702  000020                  MUX15A: MOV     #16.,R2                         ;16 LINES
020064  010177  010142                          MOV     R1,@DHMCSR                      ;SELECT A LINE
020070  012777  000003  010136                  MOV     #LINENA+TRMRDY,@DHMLSR          ;SET LINE ENABLE +TRMRDY
020076  005077  010130                          CLR     @DHMCSR                         ;CLEAR CONTROL REGISTER
020102  005005                          MUX15B: CLR     R5                              ;CLEAR EXPECTED RESULT
020104  017704  010124                          MOV     @DHMLSR,R4                      ;READ LINE STATUS
020110  117703  010116                          MOVB    @DHMCSR,R3                      ;READ LINE NUMBER
020114  042703  177760                          BIC     #177760,R3                      ;CLEAR UNWANTED BITS
020120  020103                                  CMP     R1,R3                           ;IF RECEIVED LINE=SELECTED LINE
020122  001002                                  BNE     MUX15C                          ;EXPECT LINE ENABLE AND
020124  012705  000143                          MOV     #LINENA+TRMRDY+CO+CS,R5
                                                                                        ;CLEAR TO SEND AND CARRIER ARE SET
020130  020405                          MUX15C: CMP     R4,R5                           ;COMPARE EXPECTED AND
020132  001401                                  BEQ     MUX15D                          ;RECEIVED RESULTS
020134  104060                                  ERROR   60                              ;MODEM CONTROL ERROR
020136  052777  000400  010066          MUX15D: BIS     #STEP,@DHMCSR                   ;UPDATE LINE COUNTER
020144  005302                                  DEC     R2                              ;CONTINUE IF ALL CHECKS
020146  001355                                  BNE     MUX15B                          ;ARE NOT DONE FOR THIS LINE
020150  012705  000001                          MOV     #LINENA,R5                      ;EXPECT LINE ENABLE
020154  010103                          MUX15E: MOV     R1,R3                           ;ON SELECTED LINE
020156  010177  010050                          MOV     R1,@DHMCSR                      ;SELECT LINE
020162  042777  000002  010044                  BIC     #TRMRDY,@DHMLSR                 ;CLEAR TERMINAL
020170  105227  000000                          INCB    #0                              ;DELAY FOR CABLE
020174  001375                                  BNE     .-4                             ;DITTO
020176  017704  010032                          MOV     @DHMLSR,R4                      ;READ LINE STATUS REGISTER
020202  020504                                  CMP     R5,R4                           ;ONLY LINE ENABLE SHOULD BE
020204  001704                                  BEQ     DOIT15                          ;SET ON THIS LINE
020206  104060                                  ERROR   60                              ;MODEM COMNTROL ERROR
020210  000702                                  BR      DOIT15                          ;GO DO THE NEXT LINE
```

```
                                        ;;*****************************************************************
                                        ;*TEST 57        MODEM CONTROL DIAGNOSTIC CONTINUED
                                        ;;*****************************************************************
020212  000004                          TST57:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        ****************

                                                THIS TEST VERIFIES THAT RING IS SET IF "LINE ENABLE" AND
                                        REQUEST TO SEND ARE SET FOR THE SELECTED LINE.

                                        ERRORS:
                                        *********

                                                IF ANY ERRORS OCCUR, RUN THE  MODEM CONTROL DIAGNOSTIC,
                                         DZDHK.

                                        %
020214  005077  010014                  DOIT16: CLR     @DHMLSR                 ;CLEAR LINE STATUS REGISTER
020220  004767  004310                          JSR     PC,SELINE               ;GO SELECT A LINE.
020224  000466                                  BR      TST60           ;;BR IF DONE ALL SELECTED LINES
020226  005077  010000                  MUX16:  CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
020232  042767  000340  157536                  BIC     #340,PS                 ;ENABLE INTERRUPTS
020240  116701  010000                          MOVB    LINE,R1
020244  010167  160736                          MOV     R1,$TMP2                ;SAVE LINE NBUMBER FOR ERROR MESSAGE
020250  012702  000020                  MUX16A: MOV     #16.,R2                 ;16 LINES
020254  010177  007752                          MOV     R1,@DHMCSR              ;SELECT A LINE
020260  012777  000005  007746                  MOV     #LINENA+RS,@DHMLSR      ;SET LINE ENABLE +RS
020266  005077  007740                          CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
020272  005005                          MUX16B: CLR     R5                      ;CLEAR EXPECTED RESULT
020274  017704  007734                          MOV     @DHMLSR,R4              ;READ LINE STATUS
020300  117703  007726                          MOVB    @DHMCSR,R3              ;READ LINE NUMBER
020304  042703  177760                          BIC     #177760,R3              ;CLEAR UNWANTED BITS
020310  020103                                  CMP     R1,R3                   ;IF RECEIVED LINE=SELECTED LINE
020312  001002                                  BNE     MUX16C                  ;EXPECT LINE ENABLE AND
020314  012705  000205                          MOV     #LINENA+RS+RING,R5
                                                                                ;RING IS SET
020320  020405                          MUX16C: CMP     R4,R5                   ;COMPARE EXPECTED AND
020322  001401                                  BEQ     MUX16D                  ;RECEIVED RESULTS
020324  104060                                  ERROR   60                      ;MODEM CONTROL ERROR
020326  052777  000400  007676          MUX16D: BIS     #STEP,@DHMCSR           ;UPDATE LINE COUNTER
020334  005302                                  DEC     R2                      ;CONTINUE IF ALL CHECKS
020336  001355                                  BNE     MUX16B                  ;ARE NOT DONE FOR THIS LINE
020340  012705  000001                          MOV     #LINENA,R5              ;EXPECT LINE ENABLE
020344  010103                          MUX16E: MOV     R1,R3                   ;ON SELECTED LINE
020346  010177  007660                          MOV     R1,@DHMCSR              ;SELECT LINE
020352  042777  000004  007654                  BIC     #RS,@DHMLSR             ;CLEAR REQUEST TO SEND
020360  105227  000000                          INCB    #0                      ;DELAY FOR CABLE
020364  001375                                  BNE     .-4                     ;DITTO
020366  017704  007642                          MOV     @DHMLSR,R4              ;READ LINE STATUS REGISTER
020372  020504                                  CMP     R5,R4                   ;ONLY LINE ENABLE SHOULD BE
020374  001707                                  BEQ     DOIT16                  ;SET ON THIS LINE
020376  104060                                  ERROR   60                      ;MODEM CONTROL ERROR
020400  000705                                  BR      DOIT16                  ;GO DO THE NEXT LINE
```

```
                                        ;;*****************************************************************
                                        ;*TEST 60        MODEM CONTROL DIAGNOSTIC CONTINUED
                                        ;;*****************************************************************
020402  000004                          TST60:  SCOPE
                                        .REM    %
                                        TEST ABSTRACT:
                                        ***************

                                                THIS TEST VERIFIES THAT SECONDARY RECEIVE IS SET IF "LINE
                                        ENABLE" AND SECONDARY TRANSMIT ARE SET FOR THE SELECTED LINE.

                                        ERRORS:
                                        *********

                                                IF ANY ERRORS OCCUR, RUN THE  MODEM CONTROL DIAGNOSTIC,
                                         DZDHK.

                                        %
020404  005077  007624                  DOIT17: CLR     @DHMLSR                 ;CLEAR LINE STATUS REGISTER.
020410  004767  004120                          JSR     PC,SELINE               ;GO SELECT A LINE.
020414  000466                                  BR      ENDA
020416  005077  007610                  MUX17:  CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
020422  042767  000340  157346                  BIC     #340,PS                 ;ENABLE INTERRUPTS
020430  116701  007610                          MOVB    LINE,R1
020434  010167  160546                          MOV     R1,$TMP2                ;SAVE LINE NBUMBER FOR ERROR MESSAGE
020440  012702  000020                  MUX17A: MOV     #16.,R2                 ;16 LINES
020444  010177  007562                          MOV     R1,@DHMCSR              ;SELECT A LINE
020450  012777  000011  007556                  MOV     #LINENA+SECTX,@DHMLSR   ;SET LINE ENABLE +SECTX
020456  005077  007550                          CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
020462  005005                          MUX17B: CLR     R5                      ;CLEAR EXPECTED RESULT
020464  017704  007544                          MOV     @DHMLSR,R4              ;READ LINE STATUS
020470  117703  007536                          MOVB    @DHMCSR,R3              ;READ LINE NUMBER
020474  042703  177760                          BIC     #177760,R3              ;CLEAR UNWANTED BITS
020500  020103                                  CMP     R1,R3                   ;IF RECEIVED LINE=SELECTED LINE
020502  001002                                  BNE     MUX17C                  ;EXPECT LINE ENABLE AND
020504  012705  000031                          MOV     #LINENA+SECTX+SECRX,R5
                                                                                ;SECONDARY RECEIVE IS SET
020510  020405                          MUX17C: CMP     R4,R5                   ;COMPARE EXPECTED AND
020512  001401                                  BEQ     MUX17D                  ;RECEIVED RESULTS
020514  104060                                  ERROR   60                      ;MODEM CONTROL ERROR
020516  052777  000400  007506          MUX17D: BIS     #STEP,@DHMCSR           ;UPDATE LINE COUNTER
020524  005302                                  DEC     R2                      ;CONTINUE IF ALL CHECKS
020526  001355                                  BNE     MUX17B                  ;ARE NOT DONE FOR THIS LINE
020530  012705  000001                          MOV     #LINENA,R5              ;EXPECT LINE ENABLE
020534  010103                          MUX17E: MOV     R1,R3                   ;ON SELECTED LINE
020536  010177  007470                          MOV     R1,@DHMCSR              ;SELECT LINE
020542  042777  000010  007464                  BIC     #SECTX,@DHMLSR          ;CLEAR SECONDARY TRANSMIT
020550  105227  000000                          INCB    #0                      ;DELAY FOR CABLE
020554  001375                                  BNE     .-4                     ;DITTO
020556  017704  007452                          MOV     @DHMLSR,R4              ;READ LINE STATUS REGISTER
020562  020504                                  CMP     R5,R4                   ;ONLY LINE ENABLE SHOULD BE
020564  001707                                  BEQ     DOIT17                  ;SET ON THIS LINE
020566  104060                                  ERROR   60                      ;MODEM CONTROL ERROR
020570  000705                                  BR      DOIT17                  ;GO DO THE NEXT LINE
```

```
020572  000004                  ENDA:   SCOPE
020574  012767  000240  000054          MOV     #240,$EOP       ;NOP THE SCOPE AT THE BEGINNING OF EOP
020602  005267  007434                  INC     DHNUM           ;GENERATE NEW DH11 NUMBER
020606  062767  000002  007434          ADD     #2,ADPTR        ;UPDATE THE TABLE POINTERS
020614  062767  000002  007430          ADD     #2,VCPTR
020622  062767  000002  007424          ADD     #2,BRPTR
020630  006367  006374                  ASL     SELMSK          ;SHIFT MARKER TO TEST NEXT DH11
020634  001410                          BEQ     $EOP            ;BR IF TESTED ALL SELECTED DH11'S
020636  036767  006366  006366          BIT     SELMSK,DHSEL    ;IS THIS DH11 SELECTED ?
020644  001752                          BEQ     ENDA            ;BR IF NOT
020646  105067  160230                  CLRB    $TSTNM          ;INIT TEST NUMBER
020652  000167  162062                  JMP     RSTRTA          ;GO TEST THIS DH11
                                .SBTTL  END OF PASS ROUTINE

                                ;;*****************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO START2

020656                          $EOP:
020656  000004                          SCOPE
020660  005067  160216                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
020664  005067  160332                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
020670  005267  160344                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
020674  042767  100000  160336          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
020702  005327                          DEC     (PC)+           ;;LOOP?
020704  000001                  $EOPCT: .WORD   1
020706  003022                          BGT     $DOAGN          ;;YES
020710  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
020712  000001                  $ENDCT: .WORD   1
020714  020704                          $EOPCT
020716  104401  020763                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
020722  016746  160312                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
020726  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
020730  104401  020760                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
020734  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
020740  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
020742  000005                          RESET                   ;;CLEAR THE WORLD
020744  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
020746  000240                          NOP                     ;;SAVE ROOM
020750  000240                          NOP                     ;;FOR
020752  000240                          NOP                     ;;ACT11
020754                          $DOAGN:
020754  000137                          JMP     @(PC)+          ;;RETURN
020756  002634                  $RTIWD: .WORD   START2
020760     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
020763     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
020770  050040  051501  020123
020776  000043
                                .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

021000                                  $SCOPE:
021000  104407                                  CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
021002  005067  007236                          CLR     LINE            ;INIT THE LINE NO. TO ZERO
021006  016701  006212                          MOV     DHADR,R1        ;SET UP DEVADR IN R1
021012  032777  040000  160120  1$:             BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
021020  001114                                  BNE     $OVER           ;;YES IF SW14=1
                                        ;*****START OF CODE FOR THE XOR TESTER*****
021022  000416                          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
021024  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
021030  012737  021050  000004                  MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
021036  005737  177060                          TST     @#177060        ;;TIME OUT ON XOR?
021042  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
021046  000463                                  BR      $SVLAD          ;;GO TO THE NEXT TEST
021050  022626                          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
021052  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
021056  000423                                  BR      7$              ;;LOOP ON THE PRESENT TEST
021060                                  6$:;*****END OF CODE FOR THE XOR TESTER*****
021060  032777  000400  160052                  BIT     #BIT08,@SWR     ;    P ON SPEC. TEST?
021066  001404                                  BEQ     2$              ;,   IF NO
021070  127767  160044  160004                  CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
021076  001465                                  BEQ     $OVER           ;;BR IF YES
021100  105767  157777          2$:             TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
021104  001421                                  BEQ     3$              ;;BR IF NO
021106  126767  160003  157767                  CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
021114  101015                                  BHI     3$              ;;BR IF NO
021116  032777  001000  160014                  BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
021124  001404                                  BEQ     4$              ;;BR IF NO
021126  016767  157756  157752  7$:             MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
021134  000446                                  BR      $OVER
021136  105067  157741          4$:             CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
021142  005067  160054                          CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
021146  000415                                  BR      1$              ;;ESCAPE TO THE NEXT TEST
021150  032777  004000  157762  3$:             BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
021156  001011                                  BNE     1$              ;;BR IF YES
021160  005767  160054                          TST     $PASS           ;;IF FIRST PASS OF PROGRAM
021164  001406                                  BEQ     1$              ;;      INHIBIT ITERATIONS
021166  005267  157712                          INC     $ICNT           ;;INCREMENT ITERATION COUNT
021172  026767  160024  157704                  CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
021200  002024                                  BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
021202  012767  000001  157674  1$:             MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
021210  016767  000052  160004                  MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
021216  105267  157660          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
021222  116767  157654  160006                  MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
021230  011667  157652                          MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
021234  011667  157650                          MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
021240  005067  157760                          CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
021244  112767  000001  157643                  MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
```

```
021252  016777  157624  157662  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
021260  016716  157622                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
021264  000002                          RTI                     ;;FIXES PS
021266  000010                  $MXCNT: 10                      ;;MAX. NUMBER OF ITERATIONS
                                .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 $ERRTYP ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1        HALT ON ERROR
                                ;*SW13=1        INHIBIT ERROR TYPEOUTS
                                ;*SW09=1        LOOP ON ERROR
                                ;*CALL
                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

021270                          $ERROR:
021270  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
021272  105267  157605          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
021276  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
021300  016777  157576  157634          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
021306  005267  157600                  INC     $ERTTL          ;;INC THE ERROR COUNT
021312  011667  157600                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
021316  162767  000002  157572          SUB     #2,$ERRPC
021324  117767  157566  157562          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
021332  032777  020000  157600          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
021340  001004                          BNE     20$             ;;SKIP TYPEOUTS
021342  004767  000106                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
021346  104401  001227                  TYPE    ,$CRLF
021352                          20$:
021352  122767  000001  157672          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
021360  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
021362  116767  157526  000004          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
021370  004767  001166                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
021374     000                  21$:    .BYTE   0
021375     000                          .BYTE   0
021376  000777                  22$:    BR      22$             ;;APT ERROR LOOP
021400  005777  157534          2$:     TST     @SWR            ;;HALT ON ERROR
021404  100002                          BPL     3$              ;;SKIP IF CONTINUE
021406  000000                          HALT                    ;;HALT ON ERROR!
021410  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
021412  032777  001000  157520  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
021420  001402                          BEQ     4$              ;;BR IF NO
021422  016716  157462                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
021426  005767  157572          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
021432  001402                          BEQ     5$              ;;BR IF NONE
021434  016716  157564                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
021440                          5$:
021440  022737  020744  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
021446  001001                          BNE     6$              ;;BRANCH IF NO
021450  000000                          HALT                    ;;YES
021452                          6$:
021452  000002                          RTI                     ;;RETURN
                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
```

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

021454                          $ERRTYP:
021454  104401  001227                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
021460  010046                          MOV     R0,-(SP)        ;;SAVE R0
021462  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
021464  153700  001114                  BISB    @#$ITEMB,R0
021470  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                ;;TYPE THE PC OF THE ERROR
021472  016746  157420                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                ;;ERROR ADDRESS
021476  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
021500  000426                          BR      6$              ;;GET OUT
021502  005300                  1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
021504  006300                          ASL     R0              ;;      WORK FOR THE ERROR TABLE
021506  006300                          ASL     R0
021510  006300                          ASL     R0
021512  062700  001356                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
021516  012067  000004                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
021522  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
021524  104401                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
021526  000000                  2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
021530  104401  001227                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
021534  012067  000004          3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
021540  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
021542  104401                          TYPE                    ;;TYPE THE "DATA HEADER"
021544  000000                  4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
021546  104401  001227                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
021552  011000                  5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
021554  001004                          BNE     7$              ;;GO TYPE THE DATA
021556  012600                  6$:     MOV     (SP)+,R0        ;;RESTORE R0
021560  104401  001227                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
021564  000207                          RTS     PC              ;;RETURN
021566                          7$:
021566  013046                          MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
021570  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
021572  005710                          TST     (R0)            ;;IS THERE ANOTHER NUMBER?
021574  001770                          BEQ     6$              ;;BR IF NO
021576  104401  021604                  TYPE    ,8$             ;;TYPE TWO(2) SPACES
021602  000771                          BR      7$              ;;LOOP
021604  020040     000          8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
        021610                          .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

021610  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
021614  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
021622  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
021626  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
021632  000406                          BR      $TYPON
021634  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
021642  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
021650  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
021656  010346                          MOV     R3,-(SP)        ;;SAVE R3
021660  010446                          MOV     R4,-(SP)        ;;SAVE R4
021662  010546                          MOV     R5,-(SP)        ;;SAVE R5
021664  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
021670  005404                          NEG     R4
021672  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
021676  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
021702  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
021706  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
021712  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
021714  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
021716  000404                          BR      3$              ;;GO DO MSB
021720  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
021722  006105                          ROL     R5
021724  006105                          ROL     R5
021726  010503                          MOV     R5,R3
021730  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
021732  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
021736  100016                          BPL     7$              ;;BR IF NO
021740  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
021744  001002                          BNE     4$              ;;TEST FOR 0
021746  005704                          TST     R4              ;;SUPPRESS THIS 0?
021750  001403                          BEQ     5$              ;;BR IF YES
021752  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
021754  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
021760  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
021764  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
021770  104401  022030                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
021774  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
022000  003347                          BGT     2$              ;;BR IF MORE TO DO
022002  002402                          BLT     6$              ;;BR IF DONE
022004  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
022006  000744                          BR      2$              ;;GO DO THE LAST DIGIT
022010  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
```

```
022012  012604                          MOV     (SP)+,R4        ;;RESTORE R4
022014  012603                          MOV     (SP)+,R3        ;;RESTORE R3
022016  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
022024  012616                          MOV     (SP)+,(SP)
022026  000002                          RTI                     ;;RETURN
022030     000                  B$      .BYTE   0               ;;STORAGE FOR ASCII DIGIT
022031     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
022032     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
022033     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
022034  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                ;;*****************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                ;*REPLACED WITH SPACES.
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                ;*      TYPDS                   ;;GO TO THE ROUTINE

022036                          $TYPDS:
022036  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
022040  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
022042  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
022044  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
022046  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
022050  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
022054  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
022060  100004                          BPL     1$              ;;BR IF INPUT IS POS.
022062  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
022064  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
022072  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
022074  012703  022252                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
022100  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
022104  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
022106  016001  022242                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
022112  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
022114  002402                          BLT     4$              ;;BR IF DONE
022116  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
022120  000774                          BR      3$
022122  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
022124  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
022126  001002                          BNE     5$              ;;FALL THROUGH IF 0
022130  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
022132  100407                          BMI     7$              ;;BR IF YES
022134  106316                  5$:     ASLB    (SP)            ;;MSD?
022136  103003                          BCC     6$              ;;BR IF NO
022140  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
022146  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
022152  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
022156  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
022160  005720                          TST     (R0)+           ;;JUST INCREMENTING
022162  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
022166  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
```

```
022170  003002                          BGT     8$              ;;GO TO EXIT
022172  010502                          MOV     R5,R2           ;;GET THE LSD
022174  000764                          BR      6$              ;;GO CHANGE TO ASCII
022176  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
022200  100003                          BPL     9$              ;;BR IF NO
022202  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
022210  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
022212  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
022214  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
022216  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
022220  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
022222  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
022224  104401  022252                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
022230  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
022236  012616                          MOV     (SP)+,(SP)
022240  000002                          RTI                     ;;RETURN TO USER
022242  023420                  $DTBL:  10000.
022244  001750                          1000.
022246  000144                          100.
022250  000012                          10.
022252  000004                  $DBLK:  .BLKW   4
                                .SBTTL  TYPE ROUTINE

                                ;;*****************************************************************
                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*

022262  105767  156671          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
022266  100002                          BPL     1$              ;;BR IF YES
022270  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
022272  000430                          BR      3$              ;;LEAVE
022274  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
022276  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
022302  122767  000001  156742          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
022310  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
022312  132767  000100  156733          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
022320  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
022322  010067  000004                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
022326  004767  000220                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
022332  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
022334  132767  000040  156711  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
022342  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
022344  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
022346  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
022350  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
```

```
022352  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
022354  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
022360  000002                          RTI                     ;;RETURN
022362  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
022366  001430                          BEQ     8$
022370  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
022374  001006                          BNE     5$
022376  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
022400  104401                          TYPE                    ;;TYPE A CR AND LF
022402  001227                          $CRLF
022404  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
022410  000755                          BR      2$              ;;GET NEXT CHARACTER
022412  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
022416  126726  156534          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
022422  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
022424  016746  156524                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
022430  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
022434  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
022436  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
022442  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
022446  000770                          BR      7$              ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

022450  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
022454  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
022460  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
022466  001372                          BNE     9$              ;;TAB STOP
022470  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
022472  000724                          BR      2$              ;;GET NEXT CHARACTER
022474  105777  156450          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
022500  100375                          BPL     $TYPEC
022502  116677  000002  156442          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
022510  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
022516  001003                          BNE     1$              ;;BRANCH IF NO
022520  105067  000014                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
022524  000406                          BR      $TYPEX          ;;EXIT
022526  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
022534  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
022536  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
022540  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
022542  000207                  $TYPEX: RTS     PC

                                .SBTTL  APT COMMUNICATIONS ROUTINE

                                ;;*****************************************************************
022544  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
022552  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
022560  000403                          BR      $ATYC
022562  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
022570                          $ATYC:
022570  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
022572  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
022574  105767  000206                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
022600  001450                          BEQ     5$              ;;IF NOT:  BR
```

```
022602  122767  000001  156442          CMPB    #APTENV,$ENV        ;;OPERATING UNDER APT?
022610  001031                          BNE     3$                  ;;IF NOT:  BR
022612  132767  000100  156433          BITB    #APTSPOOL,$ENVM     ;;SHOULD SPOOL MESSAGES?
022620  001425                          BEQ     3$                  ;;IF NOT:  BR
022622  017600  000004                  MOV     @4(SP),R0           ;;GET MESSAGE ADDR.
022626  062766  000002  000004          ADD     #2,4(SP)                    ;;BUMP RETURN ADDR.
022634  005767  156372          1$:     TST     $MSGTYPE            ;;SEE IF DONE W/ LAST XMISSION?
022640  001375                          BNE     1$                  ;;IF NOT:  WAIT
022642  010067  156400                  MOV     R0,$MSGAD           ;;PUT ADDR IN MAILBOX
022646  105720                  2$:     TSTB    (R0)+               ;;FIND END OF MESSAGE
022650  001376                          BNE     2$
022652  166700  156370                  SUB     $MSGAD,R0           ;;SUB START OF MESSAGE
022656  006200                          ASR     R0                  ;;GET MESSAGE LNGTH IN WORDS
022660  010067  156364                  MOV     R0,$MSGLGT          ;;PUT LENGTH IN MAILBOX
022664  012767  000004  156340          MOV     #4,$MSGTYPE         ;;TELL APT TO TAKE MSG.
022672  000413                          BR      5$
022674  017667  000004  000016  3$:     MOV     @4(SP),4$           ;;PUT MSG ADDR IN JSR LINKAGE
022702  062766  000002  000004          ADD     #2,4(SP)                    ;;BUMP RETURN ADDRESS
022710  016746  155062                  MOV     177776,-(SP)        ;;PUSH 177776 ON STACK
022714  004767  177342                  JSR     PC,$TYPE            ;;CALL TYPE MACRO
022720  000000                  4$:     .WORD   0
022722                          5$:
022722  105767  000062          10$:    TSTB    $FFLG               ;;SHOULD REPORT FATAL ERROR?
022726  001416                          BEQ     12$                 ;;IF NOT:  BR
022730  005767  156316                  TST     $ENV                ;;RUNNING UNDER APT?
022734  001413                          BEQ     12$                 ;;IF NOT:  BR
022736  005767  156270          11$:    TST     $MSGTYPE            ;;FINISHED LAST MESSAGE?
022742  001375                          BNE     11$                 ;;IF NOT:  WAIT
022744  017667  000004  156262          MOV     @4(SP),$FATAL       ;;GET ERROR #
022752  062766  000002  000004          ADD     #2,4(SP)                    ;;BUMP RETURN ADDR.
022760  005267  156246                  INC     $MSGTYPE            ;;TELL APT TO TAKE ERROR
022764  105067  000020          12$:    CLRB    $FFLG               ;;CLEAR FATAL FLAG
022770  105067  000013                  CLRB    $LFLG               ;;CLEAR LOG FLAG
022774  105067  000006                  CLRB    $MFLG               ;;CLEAR MESSAGE FLAG
023000  012601                          MOV     (SP)+,R1            ;;POP STACK INTO R1
023002  012600                          MOV     (SP)+,R0            ;;POP STACK INTO R0
023004  000207                          RTS     PC                  ;;RETURN
023006     000                  $MFLG:  .BYTE   0                   ;;MESSG. FLAG
023007     000                  $LFLG:  .BYTE   0                   ;;LOG FLAG
023010     000                  $FFLG:  .BYTE   0                   ;;FATAL FLAG
        023012                          .EVEN
        000200                  APTSIZE=200
        000001                  APTENV=001
        000100                  APTSPOOL=100
        000040                  APTCSUP=040
                                .SBTTL  TTY INPUT ROUTINE

                                ;;*****************************************************************
                                .ENABL  LSB

                                ;;*****************************************************************
                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                ;*WHEN OPERATING IN TTY FLAG MODE.
023012  022767  000176  156120  $CKSWR: CMP     #SWREG,SWR          ;;IS THE SOFT-SWR SELECTED?
```

```
023020  001074                          BNE     15$             ;;BRANCH IF NO
023022  105777  156116                  TSTB    @$TKS           ;;CHAR THERE?
023026  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
023030  117746  156112                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
023034  042716  177600                  BIC     #↑C177,(SP)     ;;STRIP-OFF THE ASCII
023040  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
023044  001062                          BNE     15$             ;;NO, RETURN TO USER
023046  126727  156062  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
023054  001456                          BEQ     15$             ;;BRANCH IF YES

023056  104401  023665                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (↑G)
023062  104401  023672          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
023066  016746  155104                  MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
023072  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
023074  104401  023703                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
023100  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
023102  005046                          CLR     -(SP)           ;;THE NEW SWR
023104  105777  156034          7$:     TSTB    @$TKS           ;;CHAR THERE?
023110  100375                          BPL     7$              ;;IF NOT TRY AGAIN

023112  117746  156030                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
023116  042716  177600                  BIC     #↑C177,(SP)     ;;MAKE IT 7-BIT ASCII


023122  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
023126  001005                          BNE     10$             ;;BRANCH IF NOT
023130  104401  023660                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (↑U)
023134  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
023140  000757                          BR      19$             ;;LET'S TRY IT AGAIN


023142  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
023146  001022                          BNE     16$             ;;BRANCH IF NO
023150  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
023154  001403                          BEQ     11$             ;;BRANCH IF YES
023156  016677  000002  155754          MOV     2(SP),@SWR      ;;SAVE NEW SWR
023164  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
023170  104401  001227          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
023174  126727  155735  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
023202  001003                          BNE     15$             ;;BRANCH IF NOT
023204  012777  000100  155732          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
023212  000002                  15$:    RTI                     ;;RETURN
023214  004767  177254          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
023220  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
023224  002420                          BLT     18$             ;;BRANCH IF YES
023226  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
023232  003015                          BGT     18$             ;;BRANCH IF YES
023234  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
023240  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
023244  001403                          BEQ     17$             ;;BRANCH IF YES
023246  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
023250  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
023252  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
023254  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
023260  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
```

```
023264  000707                          BR      7$              ;;GET THE NEXT ONE
023266  104401  001226          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
023272  000720                          BR      20$             ;;SIMULATE CONTROL-U
                                .DSABL  LSB


                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                ;
023274  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
023276  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
023304  105777  155634          1$:     TSTB    @$TKS           ;;WAIT FOR
023310  100375                          BPL     1$              ;;A CHARACTER
023312  117766  155630  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
023320  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
023326  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
023334  001013                          BNE     3$              ;;BRANCH IF NO
023336  105777  155602          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
023342  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
023344  117746  155576                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
023350  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
023354  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
023360  001366                          BNE     2$              ;;IF NOT DISCARD IT
023362  000750                          BR      1$              ;;YES, RESUME
023364  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
023372  002407                          BLT     4$              ;;BRANCH IF YES
023374  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
023402  003003                          BGT     4$              ;;BRANCH IF YES
023404  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
023412  000002                  4$:     RTI                     ;;GO BACK TO USER
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                ;*CALL:
                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

023414  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
023416  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
023420  012703  023650          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
023424  022703  023660          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
023430  101456                          BLOS    4$              ;;BR IF YES
023432  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
023434  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
023436  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
023442  001022                          BNE     5$              ;;BR IF NO
023444  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
023446  001007                          BNE     6$              ;;BR IF NO
023450  112767  000134  000170          MOVB    #'\,9$          ;;TYPE A BACK SLASH
023456  104401  023646                  TYPE    ,9$
023462  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
```

```
023466  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
023470  020327  023650                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
023474  103434                          BLO     4$              ;;BR IF YES
023476  111367  000144                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
023502  104401  023646                  TYPE    ,9$             ;;GO TYPE
023506  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
023510  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
023512  001406                          BEQ     7$              ;;BR IF NO
023514  112767  000134  000124          MOVB    #'\,9$          ;;TYPE A BACK SLASH
023522  104401  023646                  TYPE    ,9$
023526  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
023530  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
023534  001003                          BNE     8$              ;;BR IF NO
023536  104401  023660                  TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
023542  000726                          BR      1$              ;;GO START OVER
023544  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
023550  001011                          BNE     3$              ;;BRANCH IF NO
023552  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
023554  104401  001227                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
023560  104401  023650                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
023564  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
023566  104401  001226          4$:     TYPE    ,$QUES          ;;TYPE A '?'
023572  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
023574  111367  000046          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
023600  104401  023646                  TYPE    ,9$
023604  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
023610  001305                          BNE     2$              ;;LOOP IF NOT RETURN
023612  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
023616  104401  001230                  TYPE    ,$LF            ;;TYPE A LINE FEED
023622  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
023624  012603                          MOV     (SP)+,R3        ;;RESTORE R3
023626  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
023630  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
023636  012766  023650  000004          MOV     #$TTYIN,4(SP)
023644  000002                          RTI                     ;;RETURN
023646     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
023647     000                          .BYTE   0               ;;TERMINATOR
023650  000010                  $TTYIN: .BLKB   8               ;;RESERVE 8 BYTES FOR TTY INPUT
023660  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>     ;;CONTROL "U"
023665     136  006507  000012 $CNTLG: .ASCIZ  /↑G/<15><12>     ;;CONTROL "G"
023672  005015  053523  020122 $MSWR:  .ASCIZ  <15><12>/SWR = /
023700  020075     000
023703     040  047040  053505 $MNEW:  .ASCIZ  /  NEW = /
023710  036440  000040
                                .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                                ;*CHANGE IT TO BINARY.
                                ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
                                ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
                                ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
                                ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
                                ;*CALL:
                                ;*      RDOCT                   ;;READ AN OCTAL NUMBER
                                ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
```

```
                                        ;*                                  ;;HIGH ORDER BITS ARE IN $HIOCT
023714  011646                          $RDOCT: MOV     (SP),-(SP)          ;;PROVIDE SPACE FOR THE
023716  016666  000004  000002                  MOV     4(SP),2(SP)         ;;INPUT NUMBER
023724  010046                                  MOV     R0,-(SP)            ;;PUSH R0 ON STACK
023726  010146                                  MOV     R1,-(SP)            ;;PUSH R1 ON STACK
023730  010246                                  MOV     R2,-(SP)            ;;PUSH R2 ON STACK
023732  104411                          1$:     RDLIN                       ;;READ AN ASCIZ LINE
023734  012600                                  MOV     (SP)+,R0            ;;GET ADDRESS OF 1ST CHARACTER
023736  010067  000100                          MOV     R0,5$               ;;AND SAVE IT
023742  005001                                  CLR     R1                  ;;CLEAR DATA WORD
023744  005002                                  CLR     R2
023746  112046                          2$:     MOVB    (R0)+,-(SP)         ;;PICKUP THIS CHARACTER
023750  001420                                  BEQ     3$                  ;;IF ZERO GET OUT
023752  122716  000060                          CMPB    #'0,(SP)            ;;MAKE SURE THIS CHARACTER
023756  003026                                  BGT     4$                  ;;IS AN OCTAL DIGIT
023760  122716  000067                          CMPB    #'7,(SP)
023764  002423                                  BLT     4$
023766  006301                                  ASL     R1                  ;;*2
023770  006102                                  ROL     R2
023772  006301                                  ASL     R1                  ;;*4
023774  006102                                  ROL     R2
023776  006301                                  ASL     R1                  ;;*8
024000  006102                                  ROL     R2
024002  042716  177770                          BIC     #^C7,(SP)           ;;STRIP THE ASCII JUNK
024006  062601                                  ADD     (SP)+,R1            ;;ADD IN THIS DIGIT
024010  000756                                  BR      2$                  ;;LOOP
024012  005726                          3$:     TST     (SP)+               ;;CLEAN TERMINATOR FROM STACK
024014  010166  000012                          MOV     R1,12(SP)           ;;SAVE THE RESULT
024020  010267  000026                          MOV     R2,$HIOCT
024024  012602                                  MOV     (SP)+,R2            ;;POP STACK INTO R2
024026  012601                                  MOV     (SP)+,R1            ;;POP STACK INTO R1
024030  012600                                  MOV     (SP)+,R0            ;;POP STACK INTO R0
024032  000002                                  RTI                         ;;RETURN
024034  005726                          4$:     TST     (SP)+               ;;CLEAN PARTIAL FROM STACK
024036  105010                                  CLRB    (R0)                ;;SET A TERMINATOR
024040  104401                                  TYPE                        ;;TYPE UP THRU THE BAD CHAR.
024042  000000                          5$:     .WORD   0
024044  104401  001226                          TYPE    $QUES               ;;"?" "CR" & "LF"
024050  000730                                  BR      1$                  ;;TRY AGAIN
024052  000000                          $HIOCT: .WORD   0                   ;;HIGH ORDER BITS GO HERE
                                        .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.

024054  010046                          $TRAP:  MOV     R0,-(SP)            ;;SAVE R0
024056  016600  000002                          MOV     2(SP),R0            ;;GET TRAP ADDRESS
024062  005740                                  TST     -(R0)               ;;BACKUP BY 2
024064  111000                                  MOVB    (R0),R0             ;;GET RIGHT BYTE OF TRAP
024066  006300                                  ASL     R0                  ;;POSITION FOR INDEXING
024070  016000  024110                          MOV     $TRPAD(R0),R0       ;;INDEX TO TABLE
024074  000200                                  RTS     R0                  ;;GO TO ROUTINE
```

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

024076  011646                          $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
024100  016666  000004  000002                  MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
024106  000002                                  RTI                     ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

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

                                        ;       ROUTINE
                                        ;       -------
024110  024076                          $TRPAD: .WORD   $TRAP2
024112  022262                                  $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
024114  021634                                  $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
024116  021610                                  $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
024120  021650                                  $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
024122  022036                                  $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

024124  023062                                  $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

024126  023012                                  $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
024130  023274                                  $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
024132  023414                                  $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
024134  023714                                  $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;;*****************************************************************
                                        ;POWER DOWN ROUTINE
024136  012737  024302  000024          $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
024144  012737  000340  000026                  MOV     #340,@#PWRVEC+2 ;;PRIO:7
024152  010046                                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
024154  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
024156  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
024160  010346                                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
024162  010446                                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
024164  010546                                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
024166  017746  154746                          MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
024172  010667  000110                          MOV     SP,$SAVR6       ;;SAVE SP
024176  012737  024210  000024                  MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
024204  000000                                  HALT
024206  000776                                  BR      .-2             ;;HANG UP

                                        ;;*****************************************************************
                                        ;POWER UP ROUTINE
024210  012737  024302  000024          $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
024216  016706  000064                          MOV     $SAVR6,SP       ;;GET SP
024222  005067  000060                          CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
024226  005267  000054                  1$:     INC     $SAVR6          ;;WAIT FOR THE INC
024232  001375                                  BNE     1$              ;;OF  WORD
024234  012677  154700                          MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
024240  012605                                  MOV     (SP)+,R5        ;;POP STACK INTO R5
024242  012604                                  MOV     (SP)+,R4        ;;POP STACK INTO R4
```

```
024244  012603                          MOV     (SP)+,R3          ;;POP STACK INTO R3
024246  012602                          MOV     (SP)+,R2          ;;POP STACK INTO R2
024250  012601                          MOV     (SP)+,R1          ;;POP STACK INTO R1
024252  012600                          MOV     (SP)+,R0          ;;POP STACK INTO R0
024254  012737  024136  000024          MOV     #$PWRDN,@#PWRVEC  ;;SET UP THE POWER DOWN VECTOR
024262  012737  000340  000026          MOV     #340,@#PWRVEC+2   ;;PRIO:7
024270  104401                          TYPE                      ;REPORT THE POWER FAILURE
024272  024310                  $PWRMG: .WORD   $POWER            ;;POWER FAIL MESSAGE POINTER
024274  012716                          MOV     (PC)+,(SP)        ;;RESTART AT RSTRTA
024276  002740                  $PWRAD: .WORD   RSTRTA            ;;RESTART ADDRESS
024300  000002                          RTI
024302  000000                  $ILLUP: HALT                      ;;THE POWER UP SEQUENCE WAS STARTED
024304  000776                          BR      .-2               ;; BEFORE THE POWER DOWN WAS COMPLETE
024306  000000                  $SAVR6: 0                         ;;PUT THE SP HERE
024310  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
024316  000122
                                        .EVEN
```

```
                                        ;;*****************************************************
                                        ;COMMON DH11 SERVICE ROUTINES
                                        ;;*****************************************************

                                        ;THIS ROUTINE IS CALLED DURING START UP TO LOAD THE XMITTER
                                        ;OUTPUT BUFFER WITH A BINARY COUNT TEST PATTERN

024320  012701  037244                  LDTBF1: MOV     #TBUF,R1        ;POINT TO START OF BUFFER
024324  005002                                  CLR     R2              ;INIT DATA BYTE GENERATOR
024326  110221                          1$:     MOVB    R2,(R1)+        ;LOAD ONE CHAR
024330  005202                                  INC     R2              ;GENERATE NEXT CHAR
024332  022702  000400                          CMP     #400,R2         ;LOADED 256(10) BYTES
024336  001373                                  BNE     1$              ;BR IF NOT
024340  000207                                  RTS     PC              ;RETURN TO START TESTING

                                        ;THIS ROUTINE SETS UP THE ERROR INFORMATION REQUIRED BY ANY TEST
                                        ;USING A "DH1" HEADER

024342  004767  002612                  SUER1:  JSR     PC,SAPS         ;SAVE THE ERROR PSW
024346  116700  154530                          MOVB    $TSTNM,R0       ;SAVE THE TEST NO.
024352  010067  154604                          MOV     R0,$REG0        ;SAVE THE TEST NO. FOR ERROR PRINT
024356  010167  154602                          MOV     R1,$REG1        ;SAVE THE DH11 ADDR
024362  010267  154600                          MOV     R2,$REG2        ;SAVE THE REG ADDRESS
024366  010667  154604                          MOV     R6,$REG6        ;SAVE THE SP
024372  062767  000002  154576                  ADD     #2,$REG6        ;CORRECT FOR CALLING JSR
024400  000207                                  RTS     PC              ;RETURN TO CALLING ROUTINE

                                        ;THIS ROUTINE IS CALLED BY THOSE TESTS USING A "DH2" HEADER TO
                                        ;SAVE THE ERROR INFORMATION IN "DT2"

024402  004767  002552                  SUER2:  JSR     PC,SAPS         ;SAVE THE ERROR PSW
024406  116700  154470                  SUER2A: MOVB    $TSTNM,R0       ;GET THE TEST NO.
024412  010067  154544                          MOV     R0,$REG0        ;SAVE THE REGISTERS-TEST#
024416  010167  154542                          MOV     R1,$REG1        ;SAVE THE DH ADDRESS
024422  010267  154540                          MOV     R2,$REG2        ;SAVE THE REGISTER ADDRESS
024426  010367  154536                          MOV     R3,$REG3        ;SAVE THE WAS DATA
024432  010467  154534                          MOV     R4,$REG4        ;SAVE THE S/B DATA
024436  010667  154534                          MOV     R6,$REG6        ;SAVE THE STACK POINTER
024442  062767  000002  154526                  ADD     #2,$REG6        ;CORRECT FOR CALLING JSR
024450  000207                                  RTS     PC              ;RETURN TO REPORT ERROR

                                        ;THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
                                        ;BUS ERROR AND RSVD INSTR ERROR ROUTINES

024452  010067  154504                  SUER3:  MOV     R0,$REG0        ;SAVE THE REGS
024456  010167  154502                          MOV     R1,$REG1
024462  010267  154500                          MOV     R2,$REG2
024466  000207                                  RTS     PC              ;RETURN TO REPORT ERROR

                                        ;THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
                                        ;CAR/BCR MEMORY PATTERNS TESTS

024470  005067  154506                  SUER4:  CLR     $TMP0           ;SAVE THE LINE NO. WRITTEN
024474  116767  003546  154500                  MOVB    LINEA,$TMP0
024502  116700  154374                          MOVB    $TSTNM,R0       ;SAVE THE TEST NUMBER
024506  010067  154450                          MOV     R0,$REG0        ;SAVE THE REGISTER INFORMATION
```

```
024512  010167  154446                  MOV     R1,$REG1
024516  010267  154444                  MOV     R2,$REG2
024522  010367  154442                  MOV     R3,$REG3
024526  010467  154440                  MOV     R4,$REG4
024532  000207                          RTS     PC                      ;RETURN TO PATTERNS TEST

                                ;THIS ROUTINE IS CALLED TO SELECT A NEW LINE NO. BASED ON THE
                                ;VALUE OF THE LINE SELECTION PARAMETER

                                        ;CALLING SEQUENCE

                                        ;JSR    PC,SELINE               ;CALL THE ROUTINE
                                        ;BR     1$                      ;EXIT BRANCH-ROUTINE MOVES THE RETURN
                                                                        ;PC AROUND THIS BR IF MORE LINES ARE
                                                                        ;YET TO BE TESTED

024534  105767  003505          SELINE: TSTB    LINE+1                  ;FIRST TIME THROUGH FOR ANY TEST ?
024540  001010                          BNE     1$                      ;BR IF NOT
024542  105167  003477                  COMB    LINE+1                  ;SET ENTRY FLAG
024546  012767  000001  002462          MOV     #1,LINMSK               ;INIT SELECT TEST MASK TO TEST LINE 00
024554  105067  003464                  CLRB    LINE                    ;START WITH LINE #00
024560  000405                          BR      2$                      ;GO TEST FOR LINE #00
024562  105267  003456          1$:     INCB    LINE                    ;GENERATE NEW LINE NO.
024566  006367  002444                  ASL     LINMSK                  ;SHIFT SELECT MASK TO TEST NXT LINE
024572  001407                          BEQ     3$                      ;RETURN TO EXIT BRANCH - ALL LINES DONE
024574  036767  002436  002432  2$:     BIT     LINMSK,LINSEL           ;IS THE LINE SELECTED FOR TEST ??
024602  001767                          BEQ     1$                      ;BR IF NOT
024604  062716  000002                  ADD     #2,(SP)                 ;MOVE RETURN PC AROUND EXIT BRANCH
024610  000402                          BR      4$                      ;RETURN TO TEST SELECTED LINE
024612  005067  003426          3$:     CLR     LINE                    ;INIT ENTRY FLAG AND LINE NO. TO 000
024616  142777  000017  002400  4$:     BICB    #17,@DHADR              ;INIT LINE SELECT BITS IN "SCR"
024624  000207                          RTS     PC                      ;RETURN TO CALLING TEST

                                ;THIS ROUTINE IS CALLED TO CONVERT EITHER THE "DH" NUMBER OR THE
                                ;"LINE" NUMBER TO TWO ASCII CHARACTERS AND MOVE THEM INTO A
                                ;PARTICULAR MESSAGE BUFFER FOR ERROR REPORTING

                                        ;CALLING SEQUENCE

                                        ;JSR    R5,SUNUM                ;CALL TO THIS ROUTINE
                                        ;ADDR1                          ;ADDRESS OF THE NUMBER TO BE CONVERTED
                                        ;ADDR2                          ;ADDRESS OF THE MSG BUFFER SLOT

024626                          SUNUM:
024626  010046                          MOV     R0,-(SP)                ;;PUSH R0 ON STACK
024630  010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
024632  010246                          MOV     R2,-(SP)                ;;PUSH R2 ON STACK
024634  012500                          MOV     (R5)+,R0                ;GET ADDRESS OF NUMBER
024636  012501                          MOV     (R5)+,R1                ;GET MSG BUFFER ADDR
024640  111000                          MOVB    (R0),R0                 ;GET NO. TO BE CONVERTED
024642  010002                          MOV     R0,R2                   ;SAVE IT IN R2
024644  006202                          ASR     R2                      ;SHIFT MSD TO LSD POSITION
024646  006202                          ASR     R2
024650  006202                          ASR     R2
024652  042702  177770                  BIC     #177770,R2              ;CLR JUNK BITS
024656  062702  000060                  ADD     #60,R2                  ;MAKE IT ASCII
```

```
024662  110221                          MOVB    R2,(R1)+        ;PUT IT IN MSG BUFFER
024664  042700  177770                  BIC     #177770,R0      ;CLR JUNK FROM LSD
024670  062700  000060                  ADD     #60,R0          ;MAKE IT ASCII
024674  110011                          MOVB    R0,(R1)         ;PUT LSD IN THE BUFFER
024676  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
024700  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
024702  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
024704  000205                          RTS     R5              ;RETURN TO CALLER

                                ;THIS ROUTINE IS CALLED TO CLEAR THE "CAR" AND "BCR" MEMORIES
                                ;IT ASSUMES THAT THE ADDRESS OF THE "SCR" IS IN R1

024706  005067  154306          CLCABC: CLR     $TMP7           ;INIT A COUNTER
024712  116711  154302          1$:     MOVB    $TMP7,(R1)      ;SELECT A LINE
024716  005061  000006                  CLR     CAR(R1)         ;CLEAR A CAR LOCATION
024722  005061  000010                  CLR     BCR(R1)         ;CLEAR A BCR LOCATION
024726  005267  154266                  INC     $TMP7           ;GENERATE NEW LINE NO.
024732  022767  000020  154260          CMP     #20,$TMP7       ;DONE ALL LINES ?
024740  001364                          BNE     1$              ;BR IF NOT
024742  142711  000017                  BICB    #17,(R1)        ;SET "SCR" TO SELECT LINE 00
024746  000207                          RTS     PC              ;RETURN TO CALLER

                                ;THIS ROUTINE IS CALLED TO LOAD THE "BCR" MEMORY WITH ALL ONES
                                ;IT ASSUMES THAT THE ADDRESS OF THE SCR IS IN R1

024750  005067  154244          LDBCR:  CLR     $TMP7           ;INIT A COUNTER
024754  116711  154240          1$:     MOVB    $TMP7,(R1)      ;SELECT A LINE
024760  012761  177777  000010          MOV     #-1,BCR(R1)     ;LOAD BCR LOC. WITH 177777
024766  005267  154226                  INC     $TMP7           ;GENERATE NEXT LINE NO.
024772  022767  000020  154220          CMP     #20,$TMP7       ;DONE ALL LINES ?
025000  001365                          BNE     1$              ;BR IF NOT
025002  142711  000017                  BICB    #17,(R1)        ;SET "SCR" TO SELECT LINE 00
025006  000207                          RTS     PC              ;RETURN TO CALLER

                                ;THIS ROUTINE CALLED TO SET UP FOR PARITY TESTS

025010  012767  000020  154174  SUPPAR: MOV     #20,$TMP4       ;SET UP FOR 16. LINES
025016  105011                          CLRB    (R1)            ;INIT SCR TO START AT LINE 00
025020  005002                          CLR     R2              ;INIT INDEX REGISTER FOR RBUF (EVEN)
025022  012703  000200                  MOV     #200,R3         ;SET UP CONSTANT
025026  012704  000001                  MOV     #1,R4           ;INIT INDEX REG FOR RBUF (ODD)
025032  012761  037244  000006  1$:     MOV     #TBUF,CAR(R1)   ;LOAD BUS ADDRESS REWG
025040  016761  154152  000010          MOV     $TMP6,BCR(R1)   ;LOAD BYTE COUNT REG
025046  016761  154142  000004          MOV     $TMP5,LPR(R1)   ;LOAD LINE PARAMETERS
025054  105062  036244                  CLRB    RBUF(R2)        ;INIT DATA BYTE IN RBUF TO START AT 000
025060  110364  036244                  MOVB    R3,RBUF(R4)     ;SET CONSTANT IN HIGH BYTE
025064  005211                          INC     (R1)            ;SELECT NEXT LINE
025066  005203                          INC     R3              ;GENERATE NEW CONSTANT
025070  062702  000002                  ADD     #2,R2           ;UPDATE POINTERS TO RBUF (EVEN/ODD)
025074  062704  000002                  ADD     #2,R4
025100  005367  154106                  DEC     $TMP4           ;COUNT ONE LINE SETUP
025104  001352                          BNE     1$              ;BR TILL ALL 16. SET UP
025106  012704  027504                  MOV     #MULPTB,R4      ;SET UP TABLE POINTER
025112  016724  154100          2$:     MOV     $TMP6,(R4)+     ;SET UP BYTE COUNT ENTRY
025116  022704  027544                  CMP     #MULPTB+40,R4   ;SET UP ALL COUNTS ?
025122  001373                          BNE     2$              ;BR IF NOT
```

```
025124  105011                          CLRB    (R1)            ;INIT SCR TO SELECT LINE 00
025126  000207                          RTS     PC              ;RETURN TO PARITY TEST

                                ;THIS ROUTINE AUTOSIZES THE SYSTEM TO DETERMINE THE ADDRESSES AND
                                ;VECTORS OF THE DH11'S AND MODEM CONTROL'S.

025130  010046                  AUTOSZ: MOV     R0,-(SP)
025132  005003                          CLR     R3
025134  012702  027764                  MOV     #DHADRS,R2      ;POINT TO BEGINNING OF TABLE
025140  005022                  25$:    CLR     (R2)+           ;CLEAR AUTOSIZER TABLES.
025142  005203                          INC     R3
025144  020327  000102                  CMP     R3,#102         ;HAVE WE CLEARED ALL ENTRIES?
025150  001373                          BNE     25$             ;BRANCH IF NOT.
025152  013746  000004                  MOV     @#4,-(SP)       ;SAVE TRAP VECTOR.
025156  012737  025264  000004          MOV     #4$,@#4         ;SETUP FOR NON-EXISTENT MEMORY TRAP.
025164  012703  030066                  MOV     #DMADRS,R3      ;SETUP DM ADDRESS TABLE POINTER.
025170  012702  027764                  MOV     #DHADRS,R2      ;SET UP DH ADDRESS TABLE POINTER.

025174  012701  160020                  MOV     #160020,R1      ;R1=FIRST ADDRESS TO BE TESTED.

025200  005711                  1$:     TST     (R1)            ;SEE IF ADDRESS IN R1 RESPONDS.
025202  005761  000016                  TST     16(R1)          ;CHECK TO SEE IF DEVICE IS MODULO 20.
025206  052711  004000                  BIS     #4000,(R1)      ;IF IT IS, CONTINUE
                                                                ;AND CHECK TO SEE
025212  052711  001000                  BIS     #1000,(R1)      ;IF THIS ADDRESS CONTAINS
025216  052711  002000                  BIS     #2000,(R1)      ;A DH-11.
025222  032711  003000                  BIT     #3000,(R1)      ;CHECK TO INSURE THESE BITS SET.
025226  001410                          BEQ     3$              ;IF NOT, BRANCH.
                                                                ;SET THE MAINTENANCE BIT, THE NON-
025230  052711  000400                  BIS     #400,(R1)       ;EXISTENT MEMORY BIT AND THE CLEAR
                                                                ;NON-EXISTENT MEMORY INTERRUPT BIT.
025234  032711  002400                  BIT     #2400,(R1)      ;IS THIS A DH-11? (BITS 8 AND 10 SHOULD
                                                                ;CLEAR IF THIS IS A DH11.)

025240  001003                          BNE     3$              ;IF NOT, CHECK TO SEE IF THIS IS A MODEM CONTROL.
025242  042711  001000                  BIC     #1000,(R1)      ;CLEAR MAINTENANCE BIT.
025246  010122                          MOV     R1,(R2)+        ;SAVE THE ADDRESS IN THE DH ADR TABLE.


025250  020127  163760          3$:     CMP     R1,#163760      ;HAVE WE REACHED THE TOP OF THE FLOATING ADDRESSES.
025254  001406                          BEQ     5$              ;IF YES, GET OUT.
025256  062701  000020                  ADD     #20,R1          ;IF NOT, UPDATE ADDRESS AND
025262  000746                          BR      1$              ;GO CHECK IT.

025264  012716  025250          4$:     MOV     #3$,(SP)        ;IF DH ADDRESS DOES NOT RESPOND, GO TO 3$.
025270  000002                          RTI

                                ;TEST FOR MODEM CONTROL ADDRESS

025272  012737  025324  000004  5$:     MOV     #6$,@#4         ;SETUP FOR NON-EXISTENT MEMORY TRAP.
025300  012701  170500                  MOV     #170500,R1      ;R1=FIRST ADDRESS TO BE TESTED.
025304  005711                  21$:    TST     (R1)            ;SEE IF ADDRESS RESPONDS.
025306  010123                          MOV     R1,(R3)+        ;IF IT DOES, THIS IS A MODEM CONTROL,
                                                                ;SO SAVE THE ADDRESS.
025310  020127  170670          23$:    CMP     R1,#170670      ;HAVE WE REACHED THE TOP OF THE MODEM ADDRESSES?
025314  001406                          BEQ     22$             ;IF YES, GET OUT.
```

```
025316  062701  000010                  ADD     #10,R1          ;IF NOT, UPDATE ADDRESS AND
025322  000770                          BR      21$             ;GO CHECK IT.

025324  012716  025310          6$:     MOV     #23$,(SP)       ;IF DM ADDRESS DOES NOT RESPOND, GO TO 23$.
025330  000002                          RTI

025332  012637  000004          22$:    MOV     (SP)+,@#4       ;RESTORE TRAP VECTOR.
025336  162702  027764                  SUB     #DHADRS,R2      ;HAVE WE FOUND ANY DH11'S AT ALL?
025342  001003                          BNE     7$              ;IF YES, BRANCH
025344  104401  035537                  TYPE    ,MSG1           ;NO DH11'S WERE FOUND,
025350  000000                          HALT

025352  006202                  7$:     ASR     R2              ;R2 NOW CONTAINS THE NUMBER
025354  005000                          CLR     R0              ;OF DH'S FOUND.
025356  006100                  8$:     ROL     R0              ;FILL R0 WITH 1'S
025360  005200                          INC     R0              ;CORRESPONDING TO
025362  005302                          DEC     R2              ;THE NUMBER OF DH'S
025364  005702                          TST     R2              ;FOUND.
025366  001373                          BNE     8$
025370  010067  002642                  MOV     R0,$DHSEL       ;$DHSEL CONTAINS THE DH SELECTION PARAMETER.
                                                                ;IE. ALL DH'S FOUND WILL BE TESTED.

                                ;FIND DH VECTOR:
025374  012702  027764                  MOV     #DHADRS,R2      ;SETUP POINTER TO BEGINNING OF DH
025400  012705  030026                  MOV     #DHVEC,R5       ;ADDRESS TABLE AND VECTOR TABLE.
025404  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;SET IOT TRAP PRIORITY TO 7.
025412  012737  025522  000020          MOV     #12$,@#IOTVEC   ;SETUP IOT TRAP VECTOR.
025420  012703  000300                  MOV     #300,R3         ;START OF FLOATING VECTORS
025424  012704  000302                  MOV     #302,R4         ;PC OF IOT INSTR.

025430  010423                  9$:     MOV     R4,(R3)+        ;FILL VECTOR AREA WITH ADDRESS
                                                                ;OF NEXT INSTR (.+2)
025432  012724  000004                  MOV     #4,(R4)+        ;NEXT INSTRUCTION IS AN IOT TRAP.
025436  022324                          CMP     (R3)+,(R4)+     ;UPDATE R3+R4.
025440  020427  001000                  CMP     R4,#1000        ;HAVE WE REACHED TO TOP OF THE
                                                                ;VECTOR SPACE?
025444  101771                          BLOS    9$              ;IF NOT, REPEAT PROCESS.

025446  005712                  10$:    TST     (R2)            ;HAVE WE CHECK ALL DH'S?
025450  001441                          BEQ     13$             ;IF YES, GET OUT + CHECK FOR MODEM CONTROL'S VECTORS.

025452  005067  152320                  CLR     PS              ;ZERO CPU PRIORITY.
025456  052772  001000  000000          BIS     #1000,@(R2)     ;SET MAINTENANCE BIT
025464  052772  000300  000000          BIS     #300,@(R2)      ;ATTEMPT TO CAUSE RECEIVER
                                                                ;INTERRUPT.
025472  005000                          CLR     R0

025474  005200                  11$:    INC     R0              ;WAIT...
025476  001376                          BNE     11$
025500  104401  035567                  TYPE    ,MSG2           ;ERROR MSG-NO DH RECEIVER INTERRUPT OCCURRED.
025504  052772  004000  000000          BIS     #4000,@(R2)     ;DO A MASTER CLEAR
025512  042772  001000  000000          BIC     #1000,@(R2)     ;CLEAR MAINTENANCE BIT
025520  000752                          BR      10$

025522  011601                  12$:    MOV     (SP),R1
025524  042701  000007                  BIC     #7,R1           ;CLEAR GARBAGE.
```

```
025530  010125                          MOV     R1,(R5)+        ;SAVE VECTOR ADDRESS.
025532  022626                          CMP     (SP)+,(SP)+     ;POP STACK
025534  012716  025446                  MOV     #10$,(SP)       ;SETUP FOR RETURN.
025540  052772  004000  000000          BIS     #4000,@(R2)     ;DO A MASTER CLEAR
025546  042732  001000                  BIC     #1000,@(R2)+    ;CLEAR MAINTENANCE BIT.
025552  000002                          RTI

                                ;FIND MODEM CONTROL VECTORS:

025554  012702  030066          13$:    MOV     #DMADRS,R2      ;SET POINTERS TO BEGINNING OF
025560  012705  030130                  MOV     #DMVEC,R5       ;ADR TABLE & VECTOR TABLE.
025564  012737  025646  000020          MOV     #16$,@#IOTVEC   ;SET IOT TRAP VECTOR.

025572  005712                  14$:    TST     (R2)            ;HAVE WE CHECKED ALL DM'S?
025574  001441                          BEQ     17$             ;IF YES, GET OUT.
025576  005067  152174                  CLR     PS              ;ZERO CPU PRIORITY
025602  052772  001000  000000          BIS     #1000,@(R2)     ;SET MAINTENANCE BIT.
025610  052772  000300  000000          BIS     #300,@(R2)      ;ATTEMPT TO CAUSE INTERRUPT.
025616  005000                          CLR     R0

025620  005200                  15$:    INC     R0              ;WAIT....
025622  001376                          BNE     15$
025624  104401  035634                  TYPE    ,MSG3           ;ERROR MSG - NO MODEM CONTROL INTERRUPT OCCURRED.
025630  052772  004000  000000          BIS     #4000,@(R2)     ;CLEAR BITS PREVIOUSLY SET.
025636  042772  001000  000000          BIC     #1000,@(R2)     ;CLEAR MAINTENANCE BIT.
025644  000752                          BR      14$

025646  011601                  16$:    MOV     (SP),R1
025650  162701  000004                  SUB     #4,R1           ;CALCULATE VECTOR ADDRESS.
025654  010125                          MOV     R1,(R5)+        ;SAVE VECTOR ADDRESS.
025656  022626                          CMP     (SP)+,(SP)+     ;POP STACK.
025660  012716  025572                  MOV     #14$,(SP)       ;SETUP FOR RETURN.
025664  052772  004000  000000          BIS     #4000,@(R2)     ;CLEAR BITS PREVIOUSLY SET.
025672  042732  001000                  BIC     #1000,@(R2)+    ;CLEAR MAINTENANCE BIT AND
                                                                ;POINT TO NEXT MODEM CONTROL ADDRESS.
025676  000002                          RTI

025700  012737  021000  000020  17$:    MOV     #$SCOPE,@#IOTVEC  ;RESTORE IOT VECTOR FOR SCOPE ROUTINE.
025706  012600                          MOV     (SP)+,R0        ;RESTORE R0.
025710  012703  000300                  MOV     #300,R3         ;START OF FLOATING VECTORS.
025714  012704  000302                  MOV     #302,R4

025720  010423                  18$:    MOV     R4,(R3)+        ;FILL VECTOR AREA WITH ADDRESS OF NEXT
                                                                ;INSTRUCTION (.+2).
025722  012724  000000                  MOV     #0,(R4)+        ;NEXT INSTRUCTION IS A HALT.
025726  022324                          CMP     (R3)+,(R4)+     ;UPDATE R3 & R4.
025730  020427  001000                  CMP     R4,#1000        ;ARE WE DONE?
025734  101771                          BLOS    18$             ;IF NOT, REPEAT UNTIL ADDRESSES
                                                                ;377 TO 777 ARE DONE.
025736  013701  030026                  MOV     @#DHVEC,R1      ;LET R1 POINT TO 1ST DH VECTOR ADDRESS.
025742  005737  030030                  TST     @#DHVEC+2       ;IS THERE MORE THAN ONE VECTOR?
025746  001403                          BEQ     26$             ;BRANCH IF NOT.
025750  163701  030030                  SUB     @#DHVEC+2,R1    ;DETERMINE NUMBER OF ADDRESSES
                                                                ;BETWEEN DH VECTORS (10(8) OR 20(8)).
025754  005401                          NEG     R1              ;MAKE R1 POSITIVE.
025756  010167  002246          26$:    MOV     R1,ADRVEC       ;SAVE THAT NUMBER.
```

```
025762  032777  000002  153150          BIT     #BIT1,@SWR      ;SHOULD DEVICE MAP BE TYPED OUT?
025770  001442                          BEQ     20$             ;IF NOT, RETURN.
025772  104401                          TYPE                    ;TYPEOUT MAP OF DH & MODEM CONTROL'S
025774  035707                          DEVMAP                  ;FOUND.
025776  012701  027764                  MOV     #DHADRS,R1      ;R1=BEGINNING OF DH ADDRESS TABLE.
026002  012702  030026                  MOV     #DHVEC,R2       ;R2=BEGINNING OF DH VECTOR TABLE.
026006  012703  030066                  MOV     #DMADRS,R3      ;R3=BEGINNING OF MODEM CONTROL ADDRESS TABLE.
026012  012704  030130                  MOV     #DMVEC,R4       ;R4=BEGINNING OF MODEM CONTROL VECTOR TABLE.

026016  012146                  19$:    MOV     (R1)+,-(SP)     ;MOVE DATA TO BE TYPED
026020  104403                          TYPOS                   ;TYPE DATA
026022     006                          .BYTE   6
026023     001                          .BYTE   1
026024  012246                          MOV     (R2)+,-(SP)     ;MOVE DATA TO BE TYPED
026026  104403                          TYPOS                   ;TYPE DATA
026030     005                          .BYTE   5
026031     000                          .BYTE   0
026032  104401  035703                  TYPE    ,SPACE
026036  012346                          MOV     (R3)+,-(SP)     ;MOVE DATA TO BE TYPED.
026040  104403                          TYPOS                   ;TYPE DATA.
026042     006                          .BYTE   6
026043     001                          .BYTE   1
026044  104401  035703                  TYPE    ,SPACE
026050  012446                          MOV     (R4)+,-(SP)     ;MOVE DATA TO BE TYPED.
026052  104403                          TYPOS                   ;TYPE DATA.
026054     005                          .BYTE   5
026055     000                          .BYTE   0
026056  104401                          TYPE                    ;TYPE A CARRIAGE RETURN & LINE FEED.
026060  001227                          $CRLF
026062  005711                          TST     (R1)            ;HAVE WE TYPED ALL DH ENTRIES?
026064  001354                          BNE     19$             ;IF NOT, DO IT AGAIN.
026066  005713                          TST     (R3)            ;HAVE WE TYPED ALL DM ENTRIES?
026070  001352                          BNE     19$             ;IF NOT - ONE MORE TIME.
026072  104401  001227                  TYPE    ,$CRLF
026076  000207                  20$:    RTS     PC              ;IF YES, GO BACK TO MAIN PROGRAM.

                                ;THIS ROUTINE IS USED TO ACCEPT INPUT PARAMETERS FROM THE CONSOLE
                                ;TELETYPE

026100  104401                  INPARA: TYPE
026102  035442                          VCWC                    ;"ASK FOR NO. ADDRESSES BETWEEN VECTORS"
026104  104412                          RDOCT                   ;READ OCTAL NO. FM TTY
026106  012600                          MOV     (SP)+,R0        ;GET THE NO. HE TYPED
026110  001407                          BEQ     3$              ;BR IF HE TYPED <CR>
026112  022700  000010                  CMP     #10,R0          ;10(8) ADDRESSES BETWEEN VECTORS ?
026116  001406                          BEQ     4$              ;BR IF YES
026120  022700  000020                  CMP     #20,R0          ;20(8) ADDRESSES BETWEEN VECTORS ??
026124  001403                          BEQ     4$              ;BR IF YES
026126  000764                          BR      INPARA          ;ASK ALL OVER AGAIN
026130  012700  000020          3$:     MOV     #20.R0          ;SET UP CONSTANT FOR 20(8) ADDRESSES
026134  000207                  4$:     RTS     PC              ;RETURN TO CALLER


026136  012700  177777          INPARC: MOV     #-1,R0          ;SET FLAG IN R0
026142  000167  154012                  JMP     BEGINA          ;GO ASK FOR SELECT PARAMETER
```

```
026146  012767  177777  001546  INPARX: MOV     #-1,VCFLG       ;SET SETUP FLAG
026154  000167  154000                  JMP     BEGINA          ;GO START UP

026160  013701  027764          INPAR:  MOV     @#DHADRS,R1     ;MOVE ADDRESS OF FIRST DH INTO R1.
026164  032777  000001  152746          BIT     #BIT0,@SWR      ;ARE PARAMETERS TO BE INPUT MANUALLY?
026172  001405                          BEQ     2$              ;BRANCH IF NOT.
026174  104401                  1$:     TYPE                    ;ASK FOR DEVICE ADDRESS
026176  035017                          INMSG1
026200  104412                          RDOCT                   ;READ IN WHAT IS TYPED
026202  012601                          MOV     (SP)+,R1        ;GET THE NO. HE TYPED
026204  001403                          BEQ     INPAR1          ;BR IF DEFAULT
026206  004767  000160          2$:     JSR     PC,CHKADR       ;GO CHECK VALIDITY OF THE ADDR
026212  000770                          BR      1$              ;ERROR BRANCH

026214  013701  030026          INPAR1: MOV     @#DHVEC,R1      ;MOVE FIRST DH VECTOR INTO R1.
026220  032777  000001  152712          BIT     #BIT0,@SWR      ;ARE PARAMETERS TO BE INPUT MANUALLY?
026226  001405                          BEQ     2$              ;BRANCH IF NOT.
026230  104401                  1$:     TYPE                    ;ASK FOR VECTOR ADDRESS
026232  035063                          INMSG2
026234  104412                          RDOCT                   ;READ IN WHAT HE TYPES
026236  012601                          MOV     (SP)+,R1        ;GET THE ADDRESS
026240  001403                          BEQ     INPAR3          ;BR IF DEFAULT
026242  004767  000240          2$:     JSR     PC,CHKVCT       ;GO CHECK VALIDITY OF VECTOR
026246  000770                          BR      1$              ;ERROR BRANCH

026250  013701  030236          INPAR3: MOV     @#$DHSEL,R1     ;MOVE DEVICE SELECTION PARAMETER INTO R1.
026254  005700                          TST     R0              ;DID WE START AT 210?
026256  100404                          BMI     2$              ;BRANCH IF YES.
026260  032777  000001  152652          BIT     #BIT0,@SWR      ;IS PARAMETER TO BE INPUT MANUALLY?
026266  001405                          BEQ     1$              ;BRANCH IF NOT.
026270  104401                  2$:     TYPE                    ;ASK FOR DEVICE SELECTION PARAMETER
026272  035132                          INMSG3
026274  104412                          RDOCT                   ;READ IN WHAT HE TYPES
026276  012601                          MOV     (SP)+,R1        ;GET THE SELECT PARAMETER
026300  001402                          BEQ     INPAR4          ;BR IF  DEFAULT
026302  010167  000724          1$:     MOV     R1,DHSEL        ;SET UP DH11 SELECTION PARAMETER

026306  005700                  INPAR4: TST     R0              ;DID WE START AT 210?
026310  100404                          BMI     3$              ;BRANCH IF YES.
026312  032777  000001  152620          BIT     #BIT0,@SWR      ;IS LINE SELECT PARAMETER TO BE INPUT MANUALLY?
026320  001410                          BEQ     1$              ;BRANCH IF NO.
026322  104401                  3$:     TYPE                    ;ASK FOR LINE SELECT PARAMETER
026324  035330                          INMSG6
026326  104412                          RDOCT                   ;GET WHAT HE TYPES
026330  012601                          MOV     (SP)+,R1        ;GET PARAMETER
026332  001403                          BEQ     1$              ;BR IF DEFAULT
026334  010167  000674                  MOV     R1,LINSEL       ;SET UP LINE SELECT PARAMETER
026340  000403                          BR      2$              ;CONTINUE
026342  012767  177777  000664  1$:     MOV     #-1,LINSEL      ;SET UP DEFAULT (ALL LINES)
026350  032777  000400  152562  2$:     BIT     #BIT8,@SWR      ;HALT AFTER SET UP ??
026356  001403                          BEQ     EXPAR           ;BR IF  NOT
026360  104401                          TYPE                    ;TYPE CONTINUE MESSAGE PRIOR TO HALTING
026362  035372                          INMSG7
026364  000000                          HALT                    ;DEPRESS CONTINUE TO RESUME TESTING
026366  000167  154242          EXPAR:  JMP     START2          ;GO START UP THE PROGRAM
```

```
026372  020127  160020          CHKADR: CMP     R1,#160020      ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
026376  002001                          BGE     1$              ;BR IF YES
026400  000437                          BR      4$              ;BR IF NOT
026402  020127  160420          1$:     CMP     R1,#160420      ;IS IT BELOW THE HIGH LIMIT?
026406  002401                          BLT     2$              ;BR IF YES
026410  000433                          BR      4$              ;BR IF NOT
026412  032701  000017          2$:     BIT     #17,R1          ;CORRECT BOUNDARY ?
026416  001030                          BNE     4$              ;BR IF NOT
026420  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC AROUND ERROR BRANCH
026424  012702  027622                  MOV     #DHADTB,R2      ;POINT TO BEGIN OF ADDR TABLE
026430  032777  000001  152502          BIT     #BIT0,@SWR      ;ARE WE AUTOSIZING?
026436  001011                          BNE     3$              ;BRANCH IF NOT.
026440  012703  027764                  MOV     #DHADRS,R3      ;POINT TO BEGINNING OF AUTOSIZER
                                                                ;DH ADDRESS TABLE.
026444  016704  001566                  MOV     $DHSEL,R4
026450  012322                  6$:     MOV     (R3)+,(R2)+     ;MOVE CONTENTS OF AUTOSIZER DH TABLE
                                                                ;TO THE TABLE USED BY PROGRAM.
026452  006204                          ASR     R4
026454  005704                          TST     R4              ;HAVE WE MOVED ALL TABLE ENTRIES?
026456  001374                          BNE     6$              ;BRANCH IF NOT--ONE MORE TIME.
026460  000411                          BR      5$              ;RETURN TO INPUT ROUTINES.
026462  010122                  3$:     MOV     R1,(R2)+        ;SETR UP A TABLE ENTRY
026464  062701  000020                  ADD     #20,R1          ;GENERATE NEXT DH11 ADDR
026470  022702  027662                  CMP     #DHADTB+40,R2   ;END OF TABLE ?
026474  001372                          BNE     3$              ;BR IF NOT
026476  000402                          BR      5$              ;RETURN TO INPUT ROUTINES
026500  104401                  4$:     TYPE                    ;TELL HIM HE GOOFED
026502  035203                          INMSG4
026504  000207                  5$:     RTS     PC              ;RETURN TO INPUT ROUTINES

026506  020127  000300          CHKVCT: CMP     R1,#300         ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
026512  002001                          BGE     1$              ;BR IF YES
026514  000436                          BR      4$              ;BR IF NOT
026516  020127  001000          1$:     CMP     R1,#1000        ;IS IT BELOW THE HIGH LIMIT?
026522  002401                          BLT     2$              ;BR IF YES
026524  000432                          BR      4$              ;BR IF NOT
026526  032701  000007          2$:     BIT     #7,R1           ;CORRECT BOUNDARY ?
026532  001027                          BNE     4$              ;BR IF NOT
026534  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC AROUND ERROR BRANCH
026540  012702  027662                  MOV     #DHVCTB,R2      ;POINT TO BEGIN OF VECTOR TABLE
026544  032777  000001  152366          BIT     #BIT0,@SWR      ;ARE WE AUTOSIZING?
026552  001011                          BNE     3$              ;BRANCH IF NOT.
026554  012703  030026                  MOV     #DHVEC,R3       ;POINT TO BEGINING OF AUTOSIZER
                                                                ;DH VECTOR TABLE.
026560  016704  001452                  MOV     $DHSEL,R4
026564  012322                  6$:     MOV     (R3)+,(R2)+     ;MOVE CONTENTS OF AUTOSIZER VECTOR
                                                                ;TABLE TO TABLE USED BY PROGRAM.
026566  006204                          ASR     R4
026570  005704                          TST     R4              ;HAVE WE MOVED ALL TABLE ENTRIES?
026572  001374                          BNE     6$              ;BRANCH IF NOT--ONE MORE TIME.
026574  000410                          BR      5$              ;RETURN TO INPUT ROUTINES.
026576  010122                  3$:     MOV     R1,(R2)+        ;SETR UP A TABLE ENTRY
026600  060001                          ADD     R0,R1           ;GENERATE NEXT DH11 ADDR
026602  022702  027722                  CMP     #DHVCTB+40,R2   ;END OF TABLE ?
```

```
026606  001373                          BNE     3$              ;BR IF NOT
026610  000402                          BR      5$              ;RETURN TO INPUT ROUTINES
026612  104401                  4$:     TYPE                    ;TELL HIM HE GOOFED
026614  035254                          INMSG5
026616  000207                  5$:     RTS     PC              ;RETURN TO INPUT ROUTINES

                                ;THESE TWO ROUTINES SERVICE UNEXPECTED BUS ERROR AND RSVD INSTR TRAPS

026620  012767  000340  152354  BUSER:  MOV     #340,$TMPO      ;SAVE THE PSW
026626  010667  152344                  MOV     SP,$REG6        ;SAVE THE SP
026632  012601                          MOV     (SP)+,R1        ;GET THE TRAP PC
026634  012602                          MOV     (SP)+,R2        ;GET THE TRAP PSW
026636  116700  152240                  MOVB    $TSTNM,R0       ;GET TEST NO.
026642  012706  001100                  MOV     #STACK,SP       ;RESET THE STACK POINTER
026646  004767  175600                  JSR     PC,SUER3        ;GO SET UP ERROR INFO
026652  012767  026662  152230          MOV     #1$,$LPERR      ;ALWAYS COME BACK TO 1$
026660  104027                          ERROR   27              ;UNEXPECTED BUS ERROR TRAP
026662  000005                  1$:     RESET                   ;PREPARE TO RESTART
026664  004767  000240                  JSR     PC,CHPS1        ;GO CLEAR PSW
026670  000167  154034                  JMP     REST1           ;GO RESTART THE PROGRAM

026674  012767  000340  152300  RESERR: MOV     #340,$TMPO      ;SAVE THE PSW
026702  010667  152270                  MOV     SP,$REG6        ;SAVE THE SP
026706  012601                          MOV     (SP)+,R1        ;GET THE TRAP PC
026710  012602                          MOV     (SP)+,R2        ;GET THE TRAP PSW
026712  116700  152164                  MOVB    $TSTNM,R0       ;GET TEST NO.
026716  012706  001100                  MOV     #STACK,SP       ;RESET THE STACK POINTER
026722  004767  175524                  JSR     PC,SUER3        ;GO SET UP ERROR INFO
026726  012767  026736  152154          MOV     #1$,$LPERR      ;ALWAYS COME BACK TO 1$
026734  104030                          ERROR   30              ;UNEXPECTED RSVD INSTR ERROR TRAP
026736  000005                  1$:     RESET                   ;PREPARE TO RESTART
026740  004767  000164                  JSR     PC,CHPS1        ;GO CLEAR PSW
026744  000167  153760                  JMP     REST1           ;GO RESTART THE PROGRAM

                                ;THIS ROUTINE IS CALLED WHEN A TEST NEEDS TO RESTORE THE TRAP
                                ;CATCHER IN THE DH11 VECTOR

026750  016703  000252          RESTRF: MOV     DHVCT,R3        ;GET VECTOR ADDRESS
026754  010313                          MOV     R3,(R3)         ;RESTORE THE TRAP CATCHER
026756  062723  000002                  ADD     #2,(R3)+
026762  005023                          CLR     (R3)+
026764  010313                          MOV     R3,(R3)
026766  062723  000002                  ADD     #2,(R3)+
026772  005023                          CLR     (R3)+
026774  000207                          RTS     PC              ;RETURN TO CALLING TEST

                                ;THIS ROUTINE CALLED BY ANY TEST THAT NEEDS A TIMING WAIT LOOP
                                ;"TIMEA" IS INITIALIZED BY THE CALLING ROUTINE TO THE MINIMUM REQUIRED
                                ;VALUE AND "TIMEB" IS CLEARED TO 000000. IF A TIME OUT OCCURS THIS
                                ;ROUTINE WILL MOVE THE RETURN PC AROUND THE "LOOP" BRANCH BACK IN
                                ;THE ROUTINE THAT CALLED IT TO ALLOW REPORTING AN ERROR MESSAGE

026776  005267  001272          TIMEIT: INC     TIMEB           ;COUNT B
027002  001005                          BNE     1$              ;BR IF NOT ZERO
027004  005367  001262                  DEC     TIMEA           ;COUNT TIME A
027010  001002                          BNE     1$              ;BR IF NO TIMEOUT
```

```
027012  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC TO ALLOW ERROR REPORT
027016  000207                  1$:     RTS     PC              ;RETURN TO THE CALLING TEST

                                ;THIS ROUTINE CALLED BY THE AUTO ECHO TEST TO SET UP FOR TRANSFERRING
                                ;A BINARY COUNT TEST PATTERN ON ALL LINES

027020  012767  000020  152170  SETALL: MOV     #20,$TMP6       ;SET UP SIXTEEN LINES
027026  005002                          CLR     R2              ;INIT A TABLE INDEX REG
027030  012703  000200                  MOV     #200,R3         ;SET UP TO GENERATE HI BYTE OF EXPECTED DATA
027034  012704  000001                  MOV     #1,R4           ;SET UP INDEX REG TO ODD BYTES
027040  005011                          CLR     (R1)            ;START WITH LINE 00
027042  012761  037244  000006  1$:     MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDR REG
027050  012761  177400  000010          MOV     #-400,BCR(R1)   ;SET UP BYTE COUNT REG
027056  012761  031403  000004          MOV     #31403,LPR(R1)  ;SET UP FOR 4800 BAUD/8 BIT CHARS
027064  105062  036244                  CLRB    RBUF(R2)        ;START WITH DATA CHAR OF 000
027070  110364  036244                  MOVB    R3,RBUF(R4)     ;SET UP HIGH BYTE OF EXPECTED DATA
027074  005211                          INC     (R1)            ;GEN NEW LINE NO. IN SCR
027076  005203                          INC     R3              ;UPDATE THE POINTERS AND DATA
027100  062702  000002                  ADD     #2,R2
027104  062704  000002                  ADD     #2,R4
027110  005367  152102                  DEC     $TMP6           ;COUNT ONE LINE DONE
027114  001352                          BNE     1$              ;BR TIL ALL 16 SET UP
027116  016767  000112  000356          MOV     LINSEL,LINACT   ;SET SOFTWARE FLAG FOR ALL LINES ACTIVE
027124  005011                          CLR     (R1)            ;PUT SCR REG BACK TO LINE 00
027126  000207                          RTS     PC              ;RETURN TO AUTO ECHO TEST


                                ;THIS ROUTINE IS CALLED TO SET PSW PRIORITY TO 000 IN ORDER
                                ;TO BE LSI11 COMPATIBLE

027130  012746  000000          CHPS1:  MOV     #0,-(SP)        ;NEW PSW
027134  012746  027142                  MOV     #1$,-(SP)       ;NEW PC
027140  000002                          RTI                     ;CHANGE PSW
027142  000207                  1$:     RTS     PC              ;RETURN TO CALLING TEST

                                ;THIS ROUTINE DOES THE SAME THING EXCEPT IT SET THE PSW
                                ;PRIORITY TO 340 (LEVEL 7 ) TO LOCK OUT INTRS

027144  012746  000340          CHPS2:  MOV     #340,-(SP)      ;NEW PSW
027150  012746  027156                  MOV     #1$,-(SP)       ;NEW PC
027154  000002                          RTI                     ;CHANGE THE PSW
027156  000207                  1$:     RTS     PC              ;RETURN TO CALLING TEST

                                ;THIS ROUTINE IS ALSO FOR LSI11 COMPATIBILITY AND IT IS CALLED
                                ;TO SAVE THE PSW IN "$TMPO"

027160  005046                  SAPS:   CLR     -(SP)           ;TEMP STORAGE TO SAVE PSW
027162  016746  150646                  MOV     34,-(SP)        ;SAVE TRAP VECTOR POINTER
027166  012767  027176  150640          MOV     #1$,34          ;GO TO 1$ ON TRAP
027174  104400                          TRAP                    ;GO TO IT
027176  016666  000002  000006  1$:     MOV     2(SP),6(SP)     ;GET PSW SAVED
027204  012716  027212                  MOV     #2$,(SP)        ;GO TO 2$ ON RTI
027210  000002                          RTI
027212  012667  150616          2$:     MOV     (SP)+,34        ;RESTORE VECTOR
027216  012667  151760                  MOV     (SP)+,$TMPO     ;FINALLY SAVE PSW IN $TMPO
027222  000207                          RTS     PC
```

9058

```
                                ;;*************************************************************
                                ;ADDITIONAL PROGRAM CONSTANTS AND VARIABLES
                                ;;*************************************************************

        000002                          NRC=2           ;INDEX CONST. TO ACCESS NEXT RCVD CHAR REG
        000004                          LPR=4           ;INDEX CONST. TO ACCESS LINE PARAMETER REG.
        000006                          CAR=6           ;INDEX CONST. TO ACCESS CURRENT ADDRESS REG.
        000010                          BCR=10          ;INDEX CONST. TO ACCESS BYTE COUNT REG.
        000012                          BAR=12          ;INDEX CONST. TO ACCESS BUFFER ACTIVE REG.
        000014                          BKR=14          ;INDEX CONST. TO ACCESS BREAK CONTROL REG.
        000016                          SSR=16          ;INDEX CONST. TO ACCESS SILO STATUS REG.

027224  000000                  DHADR:  0               ;HOLDS THE "SCR" ADDRESS OF THE DH11 UNDER TEST
027226  000000                  DHVCT:  0               ;HOLDS THE 1ST VECTOR ADDRESS OF THE DH11 UNDER TEST
027230  000000                  SELMSK: 0               ;BIT TST MARKER FOR SELECTING DH11'S
027232  000003                  DHSEL:  3               ;SPECIFIES DH11'S SELECTED FOR TEST
027234  177777                  LINSEL: 177777          ;SPECIFIES LINES TO TEST
027236  000000                  LINMSK: 0               ;MARKER USED TO TEST FOR LINES TO TEST
027240  000000                  LMSK1:  0               ;ALTERNATE MARKER TO SUPPORT THE
                                                        ;SELECT LINES FEATURE
027242  000004                  MSTCLR: .BLKW   4       ;FOUR WORD ADDRESS TABLE USED BY THE TEST THAT
                                                        ;CHECKS OPERATION OF "MASTER CLR"

027252  177777                  PATRNA: 177777          ;BIT PATTERNS USED WITH "CAR" AND "BCR" TESTS
027254  125252                          125252
027256  052525                          052525
027260  000000                          000000          ;TABLE TERMINATOR

027262  000060                  PATRNB: 60              ;BIT PATTERNS USED IN "CAR" MEM EXT BIT TEST
027264  000300                          300
027266  000020                          20
027270  000100                          100
027272  000040                          40
027274  000200                          200
027276  000000                          0               ;TABLE TERMINATOR
027300  000000                          0               ;TABLE TERMINATOR

                                ;THIS TABLE STORES THE BYTE COUNT AND LINE PARAMETERS FOR THE
                                ;8 SUBTESTS IN THE MULTILINE PARITY/DATA TEST

027302  177400                  PRTYTB: -400                    ;256 CHARS
027304  027363                          27363                   ;2400 BAUD - ODD PARITY - 8 BITS
027306  177400                          -400                    ;256 CHARS
027310  027323                          27323                   ;2400 BAUD - EVEN PARITY - 8 BITS
027312  177600                          -200                    ;128 CHARS
027314  027362                          27362                   ;2400 BAUD - ODD PARITY - 7 BITS
027316  177600                          -200                    ;128 CHARS
027320  027322                          27322                   ;2400 BAUD - EVEN PARITY - 7 BITS
027322  177700                          -100                    ;64 CHARS
027324  027361                          27361                   ;2400 BAUD - ODD PARITY - 6 BITS
027326  177700                          -100                    ;64 CHARS
027330  027321                          27321                   ;2400 BAUD - EVEN PARITY - 6 BITS
027332  177740                          -40                     ;32 CHARS
027334  027360                          27360                   ;2400 BAUD - ODD PARITY - 5 BITS
027336  177740                          -40                     ;32 CHARS
027340  027320                          27320                   ;2400 BAUD - EVEN PARITY - 5 BITS
```

```
                                        ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
                                        ;TEST (ALL 1'S DATA TABLE)

027342  100377                          AETAB:  100377          ;TEST DATA FOR LINE 00
027344  100777                                  100777          ;TEST DATA FOR LINE 01
027346  101377                                  101377
027350  101777                                  101777
027352  102377                                  102377
027354  102777                                  102777
027356  103377                                  103377
027360  103777                                  103777
027362  104377                                  104377
027364  104777                                  104777
027366  105377                                  105377
027370  105777                                  105777
027372  106377                                  106377
027374  106777                                  106777
027376  107377                                  107377
027400  107777                                  107777          ;TEST DATA FOR LINE 17

                                        ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
                                        ;TEST (ALL 0'S DATA TABLE)

027402  100000                          AETAB0: 100000          ;TEST DATA FOR LINE 00
027404  100400                                  100400          ;TEST DATA FOR LINE 01
027406  101000                                  101000
027410  101400                                  101400
027412  102000                                  102000
027414  102400                                  102400
027416  103000                                  103000
027420  103400                                  103400
027422  104000                                  104000
027424  104400                                  104400
027426  105000                                  105000
027430  105400                                  105400
027432  106000                                  106000
027434  106400                                  106400
027436  107000                                  107000
027440  107400                                  107400          ;TEST DATA FOR LINE 17

                                        ;THIS TABLE USED BY THE AUTO ECHO TEST 2 TO RESET ACTIVE BIT WHEN A
                                        ;LINE IS DONE

027442  000001                          LINBIT: BIT00                   ;DEACTIVATE LINE 00
027444  000002                                  BIT01                   ;DEACTIVATE LINE 01
027446  000004                                  BIT02
027450  000010                                  BIT03
027452  000020                                  BIT04
027454  000040                                  BIT05
027456  000100                                  BIT06
027460  000200                                  BIT07
027462  000400                                  BIT08
027464  001000                                  BIT09
027466  002000                                  BIT10
027470  004000                                  BIT11
```

```
027472  010000                    BIT12
027474  020000                    BIT13
027476  040000                    BIT14
027500  100000                    BIT15                   ;DEACTIVATE LINE 17

027502  000000            LINACT: 0                       ;MAINTAINS STATUS OF ACTIVE LINES
                                                          ;DURING AUTO ECHO TEST 2


                          ;THIS TABLE CONTAINS 16. COUNTERS USED BYN THE MULTI-LINE
                          ;PARITY TEST TO KEEP TRACK OF TOTAL CHARS RECEIVED

027504  000020            MULPTB: .BLKW   16.             ;SIXTEEN WORD COUNTERS TABLE

                          ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE BREAK BIT
                          ;TEST

027544  120000            BRKTAB: 120000          ;TEST DATA FOR LINE 00
027546  120400                    120400          ;TEST DATA FOR LINE 01
027550  121000                    121000
027552  121400                    121400
027554  122000                    122000
027556  122400                    122400
027560  123000                    123000
027562  123400                    123400
027564  124000                    124000
027566  124400                    124400
027570  125000                    125000
027572  125400                    125400
027574  126000                    126000
027576  126400                    126400
027600  127000                    127000
027602  127400                    127400          ;TEST DATA FOR LINE 17

027604  131177            RGMSK1: 131177          ;MASK TO SPECIFY R/W BITS FOR NORMAL "SCR" REG TEST
027606  046600            RGMSK2: 46600           ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR NORMAL MODE TEST
027610  177767            RGMSK3: 177767          ;MASK TO SPECIFY R/W BITS IN "LPR"
027612  177777            RGMSK4: 177777          ;MASK TO SPECIFY R/W BITS IN "BKR"
027614  100077            RGMSK5: 100077          ;MASK TO SPECIFY R/W BITS IN "SSR"
027616  042200            RGMSK6: 42200           ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR MAINT. MODE TEST
027620  030100            INTMSK: 30100           ;MASK USED TO SELECT INTR BITS TO TEST

                          ;DH11 ADDRESS TABLE - THIS TABLE CONTAINS THE "SCR" ADDRESS FOR UP TO
                          ;SIXTEEN DH11'S

027622  160020            DHADTB: 160020          ;ADDRESS OF FIRST DH11
027624  160040                    160040          ;ADDRESS OF SECOND DH11
027626  160060                    160060
027630  160100                    160100
027632  160120                    160120
027634  160140                    160140
027636  160160                    160160
027640  160200                    160200
027642  160220                    160220
027644  160240                    160240
027646  160260                    160260
```

```
027650  160300                          160300
027652  160320                          160320
027654  160340                          160340
027656  160360                          160360
027660  160400                          160400          ;ADDRESS OF THE LAST DH11

                                ;DH11 VECTOR TABLE - THIS TABLE CONTAINS THE VECTOR ADDRESSES FOR UP
                                ;TO SIXTEEN DH11'S

027662  000330                  DHVCTB: 330             ;ADDRESS OF VECTOR FOR FIRST DH11
027664  000350                          350             ;ADDRESS OF VECTOR FOR SECOND DH11
027666  000370                          370
027670  000410                          410
027672  000430                          430
027674  000450                          450
027676  000470                          470
027700  000510                          510
027702  000530                          530
027704  000550                          550
027706  000570                          570
027710  000610                          610
027712  000630                          630
027714  000650                          650
027716  000670                          670
027720  000710                          710             ;ADDRESS OF VECTOR FOR LAST DH11

027722  000000                  VCFLG:  0               ;VECTOR SET UP FLAGG


                                ;BR PRIORITY LEVEL TABLE - THIS TABLE CONTAINS THE PRIORITY LEVELS
                                ;FOR UP TO SIXTEEN DH11'S - THE RCVR LEVEL IS STORED IN THE LOW BYTE
                                ;AND THE XMTTR LEVEL IN THE HIGH BYTE

027724  120240                  BRLVL:  120240          ;BRLEVELS FOR FIRST DH11
027726  120240                          120240          ;BR LEVELS FOR SECOND DH11
027730  120240                          120240
027732  120240                          120240
027734  120240                          120240
027736  120240                          120240
027740  120240                          120240
027742  120240                          120240
027744  120240                          120240
027746  120240                          120240
027750  120240                          120240
027752  120240                          120240
027754  120240                          120240
027756  120240                          120240
027760  120240                          120240
027762  120240                          120240          ;BR LEVELS FOR LAST DH11

                                ;THIS DH ADDRESS TABLE IS FILLED BY THE AUTOSIZER.

027764                          DHADRS:
027764  000000                  .WORD   0
027766  000000                  .WORD   0
027770  000000                  .WORD   0
```

```
027772  000000          .WORD   0
027774  000000          .WORD   0
027776  000000          .WORD   0
030000  000000          .WORD   0
030002  000000          .WORD   0
030004  000000          .WORD   0
030006  000000          .WORD   0
030010  000000          .WORD   0
030012  000000          .WORD   0
030014  000000          .WORD   0
030016  000000          .WORD   0
030020  000000          .WORD   0
030022  000000          .WORD   0
030024  000000          .WORD   0

                        ;THIS DH VECTOR TABLE IS FILLED BY THE AUTOSIZER.

030026                  DHVEC:
030026  000000          .WORD   0
030030  000000          .WORD   0
030032  000000          .WORD   0
030034  000000          .WORD   0
030036  000000          .WORD   0
030040  000000          .WORD   0
030042  000000          .WORD   0
030044  000000          .WORD   0
030046  000000          .WORD   0
030050  000000          .WORD   0
030052  000000          .WORD   0
030054  000000          .WORD   0
030056  000000          .WORD   0
030060  000000          .WORD   0
030062  000000          .WORD   0
030064  000000          .WORD   0

                        ;THIS DM ADDRESS TABLE IS FILLED BY THE AUTOSIZER.

030066                  DMADRS:
030066  000000          .WORD   0
030070  000000          .WORD   0
030072  000000          .WORD   0
030074  000000          .WORD   0
030076  000000          .WORD   0
030100  000000          .WORD   0
030102  000000          .WORD   0
030104  000000          .WORD   0
030106  000000          .WORD   0
030110  000000          .WORD   0
030112  000000          .WORD   0
030114  000000          .WORD   0
030116  000000          .WORD   0
030120  000000          .WORD   0
030122  000000          .WORD   0
030124  000000          .WORD   0
030126  000000          .WORD   0
```

```
                                        ;THIS DM VECTOR TABLE IS FILLED BY THE AUTOSIZER.

030130                                  DMVEC:
030130  000000                          .WORD   0
030132  000000                          .WORD   0
030134  000000                          .WORD   0
030136  000000                          .WORD   0
030140  000000                          .WORD   0
030142  000000                          .WORD   0
030144  000000                          .WORD   0
030146  000000                          .WORD   0
030150  000000                          .WORD   0
030152  000000                          .WORD   0
030154  000000                          .WORD   0
030156  000000                          .WORD   0
030160  000000                          .WORD   0
030162  000000                          .WORD   0
030164  000000                          .WORD   0
030166  000000                          .WORD   0

030170                                  $LNSEL:
030170  000001                          .WORD   1
030172  000002                          .WORD   2
030174  000004                          .WORD   4
030176  000010                          .WORD   10
030200  000020                          .WORD   20
030202  000040                          .WORD   40
030204  000100                          .WORD   100
030206  000200                          .WORD   200
030210  000400                          .WORD   400
030212  001000                          .WORD   1000
030214  002000                          .WORD   2000
030216  004000                          .WORD   4000
030220  010000                          .WORD   10000
030222  020000                          .WORD   20000
030224  040000                          .WORD   40000
030226  100000                          .WORD   100000

030230  000000                          ADRVEC: 0               ;ADDRESSES BETWEEN VECTORS - FILLED BY THE AUTOSIZER
030232  000000                          DHMCSR: 0               ;MODEM CONTROL CONTROL AND STATUS REGISTER.
030234  000000                          DHMLSR: 0               ;MODEM CONTROL LINE STATUS REGISTER.
030236  000000                          $DHSEL: 0               ;DEVICE SELECT PARAMETER - FILLLED BY THE AUTOSIZER.
030240     000                          DHRLVL: .BYTE   0               ;BR LEVEL FOR RCVR
030241     000                          DHTLVL: .BYTE   0               ;BR LEVEL FOR XMITTER
030242  000000                          DHNUM:  0               ;CONTAINS NUMBER OF THE DH11 UNDER TEST
030244  000000                          LINE:   0               ;CONTINES NUMBER OF THE LINE UNDER TEST
030246  000000                          LINEA:  0               ;LOCATION TO SAVE LINE NUMBER
                                        ;ADDRESS POINTERS TO SET UP TABLES WHEN INPUTTING PARAMETERS

030250  000000                          ADPTR:  0                       ;POINTS TO ADDRESS TABLE
030252  000000                          VCPTR:  0                       ;POINTS TO VECTOR TABLE
030254  000000                          BRPTR:  0                       ;POINTS TO BR LEVEL TABLE
                                        ;THE FOLLOWING TEN CONSTANTS ARE USED BY THE MODEM CONTROL PORTION OF THIS PROGRAM.
```

```
                000400          STEP=400
                000001          LINENA=1
                000002          TRMRDY=2
                000004          RS=4
                000010          SECTX=10
                000020          SECRX=20
                000040          CS=40
                000100          CO=100
                000200          RING=200
                002000          CLRMUX=2000

030256          000000          TDATA1: 0               ;DATA BUFFER FOR BASIC DATA TEST
030260          000037          TDATA2: 37              ;TEST DATA FOR FIVE BIT CHAR
030262          000077                  77              ;TEST DATA FOR SIX BIT CHAR
030264          000177                  177             ;TEST DATA FOR SEVEN BIT CHAR
030266          000377                  377             ;TEST DATA FOR EIGHT BIT CHAR
030270          000000          TITFLG: 0               ;FLAG TO ALLOW PRINTING TITLE ONLY ONCE
030272          000000          TIMEA:  0               ;GENERAL PURPOSE TIMERS
030274          000000          TIMEB:  0
030276          000000          TIMEC:  0               ;TIMER FOR TIMING TESTS
030300          000000          TNULL:  0               ;CONTAINS TWO NULL CHARS USED BY BREAK TEST
```

```
                                        ;;*************************************************************
                                        ;ERROR MESSAGE INFORMATION - MESSAGE BUFFERS AND POINTERS
                                        ;;*************************************************************

                                        ;INFORMATION FOR MESSAGE 1

030302  044104  030461  051040  EM1:    .ASCIZ  'DH11 REGISTER REFERENCE CAUSED TIMEOUT'
030310  043505  051511  042524
030316  020122  042522  042506
030324  042522  041516  020105
030332  040503  051525  042105
030340  052040  046511  047505
030346  052125     000
030351     040  050050  024503  DH1:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  REGADR'
030356  020040  020040  050050
030364  024523  020040  020040
030372  051450  024520  020040
030400  020040  042524  052123
030406  020040  042040  053105
030414  042101  020122  051040
030422  043505  042101  000122
                                        .EVEN
030430  001116  001202  001176  DT1:    .WORD   $ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,0
030436  001162  001164  001166
030444  000000

                                        ;INFORMATION FOR MESSAGE 2

030446  054523  052123  046505  EM2:    .ASCIZ  'SYSTEM CONTROL REGISTER ERROR'
030454  041440  047117  051124
030462  046117  051040  043505
030470  051511  042524  020122
030476  051105  047522  000122
030504  024040  041520  020051  DH2:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B'
030512  020040  024040  051520
030520  020051  020040  024040
030526  050123  020051  020040
030534  052040  051505  020124
030542  020040  042504  040526
030550  051104  020040  042522
030556  040507  051104  020040
030564  053440  051501  020040
030572  020040  051440  041057
030600     000
        030602                          .EVEN
030602  001116  001202  001176  DT2:    .WORD   $ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4,0
030610  001162  001164  001166
030616  001170  001172  000000

                                        ;INFORMATION FOR MESSAGE 3

030624  044104  030461  046440  EM3:    .ASCIZ  'DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG'
030632  051501  042524  020122
030640  046103  040505  020122
030646  040506  046111  042105
030654  052040  020117  046103
```

```
030662  020122  050123  041505
030670  043111  042511  020104
030676  042522  000107

                                ;INFORMATION FOR MESSAGE 4

030702  044514  042516  050040  EM4:    .ASCIZ  'LINE PARAMETER REGISTER ERROR'
030710  051101  046501  052105
030716  051105  051040  043505
030724  051511  042524  020122
030732  051105  047522  000122

                                ;INFORMATION FOR MESSAGE 5

030740  051102  040505  020113  EM5:    .ASCIZ  'BREAK CONTROL REGISTER ERROR'
030746  047503  052116  047522
030754  020114  042522  044507
030762  052123  051105  042440
030770  051122  051117     000

                                ;INFORMATION FOR MESSAGE 6

030775     123  046111  020117  EM6:    .ASCIZ  'SILO STATUS REGISTER ERROR'
031002  052123  052101  051525
031010  051040  043505  051511
031016  042524  020122  051105
031024  047522  000122

                                ;INFORMATION FOR MESSAGE 7

031030  052503  051122  047105  EM7:    .ASCIZ  'CURRENT ADDRESS REGISTER ERROR - LINE #XX'
031036  020124  042101  051104
031044  051505  020123  042522
031052  044507  052123  051105
031060  042440  051122  051117
031066  026440  046040  047111
031074  020105  054043  000130

                                ;INFORMATION FOR MESSAGE 10

031102  054502  042524  041440  EM10:   .ASCIZ  'BYTE COUNTER REGISTER ERROR - LINE #XX'
031110  052517  052116  051105
031116  051040  043505  051511
031124  042524  020122  051105
031132  047522  020122  020055
031140  044514  042516  021440
031146  054130     000

                                ;INFORMATION FOR MESSAGE 11

031151     125  042516  050130  EM11:   .ASCIZ  'UNEXPECTED DH11 RCVR INTERRUPT'
031156  041505  042524  020104
031164  044104  030461  051040
031172  053103  020122  047111
031200  042524  051122  050125
031206  000124
```

```
                                        ;INFORMATION FOR MESSAGE 12

031210  047125  054105  042520  EM12:   .ASCIZ  'UNEXPECTED DH11 XMITTR INTERRUPT'
031216  052103  042105  042040
031224  030510  020061  046530
031232  052111  051124  044440
031240  052116  051105  052522
031246  052120     000

                                        ;INFORMATION FOR MESSAGE 13

031251     103  040510  020122  EM13:   .ASCIZ  'CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT'
031256  053101  044501  040514
031264  046102  020105  040506
031272  046111  042105  052040
031300  020117  042507  042516
031306  040522  042524  051040
031314  053103  020122  047111
031322  042524  051122  050125
031330  000124

                                        ;INFORMATION FOR MESSAGE 14

031332  051124  047101  046523  EM14:   .ASCIZ  'TRANSMITTER NPR LOGIC ERROR - LINE #  '
031340  052111  042524  020122
031346  050116  020122  047514
031354  044507  020103  051105
031362  047522  020122  020055
031370  044514  042516  021440
031376  020040     000

                                        ;INFORMATION FOR MESSAGE 15

031401     130  044515  052124  EM15:   .ASCIZ  'XMITTR FAILED TO INTERRUPT - LINE #  '
031406  020122  040506  046111
031414  042105  052040  020117
031422  047111  042524  051122
031430  050125  020124  020055
031436  044514  042516  021440
031444  020040     000

                                        ;INFORMATION FOR MESSAGE 16

031447     122  053103  020122  EM16:   .ASCIZ  'RCVR FAILED TO INTERRUPT'
031454  040506  046111  042105
031462  052040  020117  047111
031470  042524  051122  050125
031476  000124

                                        ;INFORMATION FOR MESSAGE 17

031500  051124  047101  046523  EM17:   .ASCIZ  'TRANSMITTER TIMING ERROR - LINE #  '
031506  052111  042524  020122
031514  044524  044515  043516
031522  042440  051122  051117
```

```
031530  026440  046040  047111
031536  020105  020043  000040
031544  024040  041520  020051  DH6:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  SPEED   TIMEB   TIMEC'
031552  020040  024040  051520
031560  020051  020040  024040
031566  050123  020051  020040
031574  052040  051505  020124
031602  020040  042504  040526
031610  051104  020040  050123
031616  042505  020104  020040
031624  044524  042515  020102
031632  020040  044524  042515
031640  000103

                                ;INFORMATION FOR MESSAGE 20

031642  042522  042503  053111  EM20:   .ASCIZ  'RECEIVER TIMING ERROR - LINE #  '
031650  051105  052040  046511
031656  047111  020107  051105
031664  047522  020122  020055
031672  044514  042516  021440
031700  020040     000

                                ;INFORMATION FOR MESSAGE 21

031703     122  053103  020122  EM21:   .ASCIZ  'RCVR FAILED TO INTERRUPT - LINE #  '
031710  040506  046111  042105
031716  052040  020117  047111
031724  042524  051122  050125
031732  020124  020055  044514
031740  042516  021440  020040
031746     000

                                ;INFORMATION FOR MESSAGE 22

031747     103  040510  020122  EM22:   .ASCIZ  'CHAR AVAIL FAILED TO SET ON TIME - LINE #  '
031754  053101  044501  020114
031762  040506  046111  042105
031770  052040  020117  042523
031776  020124  047117  052040
032004  046511  020105  020055
032012  044514  042516  021440
032020  020040     000

                                ;INFORMATION FOR MESSAGE 23

032023     102  051501  041511  EM23:   .ASCIZ  'BASIC DATA TEST ERROR - LINE #  '
032030  042040  052101  020101
032036  042524  052123  042440
032044  051122  051117  026440
032052  046040  047111  020105
032060  020043  000040
032064  024040  041520  020051  DH7:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG   WAS     S/B'
032072  020040  024040  051520
032100  020051  020040  024040
032106  050123  020051  020040
```

```
032114  052040  051505  020124
032122  020040  042504  040526
032130  051104  020040  044103
032136  046122  043516  020040
032144  053440  051501  020040
032152  020040  051440  041057
032160     000

                                ;INFORMATION FOR MESSAGE 24

032161     101  052125  020117  EM24:   .ASCIZ  'AUTO ECHO TEST ERROR - LINE #  '
032166  041505  047510  052040
032174  051505  020124  051105
032202  047522  020122  020055
032210  044514  042516  021440
032216  020040     000

                                ;INFORMATION FOR MESSAGE 25

032221     102  042522  045501  EM25:   .ASCIZ  'BREAK BIT TEST ERROR - LINE #  '
032226  041040  052111  052040
032234  051505  020124  051105
032242  047522  020122  020055
032250  044514  042516  021440
032256  020040     000

                                ;INFORMATION FOR MESSAGE 26

032261     110  046101  026506  EM26:   .ASCIZ  'HALF-DUPLEX TEST ERROR - LINE #  '
032266  052504  046120  054105
032274  052040  051505  020124
032302  051105  047522  020122
032310  020055  044514  042516
032316  021440  020040     000

                                ;INFORMATION FOR MESSAGE 27

032323     125  042516  050130  EM27:   .ASCIZ  'UNEXPECTED BUS ERROR TRAP'
032330  041505  042524  020104
032336  052502  020123  051105
032344  047522  020122  051124
032352  050101     000
032355     040  050050  024503  DH3:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   TRPPC   TRPPS '
032362  020040  020040  050050
032370  024523  020040  020040
032376  051450  024520  020040
032404  020040  042524  052123
032412  020040  052040  050122
032420  041520  020040  052040
032426  050122  051520  000040
                                .EVEN
032434  001116  001202  001176  DT3:    .WORD   $ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,0
032442  001162  001164  001166
032450  000000

                                ;INFORMATION    FOR MESSAGE 30
```

```
032452  047125  054105  042520  EM30:   .ASCIZ  'UNEXPECTED RSVD INSTR TRAP'
032460  052103  042105  051040
032466  053123  020104  047111
032474  052123  020122  051124
032502  050101     000

                                ;INFORMATION FOR MESSAGE 31

032505     101  052125  020117  EM31:   .ASCIZ  'AUTO ECHO DATA COMPARE ERROR - LINE #  '
032512  041505  047510  042040
032520  052101  020101  047503
032526  050115  051101  020105
032534  051105  047522  020122
032542  020055  044514  042516
032550  021440  020040     000
032555     040  050050  024503  DH4:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   WASADR  SBADR    WAS     S/B'
032562  020040  020040  050050
032570  024523  020040  020040
032576  051450  024520  020040
032604  020040  042524  052123
032612  020040  053440  051501
032620  042101  020122  051440
032626  040502  051104  020040
032634  020040  040527  020123
032642  020040  020040  027523
032650  000102

                                ;INFORMATION FOR MESSAGE 32

032652  052501  047524  042440  EM32:   .ASCIZ  'AUTO ECHO TEST TIMEOUT - LINE #  '
032660  044103  020117  042524
032666  052123  052040  046511
032674  047505  052125  026440
032702  046040  047111  020105
032710  020043  000040
032714  024040  041520  020051  DH5:    .ASCIZ  ' (PC)   (LPRG)   TEST'
032722  020040  046050  051120
032730  024507  020040  052040
032736  051505  000124
                                .EVEN
032742  001116  001202  001206  DT4:    .WORD   $ERRPC,$TMP0,$TMP2,0
032750  000000

                                ;INFORMATION FOR MESSAGE 33

032752  040520  044522  054524  EM33:   .ASCIZ  'PARITY LOGIC TEST ERROR - LINE #  '
032760  046040  043517  041511
032766  052040  051505  020124
032774  051105  047522  020122
033002  020055  044514  042516
033010  021440  020040     000

                                ;INFORMATION FOR MESSAGE 34

033015     115  046125  044524  EM34:   .ASCIZ  'MULTI-LINE PARITY DATA TEST ERROR - LINE #   - SUBTEST #  '
```

```
033022  046055  047111  020105
033030  040520  044522  054524
033036  042040  052101  020101
033044  042524  052123  042440
033052  051122  051117  026440
033060  046040  047111  020105
033066  020043  020040  020055
033074  052523  052102  051505
033102  020124  020043  000040

                                ;INFORMATION FOR MESSAGE 35

033110  052515  052114  020511  EM35:   .ASCIZ  'MULTI-LINE PARITY DATA TEST TIMEOUT'
033116  044514  042516  050040
033124  051101  052111  020131
033132  040504  040524  052040
033140  051505  020124  044524
033146  042515  052517  000124
033154  024040  041520  020051  DH14:   .ASCIZ  ' (PC)   (LPRG)  ACTLIN'
033162  020040  046050  051120
033170  024507  020040  041501
033176  046124  047111     000
        033204                  .EVEN
033204  001116  001202  001210  DT6:    .WORD   $ERRPC,$TMP0,$TMP3,0
033212  000000

                                ;INFORMATION FOR MESSAGE 36

033214  044103  051101  040440  EM36:   .ASCIZ  'CHAR AVAILABLE TIMEOUT'
033222  040526  046111  041101
033230  042514  052040  046511
033236  047505  052125     000

                                ;INFORMATION FOR MESSAGE 37

033243     104  052101  020101  EM37:   .ASCIZ  'DATA COMPARE ERROR - LINE #  '
033250  047503  050115  051101
033256  020105  051105  047522
033264  020122  020055  044514
033272  042516  021440  020040
033300     000

                                ;INFORMATION FOR MESSAGE 40

033301     102  043125  042506  EM40:   .ASCIZ  'BUFFER ACTIVE REG ERROR - LINE #  '
033306  020122  041501  044524
033314  042526  051040  043505
033322  042440  051122  051117
033330  026440  046040  047111
033336  020105  020043  000040

                                ;INFORMATION FOR MESSAGE 41

033344  041522  051126  043040  EM41:   .ASCIZ  'RCVR FALSE INTERRUPT'
033352  046101  042523  044440
033360  052116  051105  052522
```

```
033366  052120     000

                                ;INFORMATION FOR MESSAGE 42

033371     123  046111  020117  EM42:   .ASCIZ  'SILO OVERFLOW ERROR'
033376  053117  051105  046106
033404  053517  042440  051122
033412  051117     000

                                ;INFORMATION FOR MESSAGE 43

033415     123  046111  020117  EM43:   .ASCIZ  'SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT'
033422  053117  051105  046106
033430  053517  043040  044501
033436  042514  020104  047524
033444  043440  047105  051105
033452  052101  020105  041522
033460  051126  044440  052116
033466  051105  052522  052120
033474     000

                                ;INFORMATION FOR MESSAGE 44

033475     116  047117  042440  EM44:   .ASCIZ  'NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT'
033502  020130  042515  047515
033510  054522  043040  044501
033516  042514  020104  047524
033524  043440  047105  051105
033532  052101  020105  046530
033540  052111  051124  044440
033546  052116  051105  052522
033554  052120     000

                                ;INFORMATION FOR MESSAGE 45

033557     130  044515  020124  EM45:   .ASCIZ  'XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT'
033564  047504  042516  043040
033572  044501  042514  020104
033600  047524  043440  047105
033606  051105  052101  020105
033614  046530  052111  051124
033622  044440  052116  051105
033630  052522  052120     000

                                ;INFORMATION FOR MESSAGE 46

033635     103  051125  042522  EM46:   .ASCIZ  'CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE #  '
033642  05[illegible]16  040440  042104
033650  04[illegible]22  051523  046440
033656  046505  051117  020131
033664  040520  052124  051105
033672  051516  052040  051505
033700  020124  051105  047522
033706  020122  020055  044514
033714  042516  021440  020040
033722     000
```

```
033723     040  050050  024503  DH10:   .ASCIZ  ' (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS     S/B'
033730  020040  046040  047111
033736  053505  020122  050040
033744  052101  051124  020116
033752  020040  042524  052123
033760  020040  042040  053105
033766  042101  020122  051040
033774  043505  042101  020122
034002  020040  040527  020123
034010  020040  020040  027523
034016  000102
                                .EVEN
034020  001116  001202  001204  DT5:    .WORD   $ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4,0
034026  001162  001164  001166
034034  001170  001172  000000

                                ;INFORMATION FOR MESSAGE 47

034042  054502  042524  041440  EM47:   .ASCIZ  'BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE #  '
034050  052517  052116  046440
034056  046505  051117  020131
034064  040520  052124  051105
034072  051516  052040  051505
034100  020124  051105  047522
034106  020122  020055  044514
034114  042516  021440  020040
034122     000

                                ;INFORMATION FOR MESSAGE 50

034123     124  051505  020124  EM50:   .ASCIZ  'TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  '
034130  044524  042515  052517
034136  020124  040527  052111
034144  047111  020107  047506
034152  020122  046530  052111
034160  042040  047117  020105
034166  020055  044514  042516
034174  021440  020040     000

                                ;INFORMATION FOR MESSAGE 51

034201     116  051120  046040  EM51:   .ASCIZ  'NPR LOGIC TEST 2 ERROR'
034206  043517  041511  052040
034214  051505  020124  020062
034222  051105  047522  000122
034230  024040  041520  020051  DH11:   .ASCIZ  ' (PC)   LINACT  LINCHK   TEST   DEVADR  REGADR   WAS     S/B'
034236  020040  044514  040516
034244  052103  020040  044514
034252  041516  045510  020040
034260  052040  051505  020124
034266  020040  042504  040526
034274  051104  020040  042522
034302  040507  051104  020040
034310  053440  051501  020040
034316  020040  051440  041057
034324     000
```

```
                                        ;INFORMATION FOR MESSAGE 52

034325     102  051501  041511  EM52:   .ASCIZ  'BASIC DATA COMPARE ERROR'
034332  042040  052101  020101
034340  047503  050115  051101
034346  020105  051105  047522
034354  000122

                                        ;INFORMATION FOR MESSAGE 53

034356  024040  041520  020051  DH12:   .ASCIZ  ' (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS     S/B'
034364  020040  050123  042505
034372  020104  020040  024040
034400  050123  020051  020040
034406  052040  051505  020124
034414  020040  042504  040526
034422  051104  020040  042522
034430  040507  051104  020040
034436  053440  051501  020040
034444  020040  051440  041057
034452     000

                                        ;INFORMATION FOR MESSAGE 55

034453     040  050050  024503  DH13:   .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG  SCRWAS  SCRS/B'
034460  020040  020040  050050
034466  024523  020040  020040
034474  051450  024520  020040
034502  020040  042524  052123
034510  020040  042040  053105
034516  042101  020122  041440
034524  051110  047114  020107
034532  051440  051103  040527
034540  020123  051440  051103
034546  027523  000102

                                        ;INFORMATIOPN FOR MESSAGE 56

034552  053117  051105  052522  EM56:   .ASCIZ  'OVERRUN BIT FAILED TO SET - LINE #  '
034560  020116  044502  020124
034566  040506  046111  042105
034574  052040  020117  042523
034602  020124  020055  044514
034610  042516  021440  020040
034616     000

                                        ;INFORMATION FOR MESSAGE 57

034617     123  047524  040522  EM57:   .ASCIZ  'STORAGE OVERFLOW BIT FAILED - LINE #  '
034624  042507  047440  042526
034632  043122  047514  020127
034640  044502  020124  040506
034646  046111  042105  026440
034654  046040  047111  020105
034662  020043  000040
```

```
                                        ;INFORMATION FOR MESSAGE 60

034666  020040  024040  041520  DH15:   .ASCIZ  /   (PC)  DEVADR    LINE   /
034674  020051  042040  053105
034702  042101  020122  020040
034710  046040  047111  020105
034716  020040     000
        034722                  .EVEN
                                ;MISCELLANEOUS MESSAGES

034722  005015  040515  047111  TITLE:  .ASCIZ  <15><12>'MAINDEC-11-DZDHM-C  DH11 DIAGNOSTIC'<15><12>
034730  042504  026503  030461
034736  042055  042132  046510
034744  041455  020040  044104
034752  030461  042040  040511
034760  047107  051517  044524
034766  006503  000012
034772  005015  042524  052123  TITLE2: .ASCIZ  <15><12>'TESTING DH11 #  '<15><12>
035000  047111  020107  044104
035006  030461  021440  020040
035014  005015     000
035017     015  052012  050131  INMSG1: .ASCIZ  <15><12>'TYPE SCR ADDRESS FOR FIRST DH11'<15><12>
035024  020105  041523  020122
035032  042101  051104  051505
035040  020123  047506  020122
035046  044506  051522  020124
035054  044104  030461  005015
035062     000
035063     015  052012  050131  INMSG2: .ASCIZ  <15><12>'TYPE VECTOR ADDRESS FOR FIRST DH11'<15><12>
035070  020105  042126  052103
035076  051117  040440  042104
035104  042522  051523  043040
035112  051117  043040  051111
035120  052123  042040  030510
035126  006461  000012
035132  005015  054524  042520  INMSG3: .ASCIZ  <15><12>'TYPE DH11 DEVICE SELECTION PARAMETER'<15><12>
035140  042040  030510  020061
035146  042504  044526  042503
035154  051440  046105  041505
035162  044524  047117  050040
035170  051101  046501  052105
035176  051105  005015     000
035203     015  044412  053116  INMSG4: .ASCIZ  <15><12>'INVALID DH11 SCR ADDRESS - TRY AGAIN'<15><12>
035210  046101  042111  042040
035216  030510  020061  041523
035224  020122  042101  051104
035232  051505  020123  020055
035240  051124  020131  043501
035246  044501  006516  000012
035254  005015  047111  040526  INMSG5: .ASCIZ  <15><12>'INVALID DH11 VECTOR ADDRESS - TRY AGAIN'<15><12>
035262  044514  020104  044104
035270  030461  053040  041505
035276  047524  020122  042101
035304  051104  051505  020123
035312  020055  051124  020131
035320  043501  044501  006516
```

```
10035  035326  000012
10036  035330  005015  054524  042520  INMSG6: .ASCIZ  <15><12>'TYPE LINE SELECTION PARAMETER'<15><12>
10037  035336  046040  047111  020105
10038  035344  042523  042514  052103
10039  035352  047511  020116  040520
10040  035360  040522  042515  042524
10041  035366  006522  000012
10042  035372  005015  042504  051120  INMSG7: .ASCIZ  <15><12>'DEPRESS "CONTINUE" TO START TESTING'<15><12>
10043  035400  051505  020123  041442
10044  035406  047117  044524  052516
10045  035414  021105  052040  020117
10046  035422  052123  051101  020124
10047  035430  042524  052123  047111
10048  035436  006507  000012
10049
10050  035442  005015  054524  042520  VCWC:   .ASCIZ  <15><12>'TYPE NO. OF ADDRESSES (OCTAL) BETWEEN VECTORS (10 OR 20)'<15><1
10051  035450  047040  027117  047440
10052  035456  020106  042101  051104
10053  035464  051505  042523  020123
10054  035472  047450  052103  046101
10055  035500  020051  042502  053524
10056  035506  042505  020116  042526
10057  035514  052103  051117  020123
10058  035522  030450  020060  051117
10059  035530  031040  024460  005015
10060  035536     000
10061  035537     116  020117  044104  MSG1:   .ASCIZ  /NO DH11'S WERE FOUND./<15><12>
10062  035544  030461  051447  053440
10063  035552  051105  020105  047506
10064  035560  047125  027104  005015
10065  035566     000
10066  035567     116  020117  044104  MSG2:   .ASCIZ  'NO DH RECEIVER INTERRUPT OCCURRED.'<15><12>
10067  035574  051040  041505  044505
10068  035602  042526  020122  047111
10069  035610  042524  051122  050125
10070  035616  020124  041517  052503
10071  035624  051122  042105  006456
10072  035632  000012
10073  035634  047516  046440  042117  MSG3:   .ASCIZ  'NO MODEM CONTROL INTERRUPT OCCURRED.'<15><12>
10074  035642  046505  041440  047117
10075  035650  051124  046117  044440
10076  035656  052116  051105  052522
10077  035664  052120  047440  041503
10078  035672  051125  042522  027104
10079  035700  005015     000
10080  035703     040  020040     000  SPACE:  .ASCIZ  '   '
10081  035707     015  042012  030510  DEVMAP: .ASCII  <15><12>'DH11, MODEM CONTROL DEVICE MAP:'<15><12>
10082  035714  026061  046440  042117
10083  035722  046505  041440  047117
10084  035730  051124  046117  042040
10085  035736  053105  041511  020105
10086  035744  040515  035120  005015
10087  035752  005015  044104  030461          .ASCII  <15><12>'DH11   DH11   MODEM CONTROL   MODEM CONTROL'
10088  035760  020040  042040  030510
10089  035766  020061  020040  047515
10090  035774  042504  020115  047503
```

```
036002  052116  047522  020114
036010  020040  047515  042504
036016  020115  047503  052116
036024  047522     114
036027     015  040412  051104          .ASCIZ  <15><12>'ADRS   VECT    ADRS      VECT'<15><12><15><12>
036034  020123  020040  042526
036042  052103  020040  020040
036050  042101  051522  020040
036056  020040  020040  042526
036064  052103  005015  005015
036072     000
036073     015  046412  042117  MSG4:   .ASCIZ  <15><12>'MODEM CONTROL ERROR, RUN DZDHK DIAGNOSTIC'<15><12>
036100  046505  041440  047117
036106  051124  046117  042440
036114  051122  051117  020054
036122  052522  020116  055104
036130  044104  020113  044504
036136  043501  047516  052123
036144  041511  005015     000
036151     015  047012  020117  MSG5:   .ASCIZ  <15><12>/NO MODEM CONTROL FOUND/<15><12>
036156  047515  042504  020115
036164  047503  052116  047522
036172  020114  047506  047125
036200  006504  000012
                                  .EVEN
                                  ;SIXTEEN CHAR COUNTERS USED BY THE AUTO ECHO TEST #3

036204  000020                    RCNT:   .BLKW   16.

                                  ;256. WORD RECEIVER INPUT BUFFER

036244  000400                    RBUF:   .BLKW   256.


                                  ;256(10) BYTE TRANSMITTER OUTPUT DATA BUFFER

                                  .EVEN
037244  000400                    TBUF:   .BLKB   256.

        000001                    .END
```

```
ABASE = 000000
ACDW1 = 000000
ACDW2 = 000000
ACPUOP= 000000
ADDW0 = 000000
ADDW1 = 000000
ADDW10= 000000
ADDW11= 000000
ADDW12= 000000
ADDW13= 000000
ADDW14= 000000
ADDW15= 000000
ADDW2 = 000000
ADDW3 = 000000
ADDW4 = 000000
ADDW5 = 000000
ADDW6 = 000000
ADDW7 = 000000
ADDW8 = 000000
ADDW9 = 000000
ADEVCT= 000000
ADEVM = 000000
ADPTR   030250
ADRVEC  030230
AENV  = 000000
AENVM = 000000
AETAB   027342
AETABO  027402
AFATAL= 000000
AMADR1= 000000
AMADR2= 000000
AMADR3= 000000
AMADR4= 000000
AMAMS1= 000000
AMAMS2= 000000
AMAMS3= 000000
AMAMS4= 000000
AMSGAD= 000000
AMSGLG= 000000
AMSGTY= 000000
AMTYP1= 000000
AMTYP2= 000000
AMTYP3= 000000
AMTYP4= 000000
APASS = 000000
APRIOR= 000000
APTCSU= 000040
APTENV= 000001
APTSIZ= 000200
APTSPO= 000100
ASWREG= 000000
ATESTN= 000000
AUNIT = 000000
AUSWR = 000000
AUTOSZ  025130
AVECT1= 000000
AVECT2= 000000
BAR   = 000012
BCR   = 000010
BIN     002156
BINA    002160
BIT0  = 000001
BIT00 = 000001
BIT01 = 000002
BIT02 = 000004
BIT03 = 000010
BIT04 = 000020
BIT05 = 000040
BIT06 = 000100
BIT07 = 000200
BIT08 = 000400
BIT09 = 001000
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BKR   = 000014
BPTVEC= 000014
BRKTAB  027544
BRLVL   027724
BRPTR   030254
BUSER   026620
CAR   = 000006
CHKADR  026372
CHKVCT  026506
CHPS1   027130
CHPS2   027144
CKSWR = 104407
CLCABC  024706
CLRMUX  002000
CO    = 000100
CR    = 000015
CRLF  = 000200
CS    = 000040
DDISP = 177570
DEVMAP  035707
DHADR   027224
DHADRS  027764
DHADTB  027622
DHMCSR  030232
DHMLSR  030234
DHNUM   030242
DHRLVL  030240
DHSEL   027232
DHTLVL  030241
DHVCT   027226
DHVCTB  027662
DHVEC   030026
DH1     030351
DH10    033723
DH11    034230
DH12    034356
DH13    034453
DH14    033154
DH15    034666
DH2     030504
DH3     032355
DH4     032555
DH5     032714
DH6     031544
DH7     032064
DISPLA  001142
DISPRE  000174
DMADRS  030066
DMVEC   030130
DOIT11  017620
DOIT15  020016
DOIT16  020214
DOIT17  020404
DSWR  = 177570
DT1     030430
DT2     030602
DT3     032434
DT4     032742
DT5     034020
DT6     033204
EMTVEC= 000030
EM1     030302
EM10    031102
EM11    031151
EM12    031210
EM13    031251
EM14    031332
EM15    031401
EM16    031447
EM17    031500
EM2     030446
EM20    031642
EM21    031703
EM22    031747
EM23    032023
EM24    032161
EM25    032221
EM26    032261
EM27    032323
EM3     030624
EM30    032452
EM31    032505
EM32    032652
EM33    032752
EM34    033015
EM35    033110
EM36    033214
EM37    033243
EM4     030702
EM40    033301
EM41    033344
EM42    033371
EM43    033415
EM44    033475
EM45    033557
EM46    033635
EM47    034042
EM5     030740
EM50    034123
EM51    034201
EM52    034325
EM56    034552
EM57    034617
EM6     030775
EM7     031030
ENDA    020572
ERRVEC= 000004
EXPAR   026366
GTSWR = 104406
HT    = 000011
INMSG1  035017
INMSG2  035063
INMSG3  035132
INMSG4  035203
INMSG5  035254
INMSG6  035330
INMSG7  035372
INPAR   026160
INPARA  026100
INPARC  026136
INPARX  026146
INPAR1  026214
INPAR3  026250
INPAR4  026306
INTMSK  027620
IOTVEC= 000020
LDBCR   024750
LDTBF1  024320
LF    = 000012
LINACT  027502
LINBIT  027442
LINE    030244
LINEA   030246
LINENA= 000001
LINMSK  027236
LINSEL  027234
LMSK1   027240
LPR   = 000004
MSG1    035537
MSG2    035567
MSG3    035634
MSG4    036073
MSG5    036151
MSTCLR  027242
MULPTB  027504
MUX11   017626
MUX11A  017656
MUX11B  017706
MUX11C  017734
MUX11D  017746
MUX11E  017762
MUX15   020030
MUX15A  020060
MUX15B  020102
MUX15C  020130
MUX15D  020136
MUX15E  020154
MUX16   020226
MUX16A  020250
MUX16B  020272
MUX16C  020320
MUX16D  020326
MUX16E  020344
MUX17   020416
MUX17A  020440
MUX17B  020462
MUX17C  020510
MUX17D  020516
MUX17E  020534
NRC   = 000002
PATRNA  027252
PATRNB  027262
PIRQ  = 177772
PIRQVE= 000240
PRTYTB  027302
PR0   = 000000
PR1   = 000040
PR2   = 000100
```

```
PR3     = 000140
PR4     = 000200
PR5     = 000240
PR6     = 000300
PR7     = 000340
PS      = 177776
PSW     = 177776
PWRVEC= 000024
RBUF    036244
RCNT    036204
RDCHR = 104410
RDLIN = 104411
RDOCT = 104412
RESERR  026674
RESTRP  026750
RESTRT  002672
RESTI   002730
RESVEC= 000010
RGMSK1  027604
RGMSK2  027606
RGMSK3  027610
RGMSK4  027612
RGMSK5  027614
RGMSK6  027616
RING    = 000200
RS      = 000004
RSTRTA  002740
R6      =%000006
R7      =%000007
SAPS    027160
SECRX = 000020
SECTX = 000010
SELINE  024534
SELMSK  027230
SETALL  027020
SPACE   035703
SSR     = 000016
STACK = 001100
START1  002474
START2  002634
STEP    = 000400
STKLMT= 177774
SUER1   024342
SUER2   024402
SUER2A  024406
SUER3   024452
SUER4   024470
SUNUM   024626
SUPPAR  025010
SWR     001140
SWREG   000176
SW0     = 000001
SW00    = 000001
SW01    = 000002
SW02    = 000004
SW03    = 000010
SW04    = 000020
SW05    = 000040
SW06    = 000100
SW07    = 000200
SW08    = 000400
SW09    = 001000
SW1     = 000002
SW10    = 002000
SW11    = 004000
SW12    = 010000
SW13    = 020000
SW14    = 040000
SW15    = 100000
SW2     = 000004
SW3     = 000010
SW4     = 000020
SW5     = 000040
SW6     = 000100
SW7     = 000200
SW8     = 000400
SW9     = 001000
TBITVE= 000014
TBUF    037244
TDATA1  030256
TDATA2  030260
TIMEA   030272
TIMEB   030274
TIMEC   030276
TIMEIT  026776
TITFLG  030270
TITLE   034722
TITLE2  034772
TKVEC = 000060
TNULL   030300
TPVEC = 000064
TRAPVE= 000034
TRMRDY= 000002
TRTVEC= 000014
TST1    003010
TST10   004010
TST11   004136
TST12   004266
TST13   004416
TST14   004556
TST15   004746
TST16   005136
TST17   005262
TST2    003100
TST20   005406
TST21   005614
TST22   006022
TST23   006222
TST24   006422
TST25   006662
TST26   007046
TST27   007226
TST3    003200
TST30   007406
TST31   007566
TST32   007746
TST33   010400
TST34   010726
TST35   011160
TST36   011302
TST37   011456
TST4    003310
TST40   011672
TST41   012200
TST42   012506
TST43   013016
TST44   013256
TST45   013730
TST46   014162
TST47   014726
TST5    003372
TST50   015300
TST51   015750
TST52   016304
TST53   017044
TST54   017260
TST55   017522
TST56   020014
TST57   020212
TST6    003560
TST60   020402
TST7    003674
TYPDS = 104405
TYPE  = 104401
TYPOC = 104402
TYPON = 104404
TYPOS = 104403
VCFLG   027722
VCPTR   030252
VCWC    035442
$APTHD  000214
$ATYC   022570
$ATY1   022544
$ATY3   022552
$ATY4   022562
$AUTOB  001134
$BASE   001306
$BDADR  001122
$BDDAT  001126
$CDW1   001312
$CDW2   001314
$CHARC  022540
$CKSWR  023012
$CMTAG  001100
$CM1  = 000010
$CM2  = 000020
$CM3  = 000010
$CM4  = 000010
$CNTLG  023665
$CNTLU  023660
$CPUOP  001260
$CRLF   001227
$DBLK   022252
$DDW0   001316
$DDW1   001320
$DDW10  001342
$DDW11  001344
$DDW12  001346
$DDW13  001350
$DDW14  001352
$DDW15  001354
$DDW2   001322
$DDW3   001324
$DDW4   001326
$DDW5   001330
$DDW6   001332
$DDW7   001334
$DDW8   001336
$DDW9   001340
$DEVCT  001242
$DEVM   001310
$DHSEL  030236
$DOAGN  020754
$DTBL   022242
$ENDAD  020744
$ENDCT  020712
$ENDMG  020763
$ENULL  020760
$ENV    001252
$ENVM   001253
$EOP    020656
$EOPCT  020704
$ERFLG  001103
$ERMAX  001115
$ERROR  021270
$ERRPC  001116
$ERRTB  001356
$ERRTY  021454
$ERTTL  001112
$ESCAP  001224
$ETABL  001252
$ETEND  001356
$FATAL  001234
$FFLG   023010
$FILLC  001156
$FILLS  001155
$GDADR  001120
$GDDAT  001124
$GET42  020734
$GTSWR  023062
$HD   = 000001
$HIBTS  000214
$HIOCT  024052
$ICNT   001104
$ILLUP  024302
$INTAG  001135
$ITEMB  001114
$LF     001230
$LFLG   023007
$LNSEL  030170
$LPADR  001106
$LPERR  001110
$MADR1  001264
$MADR2  001270
$MADR3  001274
$MADR4  001300
$MAIL   001232
$MAMS1  001262
$MAMS2  001266
$MAMS3  001272
$MAMS4  001276
$MBADR  000216
$MFLG   023006
$MNEW   023703
$MSGAD  001246
$MSGLG  001250
$MSGTY  001232
$MSWR   023672
$MTYP1  001263
$MTYP2  001267
$MTYP3  001273
$MTYP4  001277
$MXCNT  021266
$NULL   001154
$NWTST= 000001
$OCNT   022032
$OMODE  022034
$OVER   021252
$PASS   001240
$PASTM  000222
$POWER  024310
$PWRAD  024276
$PWRDN  024136
$PWRMG  024272
$PWRUP  024210
```

```
$QUES   001226          $REG7   001200          $TKB    001146          $TPS    001150          $TYPON  021650
$RDCHR  023274          $RTNAD  020756          $TKS    001144          $TRAP   024054          $TYPOS  021610
$RDLIN  023414          $SAVR6  024306          $TMP0   001202          $TRAP2  024076          $UNIT   001244
$RDOCT  023714          $SCOPE  021000          $TMP1   001204          $TRP  = 000013          $UNITM  000224
$RDSZ = 000010          $SETUP= 000137          $TMP2   001206          $TRPAD  024110          $USWR   001256
$REGAD  001160          $STUP = 177777          $TMP3   001210          $TSTM   000220          $VECT1  001302
$REG0   001162          $SVLAD  021216          $TMP4   001212          $TSTNM  001102          $VECT2  001304
$REG1   001164          $SVPC = 000214          $TMP5   001214          $TTYIN  023650          $XTSTR  021022
$REG2   001166          $SWR  = 165400          $TMP6   001216          $TYPDS  022036          $$GET4= 000000
$REG3   001170          $SWREG  001254          $TMP7   001220          $TYPE   022262          $OFILL  022033
$REG4   001172          $SWRMK= 000000          $TN   = 000061          $TYPEC  022474          .     = 037644
$REG5   001174          $TESTN  001236          $TPB    001152          $TYPEX  022542          .$X   = 000214
$REG6   001176          $TIMES  001222          $TPFLG  001157          $TYPOC  021634

. ABS.  037644     000


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 ,DZDHMC/SOL/NL:TOC←DHMMAC,DZDHMC
 RUN-TIME: 30 26 .6 SECONDS
 RUN-TIME RATIO: 507/57=8.7
 CORE USED:  41K  (81 PAGES)
```