DH11

DH11 DIAGNOSTIC

MD-11-DZDHM-C

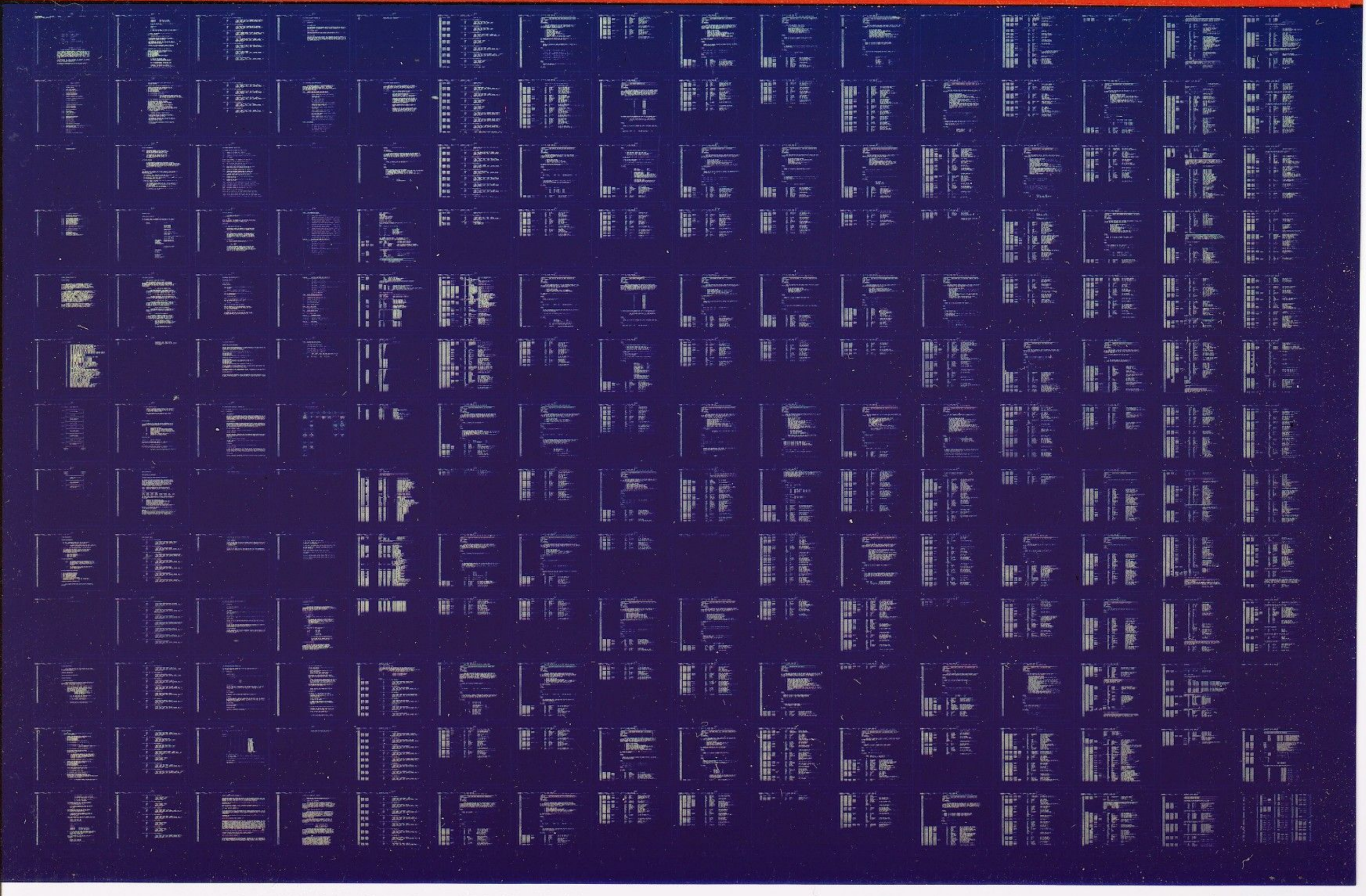
EP-DZDHM-C-DL-B

FICHE 1 OF 2

APR 197

digital

MADE IN U



DH11

DH11 DIAGNOSTIC

MD-11-DZDHM-C

EP-DZDHM-C-DL-B APR 1977

FICHE 2 OF 2 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

TABLE OF CONTENTS

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

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.

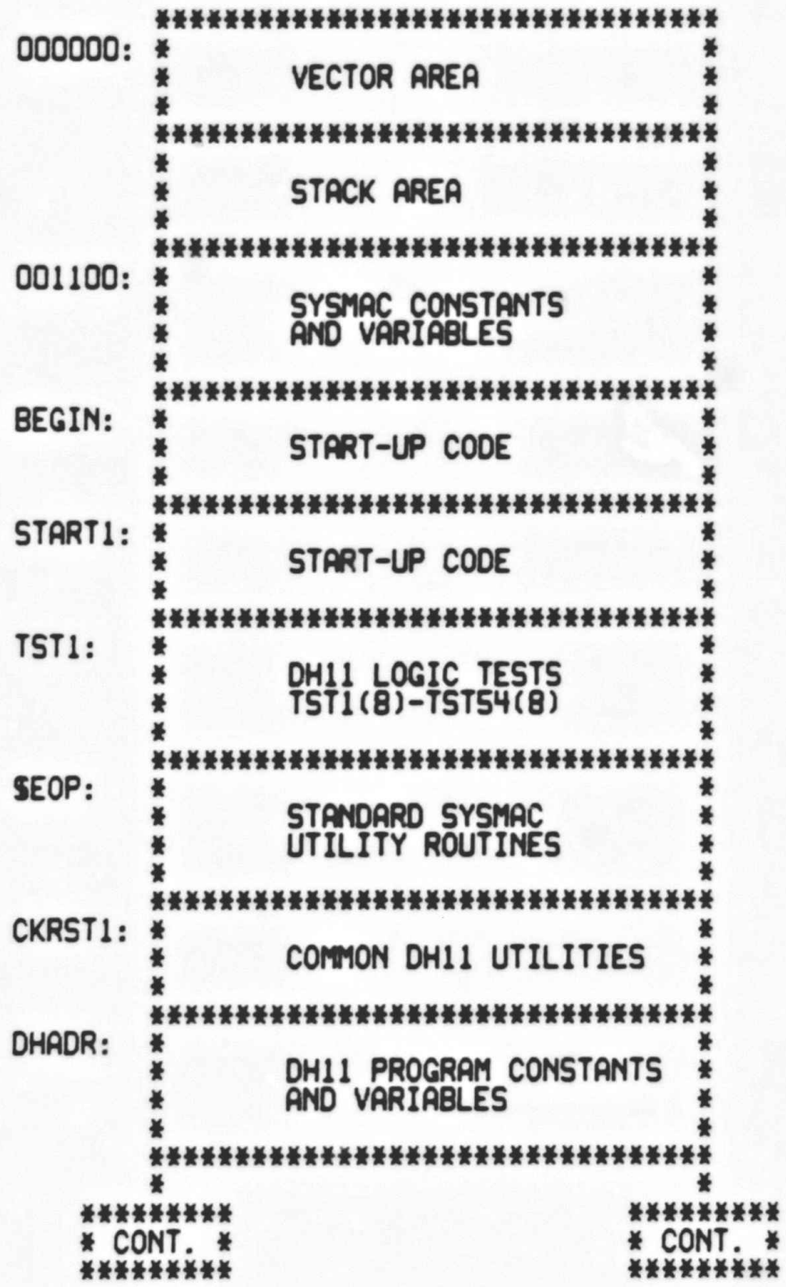
THE PROGRAM IS CONFIGURABLE BY THE AUTOSIZER OR BY CONSOLE DIALOGUE TO ENABLE IT TO AUTOMATICALLY TEST AND VERIFY ALL 16. LINES ON UP TO 16. CONTIGUOUS DH11'S (WITH NON-CONTIGUOUS/CONTIGUOUS VECTOR ASSIGNMENTS). INDIVIDUAL UNITS AND INDIVIDUAL LINES WITHIN A UNIT MAY BE SELECTED OR DESELECTED TO FACILITATE 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



```
277                                  *********                    *********
278                                  * CONT. *                    * CONT. *
279                                  *********                    *********
280                                      *                            *
281                                      ********************************
282                              EM1:    *                              *
283                                      *      SYSMAC ERROR MESSAGE    *
284                                      *      BUFFERS                 *
285                                      *                              *
286                                      ********************************
287                              TITLE:  *                              *
288                                      *      DH11 MISCELLANEOUS      *
289                                      *      MESSAGE BUFFERS         *
290                                      *                              *
291                                      ********************************
292                              RBUF:   *                              *
293                                      *      TRANSMIT AND RECEIVE    *
294                                      *      DATA BUFFERS            *
295                                      *                              *
296                                      ********************************
297
```

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. DH11-0 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:

```
354
355                                          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 MANUALLY 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 INTRODUCTORY 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 SR=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 DESCRIBED 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,$REG6,$REGO,$REG1,$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,$REG6,$REGO,$REG1,$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,$REG6,$REGO,$REG1,$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,$REG6,$REGO,$REG1,$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,$REG6,$REGO,$REG1,$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,$REG6,$REGO,$REG1,$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,$REG6,$REGO,$REG1,$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  WAS  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
```

```
1028                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 26
1029
1030                                            EM26            ;"HALF-DUPLEX TEST ERROR - LINE #  "
1031                                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
1032                                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1033                                            0               ;PRINT ALL OCTAL
1034
1035                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 27
1036
1037                                            EM27            ;"UNEXPECTED BUS ERROR TRAP"
1038                                            DH3             ; (PC)  (PS)  (SP)  TEST  TRPPC  TRPPS
1039                                            DT3             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2
1040                                            0               ;PRINT ALL OCTAL
1041
1042                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 30
1043
1044                                            EM30            ;"UNEXPECTED RSVD INSTR TRAP"
1045                                            DH3             ; (PC)  (PS)  (SP)  TEST  TRPPC  TRPPS
1046                                            DT3             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2
1047                                            0               ;PRINT ALL OCTAL
1048
1049                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 31
1050
1051                                            EM31            ;"AUTO ECHO DATA COMPARE ERROR - LINE #  "
1052                                            DH4             ; (PC)   (PS)   (SP)    TEST   WASADR  SBADR   WAS    S/
1053                                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1054                                            0               ;PRINT ALL OCTAL
1055
1056                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 32
1057
1058                                            EM32            ;"AUTO ECHO TEST TIMEOUT - LINE #  "
1059                                            DH5             ;" (PC)  (LPRG)   TEST""
1060                                            DT4             ;$ERRPC,$TMPO,$TMP2
1061                                            0               ;PRINT ALL OCTAL
1062
1063
1064                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 33
1065
1066                                            EM33            ;"PARITY LOGIC TEST ERROR - LINE #  "
1067                                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B"
1068                                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1069                                            0               ;PRINT ALL OCTAL
1070
1071                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 34
1072
1073                                            EM34            ;"MULTI-LINE PARITY DATA TEST ERROR - LINE #   - SUBTEST
1074                                            DH4             ;" (PC)  (PS)  (  )  TEST  WASADR  SBADR  WAS  S/B "
1075                                            DT2             ;$ERRPC,$TMPO,$.  6,$REG0,$REG1,$REG2,$REG3,$REG4
1076                                            0               ;PRINT ALL OCTAL
1077
```

```
1078                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 35
1079
1080                                            EM35            ;"MULTI-LINE PARITY DATA TEST TIMEOUT"
1081                                            DH14            ;" (PC)  (LPRG)  LINACT "
1082                                            DT6             ;"$ERRPC,$TMP0,$TMP3"
1083                                            0               ;PRINT ALL OCTAL
1084
1085                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 36
1086
1087                                            EM36            ;CHAR AVAILABLE TIMEOUT"
1088                                            DH5             ;" (PC)  (LPRG)   TEST"
1089                                            DT4             ;$ERRPC,$TMP0,$TMP2
1090                                            0               ;PRINT ALL OCTAL
1091
1092                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 37
1093
1094                                            EM37            ;"DATA COMPARE ERROR - LINE #  "
1095                                            DH4             ;" (PC)  (PS)  (SP)  TEST  WASADR  SBADR  WAS  S/B "
1096                                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1097                                            0               ;PRINT ALL OCTAL
1098
1099                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 40
1100
1101                                            EM40            ;"BUFFER ACTIVE REG ERROR - LINE #  "
1102                                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
1103                                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1104                                            0               ;PRINT ALL OCTAL
1105
1106                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 41
1107
1108                                            EM41            ;"RCVR FALSE INTERRUPT"
1109                                            DH5             ;" (PC)  (LPRG)   TEST"
1110                                            DT4             ;$ERRPC,$TMP0,$TMP2
1111                                            0               ;PRINT ALL OCTAL
1112
1113                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 42
1114
1115                                            EM42            ;"SILO OVERFLOW ERROR"
1116                                            DH5             ;" (PC)  (LPRG)   TEST"
1117                                            DT4             ;$ERRPC,$TMP0,$TMP2
1118                                            0               ;PRINT ALL OCTAL
1119
1120                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 43
1121
1122                                            EM43            ;"SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT"
1123                                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
1124                                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1125                                            0               ;PRINT ALL OCTAL
1126
```

```
1127                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 44
1128
1129                                            EM44            ;"NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT"
1130                                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
1131                                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1132                                            0               ;PRINT ALL OCTAL
1133
1134                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 45
1135
1136                                            EM45            ;"XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT"
1137                                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
1138                                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1139                                            0               ;PRINT ALL OCTAL
1140
1141                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 46
1142
1143                                            EM46            ;"CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE #  "
1144                                            DH10            ;" (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS
1145                                            DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4
1146                                            0               ;PRINT ALL OCTAL
1147
1148                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 47
1149
1150                                            EM47            ;"BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE #  "
1151                                            DH10            ; (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS
1152                                            DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4
1153                                            0               ;PRINT ALL OCTAL
1154
1155                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 50
1156
1157                                            EM50            ;"TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  '
1158                                            DH2             ;" (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS
1159                                            DT2             ;"$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
1160                                            0               ;PRINT ALL OCTAL
1161
1162                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 51
1163
1164                                            EM51            ;"NPR LOGIC TEST 2 ERROR"
1165                                            DH11            ;" (PC)   LINACT  LINCHK   TEST   DEVADR  REGADR   WAS
1166                                            DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4"
1167                                            0               ;PRINT ALL OCTAL
1168
1169                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 52
1170
1171                                            EM52            ;"BASIC DATA COMPARE ERROR"
1172                                            DH2             ;" (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS
1173                                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
1174                                            0               ;PRINT ALL OCTAL
1175
```

```
1176                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 53
1177
1178                                            EM50            ;"TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  "
1179                                            DH12            ;" (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS
1180                                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1181                                            0               ;PRINT ALL OCTAL
1182
1183                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 54
1184
1185                                            EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
1186                                            DH12            ;" (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS
1187                                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1188                                            0               ;PRINT ALL OCTAL
1189
1190                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 55
1191
1192                                            EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
1193                                            DH13            ;" (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG  SCRWAS
1194                                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1195                                            0               ;PRINT ALL OCTAL
1196
1197                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 56
1198
1199                                            EM56            ;"OVERRUN BIT FAILED TO SET - LINE #  "
1200                                            DH2             ; (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS
1201                                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1202                                            0               ;PRINT ALL OCTAL
1203
1204                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 57
1205
1206                                            EM57            ;"STORAGE OVERFLOW BIT FAILED - LINE #  "
1207                                            DH2             ; (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS
1208                                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
1209                                            0               ;PRINT ALL OCTAL
1210
```

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

```
1454                                    12      STORAGE INTERRUPT ENABLE
1455
1456                                            THIS BIT, WHEN SET, PERMITS THE SETTING OF BIT 14 TO GENERATE AN INTERRU
1457
```

```
1458                                    13      TRANSMITTER AND NON-EX-MEM INTERRUPT ENABLE
1459
1460                                            THIS BIT, WHEN SET, PERMITS THE SETTING OF BIT 10 OR 15 TO GENERATE AN I
1461
1462                                    14      STORAGE INTERRUPT
1463
1464                                            THIS BIT IS SET WHEN THE RECEIVER SCANNER FINDS A RECEIVER HOLDING BUFFE
1465                                            STORE THAT CHARACTER IN THE SILO, AND CANNOT DO SO BECAUSE OF A LACK OF
1466                                            AN INTERRUPT REQUEST IF BIT 12 IS SET. THIS BIT IS READ ONLY, EXCEPT IN
1467                                            IT IS READ/WRITE.
1468
1469                                    15      TRANSMITTER INTERRUPT
1470
1471                                            THIS BIT IS SET WHEN THE DH11 CONCLUDES AN NPR CYCLE THAT INCREMENTED A
1472                                            CHARACTER IN A MESSAGE BUFFER WAS LOADED INTO A UART TRANSMITTER HOLDING
1473                                            REQUEST IF BIT 13 IS SET.  THIS BIT IS READ/WRITE. (IT IS SET DURING AN
1474
```

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.

```
1564                    SPEED TABLE FOR RECEIVER AND TRANSMITTER SPEEDS:
1565
1566                                            BIT
1567
1568                    TRANSMITTER     13      12      11      10
1569
1570                    RECEIVER        9       8       7       6
1571                                    --      --      --      --
1572
1573                                    0       0       0       0       ZERO BAUD
1574                                    0       0       0       1       50 BAUDS
1575                                    0       0       1       0       75 BAUDS
1576                                    0       0       1       1       110 BAUDS
1577                                    0       1       0       0       134.5 BAUDS
1578                                    0       1       0       1       150 BAUDS
1579                                    0       1       1       0       200 BAUDS
1580                                    0       1       1       1       300 BAUDS
1581                                    1       0       0       0       600 BAUDS
1582                                    1       0       0       1       1200 BAUDS
1583                                    1       0       1       0       1800 BAUDS
1584                                    1       0       1       1       2400 BAUDS
1585                                    1       1       0       0       4800 BAUDS
1586                                    1       1       0       1       9600 BAUDS
1587                                    1       1       1       0       EXTERNAL INPUT A
1588                                    1       1       1       1       EXTERNAL INPUT B
1589
1590                    14      HALF DUPLEX/FULL DUPLEX
1591
1592                            IF THIS BIT IS SET, THIS LINE WILL OPERATE IN HALF-DUPLEX MODE. IF NOT S
1593                            IN FULL-DUPLEX MODE.
1594
1595                            IN THIS APPLICATION HALF-DUPLEX MEANS THAT THE DH11 RECEIVER IS BLINDED
1596
1597                    15      AUTO-ECHO ENABLE
1598
1599                            WHEN THIS BIT IS SET, CHARACTERS RECEIVED ON THIS LINE WILL BE HARDWARE
1600                            FURTHER DETAILS.
1601
```

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.

```
1680                            5.3     DH11 FUNCTIONAL LOGIC PARTITIONING
1681                                    ----------------------------------
1682
1683                                    THIS SECTION LISTS ALL OF THE PRINTS FOR ALL OF THE
1684                                MODULES IN THE DH11 SUBSYSTEM. IT BRIEFLY SUMMARIZES THE
1685                                FUNCTIONAL LOGIC DESCRIBED ON EACH PRINT. THIS INFORM-
1686                                ATION MAY PROVE USEFUL FOR A MODULE OR CHIP REPLACEMENT GUIDE
1687                                WHEN THE FUNCTTIONAL AREA OF LOGIC THAT IS FAULTY IS KNOWN TO
1688                                BE INTERMITTENTLY FAILING.
1689
1690                                M7277   CURRENT ADDR REG MEMORY AND ADDR SELECT
1691                                ***************************************************
1692
1693                                        SH3:    CONTROL STROBE MUX FOR THE "LPR" REGISTER
1694
1695                                                TRANSMITTER DATA MUX WITH AUTO-ECHO CONTROL
1696                                                LOGIC SELECTION
1697
1698                                                MSYN / SSYN TIMING CHAIN
1699
1700                                                DH11 MASTER CLEAR LOGIC
1701
1702                                        SH4:    UNIBUS ADDRESS SELECTION LOGIC WITH JUMPERS
1703
1704                                                TRANSMITTER SCAN COUNTER WITH XMITTER STATUS
1705                                                MULTIPLEXOR (BAR N * TBMT N)
1706
1707                                        SH5:    BYTE COUNT AND CURRENT ADDRESS MEMORY WRITE
1708                                                TIMING LOGIC
1709
1710                                                CURRENT ADDRESS MEMORY LOGIC BITS<17:08>
1711
1712                                                UNIBUS ADDRESS DRIVERS BITS <17:08>
1713
1714                                        SH6:    BYTE COUNT AND CURRENT ADDRESS MEMORY ADDRESS
1715                                                SELECT MULTIPLEXOR
1716
1717                                                CURRENT ADDRESS MEMORY LOGIC FOR BITS <07:00>
1718
1719                                                UNIBUS ADDRESS DRIVERS FOR BITS <07:00>
1720
1721                                                TRANSMITTER EVEN/ODD BYTE DATA MULTIPLEXOR
1722
1723                                M7279   FIFO BUFFER
1724                                ********************
1725
1726                                        SH1:    INPUT DATA MULTIPLEXOR FOR SILO MEMORY
1727
1728                                        SH2:    SILO MEMORY CHIPS  (FOUR 64 X 4 CHIPS)
1729
1730                                                SILO MEMORY READ/WRITE TIMING LOGIC
1731
1732                                                "SSR" REGISTER  BITS <13:08>
1733
1734                                                SILO ALARM LEVEL COMPARATOR
1735
```

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

```
1738                                    M7288   LINE PARAMETER CONTROL
1739                                    ********************************
1740
1741                                            SH3:    CLOCK TIMING SIGNAL BUFFERS
1742
1743                                            SH4:    TRANSMITTER CLOCK SELECTION MULTIPLEXORS  LINES<03:00>
1744
1745                                            SH5:    RECEIVER CLOCK SELECTION MULTIPLEXORS  LINES <03:00>
1746
1747                                                    AUTO ECHO AND HALF DUPLEX CONTROL       LINES<03:00>
1748
1749                                            SH6:    TRANSMITTER CLOCK SELECTION MULTIPLEXORS  LINES<07:04>
1750
1751                                            SH7:    RECEIVER CLOCK SELECTION MULTIPLEXORS  LINES <07:04>
1752
1753                                                    AUTO ECHO AND HALF DUPLEX CONTROL       LINES <07:04>
1754
1755                                            SH8:    TRANSMITTER CLOCK SELECTION MULTIPLEXORS  LINES<11:08>
1756
1757                                            SH9:    RECEIVER CLOCK SELECTION MULTIPLEXORS  LINES <11:08>
1758
1759                                                    AUTO ECHO AND HALF DUPLEX CONTROL       LINES <11:08>
1760
1761                                            SH10:   TRANSMITTER CLOCK SELECTION MULTIPLEXORS  LINES<15:12>
1762
1763                                            SH11:   RECEIVER CLOCK SELECTION MULTIPLEXORS  LINES <15:12>
1764
1765                                                    AUTO ECHO AND HALF DUPLEX CONTROL       LINES <15:12>
1766
1767                                    M7280 #1        MULTIPLE UART CARD FOR LINES <0-7>
1768                                    ****************************************************
1769
1770                                            SH2:    UART CHIPS      BIT<1:0>
1771
1772                                                    RECEIVER SCAN MULTIPLEXORS (CLR R DONE, STB RD, MASTER DA, AND M
1773
1774                                            SH3:    UART CHIPS      BIT<3:2>
1775
1776                                            SH4:    UART CHIPS      BIT<5:4>
1777
1778                                                    RECEIVER SCAN MULTIPLEEXORS (MASTER FE AND MASTER PE)
1779
1780                                            SH5:    UART CHIPS      BIT<7:6>
1781
1782                                                    TRANSMITTER SCAN MULTIPLEXOR
1783
1784                                                    -12 VOLT DC REGULATOR
1785
```

```
1786                                    M7280 #2        MULTIPLE UART CARD FOR LINES (8-15)
1787                                    ***************************************************
1788
1789                                            SH2:    UART CHIPS      BIT(9:8)
1790
1791                                                    RECEIVER SCAN MULTIPLEXORS (CLR R DONE, STB RD, MASTER DA, AND M
1792
1793                                            SH3:    UART CHIPS      BIT(11:10)
1794
1795                                            SH4:    UART CHIPS      BIT(13:12)
1796
1797                                                    RECEIVER SCAN MULTIPLEEXORS (MASTER FE AND MASTER PE)
1798
1799                                            SH5:    UART CHIPS      BIT(15:14)
1800
1801                                                    TRANSMITTER SCAN MULTIPLEXOR
1802
1803                                                    -12 VOLT DC REGULATOR
1804
1805                                    M7289   SYSTEM CONTROL AND RCVR SCAN
1806                                    ************************************
1807
1808                                    SH3:    TRANSMITTER AND RECEIVER SCAN LOGIC
1809
1810                                    SH4:    TRANSMITTER AND RECEIVER SCAN TIMING
1811
1812                                    SH5:    HALF-DUPLEX CONTROL LOGIC
1813
1814                                            UART DATA MULTIPLEXORS
1815
1816                                    SH6:    SYSTEM CONTROL REGISTER
1817
1818                                    M4540   DC11 - DH11 CLOCK
1819                                    ***************************
1820
1821                                    SH1:    CRYSTAL OSCILLATOR AND FREQUENCY DIVIDERS
1822
1823                                    M796    UNIBUS MASTER CONTROL
1824                                    *****************************
1825
1826                                    SH1:    NPR CONTROL LOGIC
1827
1828                                    M7281 #1        INTERRUPT CONTROL
1829                                    *********************************
1830
1831                                    "A" SECTION:    RECEIVER INTERRUPT CONTROL LOGIC
1832
1833                                    "B" SECTION:    TRANSMITTER INTERRUPT CONTROL LOGIC
1834
1835                                    M7281 #2        NPR CONTROL
1836                                    ***************************
1837
1838                                    "A" SECTION:    USED TO GAIN CONTROL OF THE BUS FOR NPR XFERS
1839
1840                                    "B" SECTION:    (NOT USED)
1841
```

```
                                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	2	3	4	5	6	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

```
2061                                    DEVICE CABLE AND LEAVE THE DEVICE CABLE
2062                                    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.

```
2156                                    EXAMPLE:        MAINTENANCE HEADER FOR TEST 1
2157                                    ********
2158
2159                                    TEST ABSTRACT:
2160                                    ******************
2161
2162                                            THIS TEST ATTEMPTS TO REFERENCE EACH OF THE EIGHT REGISTERS
2163                                    IN THE DH11 SELECTED FOR TEST USING ITS ASSIGNED UNIBUS ADDRESS. IF
2164                                    ANY ADDRESS FAILS TO RESPOND A BUS ERROR TRAP VECTORS THE TEST TO THE
2165                                    ERROR SET-UP AND CALL ROUTINE. AFTER THE ERROR IS TYPED THE TEST WILL
2166                                    TEST THE NEXT DH11 ADDRESS IN SEQUENCE UNTIL ALL EIGHT ARE TESTED.
2167
2168                                    ERRORS:
2169                                    *******
2170
2171                                    1.) [ERROR  1] REPORTS THAT THE REGISTER WHOSE ADDRESS IS IN R2 FAILED
2172                                        TO RESPOND WITH "SSYN" WHEN REFERENCED.
2173
2174                                    SYNC:           (NONE)
2175                                    *****
2176
2177                                    DEBUG:
2178                                    ******
2179
2180                                    1.) PROBLEM IS MOST LIKELY THE M7277 MODULE.
2181                                    2.) IF ALL EIGHT REGISTERS FAIL TO RESPOND, MAKE SURE THAT YOU CONFIGURED
2182                                        THE PROGRAM PROPERLY BEFORE STARTING.  IF YOU DID, CHECK THE SETTINGS
2183                                        OF THE ADDRESS SELECT JUMPERS ON THE M7277 MODULE.
2184                                    3.) IF ONE OR MORE RESPONDED PROPERLY, SET UP AN ERROR SCOPE LOOP AND
2185                                        BACKTRACK THROUGH THE LOGIC STARTING WITH THE KEY LOGIC SIGNALS
2186                                        LISTED BELOW.
2187
2188                                    KEY LOGIC:
2189                                    **********
2190
2191                                            M7277   SH3     SSYN H                  CE2
2192                                                            DEVICE RESPONDING L     E72-6
2193                                                    SH4     DEVICE SELECTED H       E09-11
2194
2195                                    \
```

```
                                .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
```

```
2252  000024  000200                    200     ;;FOR APT START UP
2253          000044                    .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
2254  000044  000214                    $APTHDR ;;POINT TO APT HEADER BLOCK
2255          000214                    .=.$X   ;;RESET LOCATION COUNTER
2256                              ;;*****************************************************************
2257                              ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
2258                              ;INTERFACE SPEC.
2259
2260  000214                      $APTHD:
2261  000214  000000              $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
2262  000216  001232              $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
2263  000220  000036              $TSTM:  .WORD   30.     ;;RUN TIM OF LONGEST TEST
2264  000222  000170              $PASTM: .WORD   120.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
2265  000224  000170              $UNITM: .WORD   120.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
2266  000226  000052                      .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
2267
2268
2269                              .SBTTL  BASIC DEFINITIONS
2270
2271                              ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
2272          001100              STACK=  1100
2273                              .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
2274                              .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
2275
2276                              ;*MISCELLANEOUS DEFINITIONS
2277          000011              HT=     11              ;;CODE FOR HORIZONTAL TAB
2278          000012              LF=     12              ;;CODE FOR LINE FEED
2279          000015              CR=     15              ;;CODE FOR CARRIAGE RETURN
2280          000200              CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
2281          177776              PS=     177776          ;;PROCESSOR STATUS WORD
2282                              .EQUIV  PS,PSW
2283          177774              STKLMT= 177774          ;;STACK LIMIT REGISTER
2284          177772              PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
2285          177570              DSWR=   177570          ;;HARDWARE SWITCH REGISTER
2286          177570              DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
2287
2288                              ;*GENERAL PURPOSE REGISTER DEFINITIONS
2289          000000              R0=     %0              ;;GENERAL REGISTER
2290          000001              R1=     %1              ;;GENERAL REGISTER
2291          000002              R2=     %2              ;;GENERAL REGISTER
2292          000003              R3=     %3              ;;GENERAL REGISTER
2293          000004              R4=     %4              ;;GENERAL REGISTER
2294          000005              R5=     %5              ;;GENERAL REGISTER
2295          000006              R6=     %6              ;;GENERAL REGISTER
2296          000007              R7=     %7              ;;GENERAL REGISTER
2297          000006              SP=     %6              ;;STACK POINTER
2298          000007              PC=     %7              ;;PROGRAM COUNTER
2299
2300                              ;*PRIORITY LEVEL DEFINITIONS
2301          000000              PR0=    0               ;;PRIORITY LEVEL 0
2302          000040              PR1=    40              ;;PRIORITY LEVEL 1
2303          000100              PR2=    100             ;;PRIORITY LEVEL 2
2304          000140              PR3=    140             ;;PRIORITY LEVEL 3
2305          000200              PR4=    200             ;;PRIORITY LEVEL 4
2306          000240              PR5=    240             ;;PRIORITY LEVEL 5
2307          000300              PR6=    300             ;;PRIORITY LEVEL 6
```

```
2308            000340                  PR7=    340             ;;PRIORITY LEVEL 7
2309
2310                                    ;*"SWITCH REGISTER" SWITCH DEFINITIONS
2311            100000                  SW15=   100000
2312            040000                  SW14=   40000
2313            020000                  SW13=   20000
2314            010000                  SW12=   10000
2315            004000                  SW11=   4000
2316            002000                  SW10=   2000
2317            001000                  SW09=   1000
2318            000400                  SW08=   400
2319            000200                  SW07=   200
2320            000100                  SW06=   100
2321            000040                  SW05=   40
2322            000020                  SW04=   20
2323            000010                  SW03=   10
2324            000004                  SW02=   4
2325            000002                  SW01=   2
2326            000001                  SW00=   1
2327                                    .EQUIV  SW09,SW9
2328                                    .EQUIV  SW08,SW8
2329                                    .EQUIV  SW07,SW7
2330                                    .EQUIV  SW06,SW6
2331                                    .EQUIV  SW05,SW5
2332                                    .EQUIV  SW04,SW4
2333                                    .EQUIV  SW03,SW3
2334                                    .EQUIV  SW02,SW2
2335                                    .EQUIV  SW01,SW1
2336                                    .EQUIV  SW00,SW0
2337
2338                                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
2339            100000                  BIT15=  100000
2340            040000                  BIT14=  40000
2341            020000                  BIT13=  20000
2342            010000                  BIT12=  10000
2343            004000                  BIT11=  4000
2344            002000                  BIT10=  2000
2345            001000                  BIT09=  1000
2346            000400                  BIT08=  400
2347            000200                  BIT07=  200
2348            000100                  BIT06=  100
2349            000040                  BIT05=  40
2350            000020                  BIT04=  20
2351            000010                  BIT03=  10
2352            000004                  BIT02=  4
2353            000002                  BIT01=  2
2354            000001                  BIT00=  1
2355                                    .EQUIV  BIT09,BIT9
2356                                    .EQUIV  BIT08,BIT8
2357                                    .EQUIV  BIT07,BIT7
2358                                    .EQUIV  BIT06,BIT6
2359                                    .EQUIV  BIT05,BIT5
2360                                    .EQUIV  BIT04,BIT4
2361                                    .EQUIV  BIT03,BIT3
2362                                    .EQUIV  BIT02,BIT2
2363                                    .EQUIV  BIT01,BIT1
```

```
2364                                    .EQUIV  BIT00,BIT0
2365
2366                                    ;*BASIC "CPU" TRAP VECTOR ADDRESSES
2367            000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
2368            000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
2369            000014                  TBITVEC=14              ;;"T" BIT
2370            000014                  TRTVEC= 14              ;;TRACE TRAP
2371            000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
2372            000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
2373            000024                  PWRVEC= 24              ;;POWER FAIL
2374            000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
2375            000034                  TRAPVEC=34              ;;"TRAP" TRAP
2376            000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
2377            000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
2378            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       ;;INTERRUPT 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
```

```
2435  001224  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
2436  001226     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
2437  001227     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
2438  001230  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
2439                                  ;;*****************************************************************
2440                                  .SBTTL  APT MAILBOX-ETABLE
2441
2442                                  ;;*****************************************************************
2443                                  .EVEN
2444  001232                          $MAIL:                          ;;APT MAILBOX
2445  001232  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
2446  001234  000000                  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
2447  001236  000000                  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
2448  001240  000000                  $PASS:  .WORD   APASS           ;;PASS COUNT
2449  001242  000000                  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
2450  001244  000000                  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
2451  001246  000000                  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
2452  001250  000000                  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
2453  001252                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
2454  001252     000                  $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
2455  001253     000                  $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
2456  001254  000000                  $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
2457  001256  000000                  $USWR:  .WORD   AUSWR           ;;USER SWITCHES
2458  001260  000000                  $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
2459                                  ;*                              BITS 15-11=CPU TYPE
2460                                  ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
2461                                  ;*                                      11/70=06,PDQ=07,Q=10
2462                                  ;*                              BIT 10=REAL TIME CLOCK
2463                                  ;*                              BIT  9=FLOATING POINT PROCESSOR
2464                                  ;*                              BIT  8=MEMORY MANAGEMENT
2465  001262     000                  $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
2466  001263     000                  $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
2467                                  ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
2468                                  ;*                                      900 NSEC CORE=001
2469                                  ;*                                      300 NSEC BIPOLAR=002
2470                                  ;*                                      500 NSEC MOS=003
2471  001264  000000                  $MADR1: .WORD   AMADR1          ;;HIGH ADDRESS,BLK#1
2472                                  ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
2473  001266     000                  $MAMS2: .BYTE   AMAMS2          ;;HIGH ADDRESS,M.S. BYTE
2474  001267     000                  $MTYP2: .BYTE   AMTYP2          ;;MEM.TYPE,BLK#2
2475  001270  000000                  $MADR2: .WORD   AMADR2          ;;MEM.LAST ADDRESS,BLK#2
2476  001272     000                  $MAMS3: .BYTE   AMAMS3          ;;HIGH ADDRESS,M.S.BYTE
2477  001273     000                  $MTYP3: .BYTE   AMTYP3          ;;MEM.TYPE,BLK#3
2478  001274  000000                  $MADR3: .WORD   AMADR3          ;;MEM.LAST ADDRESS,BLK#3
2479  001276     000                  $MAMS4: .BYTE   AMAMS4          ;;HIGH ADDRESS,M.S.BYTE
2480  001277     000                  $MTYP4: .BYTE   AMTYP4          ;;MEM.TYPE,BLK#4
2481  001300  000000                  $MADR4: .WORD   AMADR4          ;;MEM.LAST ADDRESS,BLK#4
2482  001302  000000                  $VECT1: .WORD   AVECT1          ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
2483  001304  000000                  $VECT2: .WORD   AVECT2          ;;INTERRUPT VECTOR#2BUS PRIORITY#2
2484  001306  000000                  $BASE:  .WORD   ABASE           ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
2485  001310  000000                  $DEVM:  .WORD   ADEVM           ;;DEVICE MAP
2486  001312  000000                  $CDW1:  .WORD   ACDW1           ;;CONTROLLER DESCRIPTION WORD#1
2487  001314  000000                  $CDW2:  .WORD   ACDW2           ;;CONTROLLER DESCRIPTION WORD#2
2488  001316  000000                  $DDW0:  .WORD   ADDW0           ;;DEVICE DESCRIPTOR WORD#0
2489  001320  000000                  $DDW1:  .WORD   ADDW1           ;;DEVICE DESCRIPTOR WORD#1
2490  001322  000000                  $DDW2:  .WORD   ADDW2           ;;DEVICE DESCRIPTOR WORD#2
```

```
 2491  001324  000000                $DDW3:   .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
 2492  001326  000000                $DDW4:   .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
 2493  001330  000000                $DDW5:   .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
 2494  001332  000000                $DDW6:   .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
 2495  001334  000000                $DDW7:   .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
 2496  001336  000000                $DDW8:   .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
 2497  001340  000000                $DDW9:   .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
 2498  001342  000000                $DDW10:  .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
 2499  001344  000000                $DDW11:  .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
 2500  001346  000000                $DDW12:  .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
 2501  001350  000000                $DDW13:  .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
 2502  001352  000000                $DDW14:  .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
 2503  001354  000000                $DDW15:  .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
 2504
 2505
 2506  001356                        $ETEND:
 2507
```

```
                                .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

001356  030302                          EM1             ;"DH11 REGISTER REFERENCE CAUSED TIMEOUT"
001360  030351                          DH1             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR "
001362  030430                          DT1             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2
001364  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 2

001366  030446                          EM2             ;"SYSTEM CONTROL REGISTER ERROR"
001370  030504                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001372  030602                          DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001374  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 3

001376  030624                          EM3             ;"DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG"
001400  030504                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001402  030602                          DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001404  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 4

001406  030702                          EM4             ;"LINE PARAMETER REGISTER ERROR"
001410  030504                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001412  030602                          DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001414  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 5

001416  030740                          EM5             ;"BREAK CONTROL REGISTER ERROR"
001420  030504                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001422  030602                          DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001424  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 6

001426  030775                          EM6             ;"SILO STATUS REGISTER ERROR"
001430  030504                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001432  030602                          DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001434  000000                          0               ;PRINT ALL OCTAL
```

```
                                ;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
```

```
2620
2621                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 17
2622
2623  001536  031500                            EM17            ;"TRANSMITTER TIMING ERROR - LINE #  "
2624  001540  031544                            DH6             ;" (PC)  (PS)  (SP)  TEST  DEVADR  SPEED   TIMEB   TIMEC"
2625  001542  030602                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2626  001544  000000                            0               ;PRINT ALL OCTAL
2627
2628                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 20
2629
2630  001546  031642                            EM20            ;RECEIVER TIMING ERROR - LINE #  "
2631  001550  031544                            DH6             ;" (PC)  (PS)  (SP)  TEST  DEVADR  SPEED   TIMEB   TIMEC"
2632  001552  030602                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2633  001554  000000                            0               ;PRINT ALL OCTAL
2634
2635                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 21
2636
2637  001556  031703                            EM21            ;"RCVR FAILED TO INTERRUPT - LINE #  "
2638  001560  030504                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2639  001562  030602                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2640  001564  000000                            0               ;PRINT ALL OCTAL
2641
2642                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 22
2643
2644  001566  031747                            EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
2645  001570  030504                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2646  001572  030602                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2647  001574  000000                            0               ;PRINT ALL OCTAL
2648
2649                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 23
2650
2651  001576  032023                            EM23            ;"BASIC DATA TEST ERROR - LINE #  "
2652  001600  032064                            DH7             ;" (PC)  (PS)  (SP)  TEST  DEVADR  CHRLNG  WAS  S/B "
2653  001602  030602                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2654  001604  000000                            0               ;PRINT ALL OCTAL
2655
2656
2657                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 24
2658
2659  001606  032161                            EM24            ;"AUTO ECHO TEST ERROR - LINE #  "
2660  001610  030504                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2661  001612  030602                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2662  001614  000000                            0               ;PRINT ALL OCTAL
2663
2664                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 25
2665
2666  001616  032221                            EM25            ;"BREAK BIT TEST ERROR - LINE #  "
2667  001620  030504                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2668  001622  030602                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2669  001624  000000                            0               ;PRINT ALL OCTAL
2670
2671                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 26
2672
2673  001626  032261                            EM26            ;"HALF-DUPLEX TEST ERROR - LINE #  "
2674  001630  030504                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2675  001632  030602                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
```

```
2676  001634  000000                    0               ;PRINT ALL OCTAL
2677
2678                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 27
2679
2680  001636  032323                    EM27            ;"UNEXPECTED BUS ERROR TRAP"
2681  001640  032355                    DH3             ;  (PC)  (PS)  (SP)  TEST  TRPPC  TRPPS
2682  001642  032434                    DT3             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2
2683  001644  000000                    0               ;PRINT ALL OCTAL
2684
2685                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 30
2686
2687  001646  032452                    EM30            ;"UNEXPECTED RSVD INSTR TRAP"
2688  001650  032355                    DH3             ;  (PC)  (PS)  (SP)  TEST  TRPPC  TRPPS
2689  001652  032434                    DT3             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2
2690  001654  000000                    0               ;PRINT ALL OCTAL
2691
2692                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 31
2693
2694  001656  032505                    EM31            ;"AUTO ECHO DATA COMPARE ERROR - LINE #  "
2695  001660  032555                    DH4             ;  (PC)   (PS)   (SP)    TEST   WASADR  SBADR   WAS    S/B "
2696  001662  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2697  001664  000000                    0               ;PRINT ALL OCTAL
2698
2699                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 32
2700
2701  001666  032652                    EM32            ;"AUTO ECHO TEST TIMEOUT - LINE #  "
2702  001670  032714                    DH5             ;"  (PC)  (LPRG)   TEST"
2703  001672  032742                    DT4             ;$ERRPC,$TMP0,$TMP2
2704  001674  000000                    0               ;PRINT ALL OCTAL
2705
2706
2707                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 33
2708
2709  001676  032752                    EM33            ;"PARITY LOGIC TEST ERROR - LINE #  "
2710  001700  030504                    DH2             ;"  (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B"
2711  001702  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2712  001704  000000                    0               ;PRINT ALL OCTAL
2713
2714                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 34
2715
2716  001706  033015                    EM34            ;"MULTI-LINE PARITY DATA TEST ERROR - LINE #   - SUBTEST #  "
2717  001710  032555                    DH4             ;"  (PC)  (PS)  (SP)  TEST  WASADR  SBADR  WAS  S/B "
2718  001712  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2719  001714  000000                    0               ;PRINT ALL OCTAL
2720
2721                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 35
2722
2723  001716  033110                    EM35            ;"MULTI-LINE PARITY DATA TEST TIMEOUT"
2724  001720  033154                    DH14            ;"  (PC)  (LPRG)  LINACT "
2725  001722  033204                    DT6             ;"$ERRPC,$TMP0,$TMP3"
2726  001724  000000                    0               ;PRINT ALL OCTAL
2727
2728                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 36
2729
2730  001726  033214                    EM36            ;CHAR AVAILABLE TIMEOUT"
2731  001730  032714                    DH5             ;"  (PC)  (LPRG)   TEST"
```

```
2732  001732  032742                  DT4             ;$ERRPC,$TMP0,$TMP2
2733  001734  000000                  0               ;PRINT ALL OCTAL
2734
2735                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 37
2736
2737  001736  033243                  EM37            ;"DATA COMPARE ERROR - LINE #  "
2738  001740  032555                  DH4             ;" (PC)  (PS)  (SP)  TEST  WASADR  SBADR  WAS  S/B "
2739  001742  030602                  DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2740  001744  000000                  0               ;PRINT ALL OCTAL
2741
2742                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 40
2743
2744  001746  033301                  EM40            ;"BUFFER ACTIVE REG ERROR - LINE #  "
2745  001750  030504                  DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2746  001752  030602                  DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2747  001754  000000                  0               ;PRINT ALL OCTAL
2748
2749                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 41
2750
2751  001756  033344                  EM41            ;"RCVR FALSE INTERRUPT"
2752  001760  032714                  DH5             ;" (PC)  (LPRG)  TEST"
2753  001762  032742                  DT4             ;$ERRPC,$TMP0,$TMP2
2754  001764  000000                  0               ;PRINT ALL OCTAL
2755
2756                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 42
2757
2758  001766  033371                  EM42            ;"SILO OVERFLOW ERROR"
2759  001770  032714                  DH5             ;" (PC)  (LPRG)  TEST"
2760  001772  032742                  DT4             ;$ERRPC,$TMP0,$TMP2
2761  001774  000000                  0               ;PRINT ALL OCTAL
2762
2763                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 43
2764
2765  001776  033415                  EM43            ;"SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT"
2766  002000  030504                  DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2767  002002  030602                  DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2768  002004  000000                  0               ;PRINT ALL OCTAL
2769
2770                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 44
2771
2772  002006  033475                  EM44            ;"NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT"
2773  002010  030504                  DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2774  002012  030602                  DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2775  002014  000000                  0               ;PRINT ALL OCTAL
2776
2777                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 45
2778
2779  002016  033557                  EM45            ;"XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT"
2780  002020  030504                  DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2781  002022  030602                  DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2782  002024  000000                  0               ;PRINT ALL OCTAL
2783
2784                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 46
2785
2786  002026  033635                  EM46            ;"CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE #  "
2787  002030  033723                  DH10            ;" (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS     S/B"
```

```
2788  002032  034020                    DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4
2789  002034  000000                    0               ;PRINT ALL OCTAL
2790
2791                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 47
2792
2793  002036  034042                    EM47            ;"BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE #  "
2794  002040  033723                    DH10            ; (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS     S/B"
2795  002042  034020                    DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4
2796  002044  000000                    0               ;PRINT ALL OCTAL
2797
2798                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 50
2799
2800  002046  034123                    EM50            ;"TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  '
2801  002050  030504                    DH2             ;" (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
2802  002052  030602                    DT2             ;"$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
2803  002054  000000                    0               ;PRINT ALL OCTAL
2804
2805                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 51
2806
2807  002056  034201                    EM51            ;"NPR LOGIC TEST 2 ERROR"
2808  002060  034230                    DH11            ;" (PC)   LINACT  LINCHK   TEST   DEVADR  REGADR   WAS     S/B"
2809  002062  034020                    DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4"
2810  002064  000000                    0               ;PRINT ALL OCTAL
2811
2812                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 52
2813
2814  002066  034325                    EM52            ;"BASIC DATA COMPARE ERROR"
2815  002070  030504                    DH2             ;" (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
2816  002072  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
2817  002074  000000                    0               ;PRINT ALL OCTAL
2818
2819                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 53
2820
2821  002076  034123                    EM50            ;"TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  "
2822  002100  034356                    DH12            ;" (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
2823  002102  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2824  002104  000000                    0               ;PRINT ALL OCTAL
2825
2826                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 54
2827
2828  002106  031747                    EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
2829  002110  034356                    DH12            ;" (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
2830  002112  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2831  002114  000000                    0               ;PRINT ALL OCTAL
2832
2833                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 55
2834
2835  002116  031747                    EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
2836  002120  034453                    DH13            ;" (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG  SCRWAS  SCRS/B
2837  002122  030602                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2838  002124  000000                    0               ;PRINT ALL OCTAL
2839
2840                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 56
2841
2842  002126  034552                    EM56            ;"OVERRUN BIT FAILED TO SET - LINE #  "
2843  002130  030504                    DH2             ;" (PC)   (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
```

```
2844  002132  030602                  DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
2845  002134  000000                  0               ;PRINT ALL OCTAL
2846
2847                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 57
2848
2849  002136  034617                  EM57            ;"STORAGE OVERFLOW BIT FAILED - LINE #  "
2850  002140  030504                  DH2             ;"  (PC)    (PS)     (SP)     TEST    DEVADR   REGADR    WAS      S/B"
2851  002142  030602                  DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
2852  002144  000000                  0               ;PRINT ALL OCTAL
2853
2854                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 60
2855  002146  036073                  MSG4            ;MODEM CONTROL ERROR.. RUN DZDHK
2856  002150  034666                  DH15            ;"  (PC) DEVADR  LINE    "
2857  002152  032742                  DT4             ;$ERRPC,$TMP0,$TMP2
2858  002154  000000                  0
2859
```

```
2860  002156  005000                  BEGIN:  CLR     R0              ;INIT R0 TO INDICATE DEFAULT PARAMETERS
2861  002160  005067  026104          BEGINA: CLR     TITFLG          ;INIT TITLE MESSAGE FLAG
2862                                  .SBTTL  INITIALIZE THE COMMON TAGS
2863                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
2864  002164  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
2865  002170  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
2866  002172  022706  001140                  CMP     #SWR,R6 ;;DONE?
2867  002176  001374                          BNE     .-6             ;;LOOP BACK IF NO
2868  002200  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
2869                                  ;;INITIALIZE A FEW VECTORS
2870  002204  012737  021000  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
2871  002212  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
2872  002220  012737  021270  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
2873  002226  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
2874  002234  012737  024054  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
2875  002242  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
2876  002250  012737  024136  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
2877  002256  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
2878  002264  016767  016422  016412          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
2879  002272  005067  176724                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
2880  002276  005067  176722                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
2881  002302  112767  000001  176605          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
2882  002310  012767  002310  176570          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2883  002316  012767  002316  176564          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
2884                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2885                                  ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
2886  002324  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
2887  002330  012737  002364  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
2888  002336  012767  177570  176574          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
2889  002344  012767  177570  176570          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
2890  002352  022777  177777  176560          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
2891  002360  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
2892                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
2893  002362  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
2894  002364  012716  002372          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
2895  002370  000002                          RTI
2896  002372  012767  000176  176540  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
2897  002400  012767  000174  176534          MOV     #DISPREG,DISPLAY
2898  002406  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
2899
2900  002412  005067  176622                  CLR     $PASS           ;;CLEAR PASS COUNT
2901  002416  132767  000200  176627          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
2902  002424  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
2903  002426  012767  001254  176504          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
2904  002434                          67$:
2905                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2906  002434  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
2907  002440  001012                          BNE     68$             ;;BRANCH IF YES
2908  002442  126727  176604  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
2909  002450  001406                          BEQ     68$             ;;BRANCH IF YES
2910  002452  026727  176462  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
2911  002460  001005                          BNE     69$             ;;BRANCH IF NO
2912  002462  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
2913  002464  000403                          BR      69$
2914  002466  112767  000001  176440  68$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
2915  002474                          69$:
```

```
2916  002474  012767  026620  175302  START1: MOV     #BUSER,ERRVEC   ;SET UP THE BUS ERROR VECTOR
2917  002502  012767  000340  175276          MOV     #340,ERRVEC+2
2918  002510  012767  026674  175272          MOV     #RESERR,RESVEC  ;SET UP THE RSVD INSTR VECTOR
2919  002516  012767  000340  175266          MOV     #340,RESVEC+2
2920  002524  005767  025540                  TST     TITFLG          ;HAVE WE TYPED TITLE ONCE ?
2921  002530  001012                          BNE     1$              ;BR IF YES
2922  002532  104401                          TYPE                    ;GO TYPE PROGRAM TITLE
2923  002534  034722                          TITLE
2924  002536  005167  025526                  COM     TITFLG          ;SET FLAG - TYPE TITLE ONLY ONCE PER LOAD
2925  002542  032777  000001  176370          BIT     #BIT0,@SWR      ;DO WE WANT TO AUTOSIZE?
2926  002550  001002                          BNE     1$              ;BRANCH IF NOT.
2927  002552  004767  022352                  JSR     PC,AUTOSZ       ;GO AUTOSIZE.
2928  002556  005767  025140          1$:     TST     VCFLG           ;START AT 200 ??
2929  002562  001413                          BEQ     11$             ;BR IF NOT
2930  002564  032777  000001  176346          BIT     #BIT0,@SWR      ;ARE PARAMETERS TO BE INPUT MANUALLY?
2931  002572  001003                          BNE     9$              ;BRANCH IF YES.
2932  002574  016700  025430                  MOV     ADRVEC,R0       ;OTHERWISE, GET ADDRESSES BETWEEN VECTORS FROM AUTOSIZER
2933  002600  000402                          BR      10$
2934  002602  004767  023272          9$:     JSR     PC,INPARA       ;GO ASK FOR PARAMETERS
2935  002606  005067  025110          10$:    CLR     VCFLG           ;RE INIT VECTOR FLAG
2936  002612  005700                  11$:    TST     R0              ;USE DEFAULT PARAMETERS ?
2937  002614  001407                          BEQ     START2          ;BR IF YES
2938  002616  022700  177777                  CMP     #-1,R0          ;CHANGE DH SELECT PARAM ONLY ?
2939  002622  001002                          BNE     2$              ;BR IF NOT
2940  002624  000167  023420                  JMP     INPAR3          ;GO ASK FOR SELECT PARAM.
2941  002630  000167  023324          2$:     JMP     INPAR           ;GO ASK FOR ALL PARAMETERS
2942
2943  002634  012767  027620  025406  START2: MOV     #DHADTB-2,ADPTR ;GET POINTER TO ADDRESS TABLE
2944  002642  012767  027660  025402          MOV     #DHVCTB-2,VCPTR ;GET POINTER TO VECTOR TABLE
2945  002650  012767  027722  025376          MOV     #BRLVL-2,BRPTR  ;GET POINTER TO BR LEVEL TABLE
2946  002656  012767  177777  025356          MOV     #-1,DHNUM       ;START WITH DH #00
2947  002664  012767  000001  024336          MOV     #1,SELMSK       ;SET UP DH11 BIT TEST MARKER
2948
2949  002672  005267  025344          RESTRT: INC     DHNUM           ;GENERATE DH11 DEV NUMBER
2950  002676  062767  000002  025344          ADD     #2,ADPTR        ;UPDATE TABLE POINTERS
2951  002704  062767  000002  025340          ADD     #2,VCPTR
2952  002712  062767  000002  025334          ADD     #2,BRPTR
2953  002720  036767  024304  024304          BIT     SELMSK,DHSEL    ;TEST FOR SELECTED DH11
2954  002726  001004                          BNE     RSTRTA          ;BR IF SELECTED FOR TEST
2955  002730  006367  024274          REST1:  ASL     SELMSK          ;SHIFT MARKER TO TEST NEXT DH11
2956  002734  001737                          BEQ     START2          ;BR IF 16 TESTED - START OVER
2957  002736  000755                          BR      RESTRT          ;GO TEST IF THIS ONE SELECTED
2958  002740  017767  025304  024256  RSTRTA: MOV     @ADPTR,DHADR    ;SET UP DH11 ADDRESS
2959  002746  017767  025300  024252          MOV     @VCPTR,DHVCT    ;SET UP THE DH11 VECTOR ENTRY
2960  002754  017767  025274  025256          MOV     @BRPTR,DHRLVL   ;GET BR LEVEL VALUES
2961  002762  004567  021640                  JSR     R5,SUNUM        ;GO SET DH NUMBER IN THE MESSAGE BUFFER
2962  002766  030242                          DHNUM
2963  002770  035012                          TITLE2+20
2964  002772  104401                          TYPE                    ;GO PRINT "TESTING DH11 #XX"
2965  002774  034772                          TITLE2
2966  002776  004767  021316                  JSR     PC,LDTBF1       ;GO LOAD XMITTR OUTPUT BUFFER WITH
2967                                                                  ;BINARY COUNT PATTERN
2968  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
```

```
3086  003126  062745  000004          ADD     #LPR,-(R5)      ;GENERATE LPR ADDRESS
3087  003132  005745                  TST     -(R5)           ;POINT R5 TO FIRST ADDR ENTRY (SCR)
3088  003134  005004                  CLR     R4              ;RESULT S/B 000000 AFTER MASTER CLEAR
3089
3090  003136  012502          1$:     MOV     (R5)+,R2        ;GET REG ADDRESS
3091  003140  022705  027252          CMP     #MSTCLR+10,R5   ;DONE ALL FOUR REGS ??
3092  003144  001415                  BEQ     TST3            ;;BR IF YES
3093  003146  012712  177777  2$:     MOV     #-1,(R2)        ;SET 1'S IN REGISTER
3094  003152  052711  004000          BIS     #BIT11,(R1)     ;ISSUE MASTER CLEAR
3095  003156  011203                  MOV     (R2),R3         ;GET CONTENT OF REGISTER
3096  003160  001766                  BEQ     1$              ;;BR IF IT'S ALL ZEROES
3097
3098  003162  004767  021214          JSR     PC,SUER2        ;GO SET UP ERROR INFO
3099  003166  012767  003146  175714  MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
3100  003174  104003                  ERROR   3               ;MASTER CLR FAILED TO CLR SEL. REG.
3101  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
                        %
```

```
3158  003202  012767  003232  175700          MOV     #4$,$LPERR      ;SET UP ERROR LOOP RETURN
3159  003210  010102                          MOV     R1,R2           ;GET REGISTER ADDRESS
3160  003212  012705  000001                  MOV     #1,R5           ;SET UP TO START WITH BIT00
3161
3162  003216  030567  024362          1$:     BIT     R5,RGMSK1       ;SHALL WE TEST THIS BIT ?
3163  003222  001003                          BNE     4$              ;BR IF YES
3164  003224  006305                  2$:     ASL     R5              ;SHIFT TO TST NEXT BIT
3165  003226                          3$:
3166  003226  001430                          BEQ     TST4            ;;<BR IF DONE ALL R/W BITS>
3167  003230  000772                          BR      1$              ;GO TEST NEXT BIT
3168
3169  003232  010504                  4$:     MOV     R5,R4           ;RESULT S/B IN R4
3170  003234  005012                          CLR     (R2)            ;INIT REG BEING TESTED
3171  003236  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
3172  003244  010512                          MOV     R5,(R2)         ;SET THE BIT
3173  003246  011203                          MOV     (R2),R3         ;GET THE WAS DATA
3174  003250  020403                          CMP     R4,R3           ;RESULT = S/B DATA ??
3175  003252  001403                          BEQ     5$              ;BR IF YES
3176
3177  003254  004767  021122                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
3178  003260  104002                          ERROR   2               ;SELECTED BIT FAILED TO SET IN SCR
3179
3180  003262  005004                  5$:     CLR     R4              ;SET UP TO CLEAR THE BIT S/B=000000
3181  003264  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3182  003272  040512                          BIC     R5,(R2)         ;CLR THE SELECTED BIT
3183  003274  011203                          MOV     (R2),R3         ;GET THE WAS DATA
3184  003276  001403                          BEQ     6$              ;BR IF IT CLEARED
3185
3186  003300  004767  021076                  JSR     PC,SUER2        ;GO SET UP THE ERROR INFO
3187  003304  104002                          ERROR   2               ;SELECTED BIT FAILED TO CLEAR IN SCR
3188  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:
                                **********
```

```
3301
3302                                        SAME AS TEST 03 WITH THE FOLLOWING ADDITION
3303
3304                                        M7289   SH4     74121 ONE-SHOTS E35-6, E23-6
3305                                  %
3306  003374  012767  003424  175506        MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
3307  003402  010102                        MOV     R1,R2           ;MAKE IT REG ADDR TOO
3308  003404  012705  000001                MOV     #1,R5           ;INIT BIT TEST MARKER
3309  003410  030567  024202          1$:   BIT     R5,RGMSK6       ;IS IT A READ ONLY BIT ??
3310  003414  001003                        BNE     3$              ;BR IF YES - TEST IT
3311  003416  006305                  2$:   ASL     R5              ;SHIFT THE BIT MARKER
3312  003420  001457                        BEQ     TST6            ;;BR IF DONE ALL SELECTED BITS
3313  003422  000772                        BR      1$              ;GO TEST FOR THIS BIT
3314
3315  003424  010504                  3$:   MOV     R5,R4           ;SET UP S/B DATA
3316  003426  052704  001000                BIS     #BIT09,R4       ;PUT IN THE MAINT. BIT
3317  003432  005012                        CLR     (R2)            ;INIT REG BEING TESTED
3318  003434  052712  001000                BIS     #BIT09,(R2)     ;TURN ON MAINT. MODE
3319  003440  112761  000000  000016        MOVB    #0,SSR(R1)      ;SCOPE SYNC
3320  003446  050512                        BIS     R5,(R2)         ;SET THE SELECTED BIT
3321  003450  011203                        MOV     (R2),R3         ;GET THE WAS DATA
3322  003452  020304                        CMP     R3,R4           ;DID SELECTED BIT GET SET ??
3323  003454  001404                        BEQ     4$              ;BR IF IT DID
3324
3325  003456  004767  020720                JSR     PC,SUER2        ;GO SET UP ERROR INFO
3326  003462  104002                        ERROR   2               ;SELECTED BIT FAILED TO SET IN MAINT MODE
3327  003464  000754                        BR      2$              ;GO TEST NEXT BIT
3328
3329  003466  042712  001000          4$:   BIC     #BIT09,(R2)     ;TURN OFF MAINT. MODE
3330  003472  042704  001000                BIC     #BIT09,R4       ;CLR MAINT BIT IN S/B DATA
3331  003476  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3332  003504  040512                        BIC     R5,(R2)         ;ATTEMPT TO CLR SELECTED BIT
3333  003506  011203                        MOV     (R2),R3         ;GET THE WAS DATA
3334  003510  020304                        CMP     R3,R4           ;DID BIT GET CLEARED ??
3335  003512  001404                        BEQ     5$              ;BR IF IT DIDN'T
3336
3337  003514  004767  020662                JSR     PC,SUER2        ;GO SET UP ERROR INFO
3338  003520  104002                        ERROR   2               ;SELECTED BIT GOT CLEARED WITH MAINT MODE OFF
3339  003522  000735                        BR      2$              ;GO TEST NEXT BIT
3340
3341  003524  012704  001000          5$:   MOV     #BIT09,R4       ;SET UP S/B DATA
3342  003530  050412                        BIS     R4,(R2)         ;SET MAINT. MODE
3343  003532  012761  000000  000004        MOV     #0,LPR(R1)      ;SCOPE SYNC
3344  003540  040512                        BIC     R5,(R2)         ;NOW CLR SELECTED BIT
3345  003542  011203                        MOV     (R2),R3         ;GET THE WAS DATA
3346  003544  020304                        CMP     R3,R4           ;DID BIT GET CLEARED OK ??
3347  003546  001723                        BEQ     2$              ;BR IF YES
3348
3349  003550  004767  020626                JSR     PC,SUER2        ;GO SET UP ERROR INFO
3350  003554  104002                        ERROR   2               ;FAILED TO CLR SELECTED BIT IN MAINT MODE
3351  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
```

```
3408  003576  012705  000001              MOV     #1,R5           ;INIT BIT TEST MARKER
3409  003602  030567  024002      1$:     BIT     R5,RGMSK3       ;TEST THIS BIT ??
3410  003606  001003                      BNE     3$              ;BR IF YES
3411  003610  006305              2$:     ASL     R5              ;SHIFT THE MARKER
3412  003612  001430                      BEQ     TST7            ;;BR IF DONE ALL BITS
3413  003614  000772                      BR      1$              ;GO TEST NXT BIT
3414
3415  003616  010504              3$:     MOV     R5,R4           ;SET UP S/B DATA
3416  003620  005012                      CLR     (R2)            ;INIT REG BEING TESTED
3417  003622  112761  000000  000016      MOVB    #0,SSR(R1)      ;SCOPE SYNC
3418  003630  010512                      MOV     R5,(R2)         ;SET LPR BIT
3419  003632  011203                      MOV     (R2),R3         ;GET THE WAS DATA
3420  003634  020304                      CMP     R3,R4           ;DID IT SET
3421  003636  001403                      BEQ     4$              ;BR IF IT SET PROPERLY
3422
3423  003640  004767  020536              JSR     PC,SUER2        ;GO SET UP ERROR INFO
3424  003644  104004                      ERROR   4               ;LPR BIT FAILED TO SET PROPERLY
3425
3426  003646  005004              4$:     CLR     R4              ;GET READY TO CLEAR SELECTED BIT
3427  003650  112761  000000  000017      MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3428  003656  040512                      BIC     R5,(R2)         ;CLEAR THE BIT
3429  003660  011203                      MOV     (R2),R3         ;GET THE WAS DATA
3430  003662  001752                      BEQ     2$              ;BR IF BIT CLEARED PROPERLY
3431
3432  003664  004767  020512              JSR     PC,SUER2        ;GO SET UP ERROR INFO
3433  003670  104004                      ERROR   4               ;LPR BIT FAILED TO CLEAR PROPERLY
3434  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
```

```
3491  003706  062702  000014            ADD     #BKR,R2
3492  003712  012705  000001            MOV     #1,R5           ;INIT BIT TEST MARKER
3493  003716  030567  023670    1$:     BIT     R5,RGMSK4       ;TEST THIS BIT ??
3494  003722  001003                    BNE     3$              ;BR IF YES
3495  003724  006305            2$:     ASL     R5              ;SHIFT BIT MARKER
3496  003726  001430                    BEQ     TST10           ;;BR IF ALL BITS TESTED
3497  003730  000772                    BR      1$              ;GO TEST THE BIT
3498
3499  003732  010504            3$:     MOV     R5,R4           ;SET UP S/B DATA
3500  003734  005012                    CLR     (R2)            ;INIT REG BEING TESTED
3501  003736  112761  000000  000016    MOVB    #0,SSR(R1)      ;SCOPE SYNC
3502  003744  050512                    BIS     R5,(R2)         ;SET THE SELECTED BIT IN "BKR"
3503  003746  011203                    MOV     (R2),R3         ;GET THE WAS DATA
3504  003750  020304                    CMP     R3,R4           ;DID BIT SET OK
3505  003752  001403                    BEQ     4$              ;BR IF YES
3506
3507  003754  004767  020422            JSR     PC,SUER2        ;GO SET UP ERROR INFO
3508  003760  104005                    ERROR   5               ;BKR BIT FAILED TO SET PROPERLY
3509
3510  003762  005004            4$:     CLR     R4              ;SET UP S/B DATA
3511  003764  112761  000000  000017    MOVB    #0,SSR+1(R1)    ;SCOPEE SYNC
3512  003772  040512                    BIC     R5,(R2)         ;CLEAR BKR BIT
3513  003774  011203                    MOV     (R2),R3         ;GET THE BKR WAS DATA
3514  003776  001752                    BEQ     2$              ;BR IF BKR BIT CLEARED OK
3515
3516  004000  004767  020376            JSR     PC,SUER2        ;GO SET UP ERROR INFO
3517  004004  104005                    ERROR   5               ;BKR BIT FAILED TO CLR PROPERLY
3518  004006  000746                    BR      2$              ;GO SELECT NEXT BIT
```

```
3519                                    ;;*****************************************************************
3520                                    ;*TEST 10       TEST THAT ALL R/W BITS IN "SSR" CAN BE SET/CLR
3521                                    ;;*****************************************************************
3522  004010  000004                    TST10:  SCOPE
3523                                    .REM    %
3524                                    TEST ABSTRACT:
3525                                    **************
3526
3527                                            THIS TEST VERIFIES THAT ALL R/W BITS IN THE SILO STATUS REGISTER
3528                                    (SSR) CAN BE SET AND CLEARED INDIVIDUALLY. IT USES A BIT MASK (RGMSK5:  100077)
3529                                    TO DEFINE THE R/W BITS (15,5,4,3,2,1, AND 0). R5 ALWAYS CONTAINS THE BIT
3530                                    CURRENTLY SELECTED FOR TEST. THE TEST SEQUENCE IS AS FOLLOWS:
3531
3532                                            1. SELECT A BIT TO TEST
3533                                            2. SET THE BIT AND VERIFY THAT IT SET PROPERLY
3534                                            3. CLEAR THE BIT AND VERIFY THAT IT CLEARED PROPERLY
3535                                            4. REPEAT 1 THRU 3 UNTIL ALL BITS ARE TESTED
3536
3537                                    ANY ERRORS DETECTED ARE REPORTED AND THEN THE TEST RESUMES WITH THE
3538                                    NEXT BIT IN SEQUENCE.
3539
3540                                    ERRORS:
3541                                    ********
3542
3543                                    1.) [ERROR  6] IS CALLED TO REPOORT BOTH FAIL TO SET PROPERLY AND
3544                                        FAIL TO CLEAR PROPERLY FAULTS.
3545
3546                                    SYNC:
3547                                    *****
3548
3549                                    1.) FAIL TO SET:  M7277 SH4     LOAD LPR H      EP2
3550                                    2.) FAIL TO CLR:  M7277 SH4     LOAD BCR H      FU1
3551
3552                                    DEBUG:
3553                                    ******
3554
3555                                    1.) THE ONLY DIFFERENCES BETWEEN THEE DATA PATHS USED BY THIS TEST AND
3556                                        THAT USED BY THE PREVIOUS TESTS ARE THE ACTUAL "SSR" REGISTER
3557                                        CHIPS AND THE INPUT PIN SELECTED ON THE OUTPUT DATA MULTIPLEXORS.
3558                                    2.) IF ALL BITS FAIL IT IS MOST LIKELY THE M7277
3559                                    3.) IF BITS <13:08> FAIL IT IS MOST LIKELY THE M7279
3560                                    4.) IF JUST ONE OR TWO BITS FAIL IT IS MOST LIKELY THE M7278
3561
3562                                    KEY LOGIC:
3563                                    ***********
3564
3565                                            M7277   SH4     LOAD SSR LOW BYTE H     CR1
3566                                                            LOAD SSR HIGH BYTE H    CP2
3567                                                            DATA TO BUS H           EN2
3568                                                            DATA SOURCE (A,B,C) H   DU1,DU2,DT2
3569
3570                                            M7279   SH2     SSR <13:08> H  (E20 AND E24)
3571
3572                                            M7278   SH5 - SH8       REGISTER CHIPS E53,E68, OR E69 (74175'S)
3573                                                                    OUTPUT MUX CHIPS - (74151'S  PIN 12)
3574                                    %
```

```
3575  004012  012767  004046  175070          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
3576  004020  010102                          MOV     R1,R2           ;GENERATE "SSR" ADDRESS IN R2
3577  004022  062702  000016                  ADD     #SSR,R2
3578  004026  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MARKER
3579  004032  030567  023556          1$:     BIT     R5,RGMSK5       ;TEST THIS BIT ??
3580  004036  001003                          BNE     3$              ;BR IF YES
3581  004040  006305                  2$:     ASL     R5              ;SHIFT BIT MARKER
3582  004042  001435                          BEQ     TST11           ;;BR IF ALL BITS TESTED
3583  004044  000772                          BR      1$              ;GO TEST THE BIT
3584
3585  004046  010504                  3$:     MOV     R5,R4           ;SET UP S/B DATA
3586  004050  005012                          CLR     (R2)            ;INIT REG BEING TESTED
3587  004052  012761  000000  000004          MOV     #0,LPR(R1)      ;SCOPE SYNC
3588  004060  050512                          BIS     R5,(R2)         ;SET THE SELECTED BIT IN "SSR"
3589  004062  011203                          MOV     (R2),R3         ;GET THE WAS DATA
3590  004064  042703  077700                  BIC     #77700,R3       ;CLEAR OUT DON'T CARE BITS
3591  004070  020304                          CMP     R3,R4           ;DID BIT SET OK
3592  004072  001403                          BEQ     4$              ;BR IF YES
3593
3594  004074  004767  020302                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
3595  004100  104006                          ERROR   6               ;SSR BIT FAILED TO SET PROPERLY
3596
3597  004102  005004                  4$:     CLR     R4              ;SET UP S/B DATA
3598  004104  012761  000000  000014          MOV     #0,BKR(R1)      ;SCOPE SYNC
3599  004112  040512                          BIC     R5,(R2)         ;CLEAR SSR BIT
3600  004114  011203                          MOV     (R2),R3         ;GET THE SSR WAS DATA
3601  004116  042703  077700                  BIC     #77700,R3       ;CLEAR JUNK BITS
3602  004122  020304                          CMP     R3,R4           ;DID THE SSR BIT GET CLEARED ??
3603  004124  001745                          BEQ     2$              ;BR IF SSR BIT CLEARED OK
3604
3605  004126  004767  020250                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
3606  004132  104006                          ERROR   6               ;SSR BIT FAILED TO CLR PROPERLY
3607  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$:     BIS     R5,R4           ;SET BIT "N" IN S/B DATA
3677  004240  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3678  004246  050512                          BIS     R5,(R2)         ;SET BIT "N" IN LPR
3679  004250  011203                          MOV     (R2),R3         ;GET THE WAS DATA
3680  004252  020304                          CMP     R3,R4           ;DID BIT "N" SET PROPERLY ?
3681  004254  001744                          BEQ     2$              ;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"
```

```
3742  004332  005012                        CLR     (R2)            ;INIT REG BEING TESTED
3743  004334  016712  023252                MOV     RGMSK4,(R2)     ;SET ALL R/W BITS IN BKR
3744  004340  112761  000000  000016        MOVB    #0,SSR(R1)      ;SCOPE SYNC
3745  004346  040512                        BIC     R5,(R2)         ;CLEAR BIT "N" IN BKR
3746  004350  011203                        MOV     (R2),R3         ;GET THE WAS DATA
3747  004352  020304                        CMP     R3,R4           ;DID IT CLEAR OK ?
3748  004354  001404                        BEQ     4$              ;BR IF YES
3749
3750  004356  004767  020020                JSR     PC,SUER2        ;GO SET UP ERROR INFO
3751  004362  104005                        ERROR   5               ;BIT "N" FAILED TO CLR PROPERLY
3752  004364  000754                        BR      2$              ;GO TEST NEXT BIT
3753
3754  004366  050504                4$:     BIS     R5,R4           ;SET BIT "N" IN S/B DATA
3755  004370  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3756  004376  050512                        BIS     R5,(R2)         ;SET BIT "N" IN BKR
3757  004400  011203                        MOV     (R2),R3         ;GET THE WAS DATA
3758  004402  020304                        CMP     R3,R4           ;DID BIT "N" SET PROPERLY ?
3759  004404  001744                        BEQ     2$              ;BR IF YES
3760
3761  004406  004767  017770                JSR     PC,SUER2        ;GO SET UP ERROR INFO
3762  004412  104005                        ERROR   5               ;BIT "N" FAILED TO SET PROPERLY
3763  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                        1$: MOV     R1,R2           ;SET UP THE REG ADDR
004430  062702  000016                    ADD     #SSR,R2
004434  012705  000001                    MOV     #1,R5           ;INIT BIT TEST MASK
004440  030567  023150                    BIT     R5,RGMSK5       ;TEST THIS BIT ??
004444  001003                          BNE     3$              ;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"
```

```
3820  004462  005012                        CLR     (R2)            ;INIT REG BEING TESTED
3821  004464  016712  023124                MOV     RGMSK5,(R2)     ;SET ALL R/W BITS IN SSR
3822  004470  012761  000000  000004        MOV     #0,LPR(R1)      ;SCOPE SYNC
3823  004476  040512                        BIC     R5,(R2)         ;CLEAR BIT "N" IN SSR
3824  004500  011203                        MOV     (R2),R3         ;GET THE WAS DATA
3825  004502  042703  077700                BIC     #77700,R3       ;CLEAR JUNK BITS
3826  004506  020304                        CMP     R3,R4           ;DID IT CLEAR OK ?
3827  004510  001404                        BEQ     4$              ;BR IF YES
3828
3829  004512  004767  017664                JSR     PC,SUER2        ;GO SET UP ERROR INFO
3830  004516  104006                        ERROR   6               ;BIT "N" FAILED TO CLR PROPERLY
3831  004520  000752                        BR      2$              ;GO TEST NEXT BIT
3832
3833  004522  050504                4$:     BIS     R5,R4           ;SET BIT "N" IN S/B DATA
3834  004524  012761  000000  000014        MOV     #0,BKR(R1)      ;SCOPE SYNC
3835  004532  050512                        BIS     R5,(R2)         ;SET BIT "N" IN SSR
3836  004534  011203                        MOV     (R2),R3         ;GET THE WAS DATA
3837  004536  042703  077700                BIC     #77700,R3       ;CLEAR JUNK BITS
3838  004542  020304                        CMP     R3,R4           ;DID BIT "N" SET PROPERLY ?
3839  004544  001740                        BEQ     2$              ;BR IF YES
3840
3841  004546  004767  017630                JSR     PC,SUER2        ;GO SET UP ERROR INFO
3842  004552  104006                        ERROR   6               ;BIT "N" FAILED TO SET PROPERLY
3843  004554  000734                        BR      2$              ;GO SELECT NEXT BIT
```

```
3844                                    ;;*********************************************************************
3845                                    ;*TEST 14        "CAR" MEMORY ADDRESSING TEST
3846                                    ;;*********************************************************************
3847  004556  000004                    TST14:  SCOPE
3848                                    .REM    %
3849                                    TEST ABSTRACT:
3850                                    ***************
3851
3852                                            THIS TEST VERIFIES THAT EACH LOCATION IN THE CURRENT ADDRESS
3853                                    MEMORY CAN BE UNIQUELY ADDRESSED. IT WRITES THE PATTERN SHOWN BELOW
3854                                    INTO THE MEMORY AND THEN READS BACK EACH LOCATION TO VERIFY THAT IT
3855                                    WAS WRITTEN CORRECTLY. SINCE THE MEMORY LOGIC IS PARTITIONED INTO
3856                                    FOUR 16 X 4 READ/WRITE MEMORY CHIPS, THE PATTERN RESULTS IN THE LINE
3857                                    NUMBER (00 - 17(8)) BEING WRITTEN AS DATA INTO TO EACH LOCATION
3858                                    (00 - 17(8)) IN EACH CHIP. THAT IS:  LOC00 = 00, LOC 01 = 01,
3859                                    .......LOC 17 = 17.
3860
3861
3862                                            MEMORY PATTERN: LOCATION        CONTENTS        (BOTH OCTAL)
3863
3864                                                            00              000000
3865                                                            01              010421
3866                                                            02              021042
3867                                                            03              031463
3868                                                            04              042104
3869                                                            05              052525
3870                                                            06              063146
3871                                                            07              073567
3872                                                            10              104210
3873                                                    11                              114631
3874                                                            12              125252
3875                                                            13              135673
3876                                                            14              146314
3877                                                            15              156735
3878                                                            16              167356
3879                                                            17              177777
3880
3881                                    ANY ERRORS DETECTED ARE REPORTED AND THEN THE TEST RESUMES CHECKING
3882                                    THEE NEXT LOCATION IN SEQUENCE UNTIL ALL 16. HAVE BEEN CHECKED.
3883
3884                                            NOTE:   THIS TEST ALWAYS CHECKS ALL 16. LINES REGARDLESS
3885                                                    OF HOW THE LINE SELECTION PARAMETER WAS INITIALLY
3886                                                    SET UP.
3887
3888                                    ERRORS:
3889                                    ********
3890
3891                                    1.) [ERROR  7] IS CALLED TO REPORT ANY LINES (LOCATIONS) THAT FAIL.
3892                                        THE FAILING LINE # IS INCLUDED AS PART OF THE ERROR HEADER MESSAGE.
3893
3894                                    SYNC:
3895                                    ******
3896
3897                                    1.) WRITE SYNC: M7277   SH4     LOAD SSR LOW BYTE H     CR1
3898                                    2.) READ SYNC:  M7277   SH4     LOAD SSR HIGH BYTE H    CP2
3899
```

```
                                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.
```

```
3956  004634  062704  010421          ADD     #10421,R4       ;GENERATE NEW DATA
3957  004640  000760                  BR      1$              ;GO DO NEXT LINE
3958
3959  004642  004767  017666    3$:   JSR     PC,SELINE       ;GO SELECT A LINE NO.
3960  004646  000434                  BR      7$              ;;BR IF CHECKED ALL LINES
3961  004650  105767  023370          TSTB    LINE            ;DOING LINE 00 ?
3962  004654  001001                  BNE     4$              ;BR IF NOT
3963  004656  005004                  CLR     R4              ;INIT S/B DATA
3964
3965  004660  156711  023360    4$:   BISB    LINE,(R1)       ;SELECT A LINE
3966  004664  112761  000000  000017  MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3967  004672  011203                  MOV     (R2),R3         ;GET CONTENTS OF CAR
3968  004674  020304                  CMP     R3,R4           ;WAS DATA OK ?
3969  004676  001412                  BEQ     5$              ;BR IF YES
3970
3971  004700  004767  017476          JSR     PC,SUER2        ;GO SET UP ERROR INFO
3972  004704  004567  017716          JSR     R5,SUNUM        ;SET UP LINE NO. IN MSG BUFFER
3973  004710  030244                  LINE
3974  004712  031077                  EM7+47
3975  004714  012767  004732  174166  MOV     #6$,$LPERR      ;SET UP ERROR LOOP RETURN
3976  004722  104007                  ERROR   7               ;CAR ADDRESSING ERROR
3977
3978  004724  062704  010421    5$:   ADD     #10421,R4       ;GENERATE NEW S/B DATA
3979  004730  000744                  BR      3$              ;GO CHECK NEXT LINE
3980
3981  004732  005067  023306    6$:   CLR     LINE            ;RESTART AT LINE 00 IF LOOPING
3982  004736  000721                  BR      1$              ;GO RESTART
3983
3984  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
```

```
4041                                    DEBUG:
4042                                    ******
4043
4044                                    1.) ANALYZE THE ERROR REPORTS CAREFULLY ASKING THE FOLLOWING QUESTIONS:
4045
4046                                            A. DOES THE FAULT AFFECT ONLY ONE LINE ?
4047                                            B. DOES THE FAULT AFFECT ONLY ONE 4-BIT DATA GROUP ?
4048                                               IE <15:12>, <11:08>, <07:04>, OR <03:00>
4049                                            C. DOES ANY DATA AT ALL APPEAR TO BE WRITTEN ?
4050
4051                                    2.) IF "A" IS TRUE THEN SUSPECT AN ADDRESSING PROBLEM IN THE MEMORY
4052                                        ADDRESS MUX.
4053
4054                                    3.) IF "B" IS TRUE THEN SUSPECT A DATA MUX, UP-COUNTER, MEMORY,
4055                                        OR INVERTER CHIP PROBLEM.
4056
4057                                    4.) IF "C" IS TRUE SUSPECT A MEMORY WRITE TIMING PROBLEM.
4058
4059                                    5.) IN MOST CASES THE FAULT IS MOST LIKELY THE M7277 OR M7278.
4060
4061                                    KEY LOGIC:
4062                                    **********
4063
4064                                            M7277   SH4     LOAD BC H               FU2
4065                                                            DATA TO BUS H           EN2
4066                                                            DATA SOURCE (A,B,C)     DU1,DU2,DT2
4067
4068                                                    SH5     MEMADD SOURCE SEL H     E55-8
4069                                                            BC MEM WRITE ENAB L     E57-4
4070                                                            BUF ADDRS TO BUS H      E33-1 (SHD BE LOW)
4071                                            M7278   SH3     BITS<15:08>
4072                                                            74157 INPUT MUX CHIPS E18,E19
4073                                                            74193 UP COUNTER CHIPS E27,E26
4074                                                            7489 MEMORY CHIPS E33,E34
4075                                                            7404 INVERTER CHIPS E41,E42
4076
4077                                                    SH4     BITS<07:00>
4078                                                            74157 INPUT MUX CHIPS E16,E17
4079                                                            74193 UP COUNTER CHIPS E24,E25
4080                                                            7489 MEMORY CHIPS E31,E32
4081                                                            7404 INVERTER CHIPS E39,E40
4082
4083                                                    SH5 THRU SH8    74151 DATA MUX OUTPUT CHIPS (PIN 1 INPUT)
4084                                    %
4085  004750  010102                            MOV     R1,R2           ;COPY IT IN R2
4086  004752  062702  000010                   ADD     #BCR,R2         ;SET UP REGADR IN R2
4087  004756  016767  022252  174234           MOV     LINSEL,$TMP7    ;SAVE LINE SELECT PARAMETER
4088  004764  012767  177777  022242           MOV     #-1,LINSEL      ;DO ALL LINES FOR THIS TEST
4089  004772  004767  017536          1$:      JSR     PC,$ELINE       ;GO SELECT A LINE NO.
4090  004776  000415                           BR      3$              ;;BR IF DONE ALL SELECTED LINES
4091  005000  105767  023240                   TSTB    LINE            ;DOING LINE 00 ?
4092  005004  001001                           BNE     2$              ;BR IF NOT
4093  005006  005004                           CLR     R4              ;INIT TEST DATA
4094
4095  005010  156711  023230          2$:      BISB    LINE,(R1)       ;SELECT A LINE
4096  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 ??
```

```
 4183  005210  001406                        BEQ     3$              ;;BR IF ALL 1'S
 4184
 4185  005212  004767  017164                JSR     PC,SUER2        ;GO SET UP ERROR INFO
 4186  005216  012767  005174  173664        MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
 4187  005224  104007                        ERROR   7               ;FAILED TO SET ALL 1'S IN SELECTED CAR
 4188
 4189  005226  005004                3$:     CLR     R4              ;RESULT IN CAR S/B = 000000
 4190  005230  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
 4191  005236  010412                        MOV     R4,(R2)         ;CLEAR SELECTED CAR
 4192  005240  011203                        MOV     (R2),R3         ;GET THE WAS DATA
 4193  005242  001741                        BEQ     1$              ;BR IF CAR GOT CLEARED
 4194
 4195  005244  004767  017132                JSR     PC,SUER2        ;GO SET UP FOR ERROR CALL
 4196  005250  012767  005226  173632        MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN
 4197  005256  104007                        ERROR   7               ;FAILED TO CLR ALL BITS IN SELECTED CAR
 4198  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 ??
```

```
4255  005334  001406                        BEQ     3$              ;;BR IF ALL 1'S
4256
4257  005336  004767  017040                JSR     PC,SUER2        ;GO SET UP ERROR INFO
4258  005342  012767  005320  173540        MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
4259  005350  104010                        ERROR   10              ;FAILED TO SET ALL 1'S IN SELECTED BCR
4260
4261  005352  005004                3$:     CLR     R4              ;RESULT IN BCR S/B = 000000
4262  005354  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
4263  005362  010412                        MOV     R4,(R2)         ;CLEAR SELECTED BCR
4264  005364  011203                        MOV     (R2),R3         ;GET THE WAS DATA
4265  005366  001741                        BEQ     1$              ;BR IF BCR GOT CLEARED
4266
4267  005370  004767  017006                JSR     PC,SUER2        ;GO SET UP FOR ERROR CALL
4268  005374  012767  005352  173506        MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN
4269  005402  104010                        ERROR   10              ;FAILED TO CLR ALL BITS IN SELECTED BCR
4270  005404  000732                        BR      1$              ;GO TEST NEXT LINE
```

```
                                ;;*******************************************************************
                                ;*TEST 20       "CAR" MEMORY PATTERNS TEST / O'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
```

```
4327  005450  116711  173534          3$:     MOVB    $TMP3,(R1)      ;SELECT A LINE TO CLEAR
4328  005454  005012                          CLR     (R2)            ;CLR CAR FOR THAT LINE
4329  005456  105267  173526                  INCB    $TMP3           ;GENERATE NEW LINE NO.
4330  005462  126727  173522  000020          CMPB    $TMP3,#20       ;DONE CLEARING ALL LINES ?
4331  005470  001367                          BNE     3$              ;BR IF NOT
4332
4333  005472  116711  022550                  MOVB    LINEA,(R1)      ;SET LINE SELECT BITS
4334  005476  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
4335  005504  016712  173474                  MOV     $TMP1,(R2)      ;LOAD CAR WITH TEST PATTERN
4336
4337  005510  105067  173472                  CLRB    $TMP2           ;INIT A LINE COUNTER
4338  005514  016704  173464          4$:     MOV     $TMP1,R4        ;SET UP S/B DATA
4339  005520  116711  173462                  MOVB    $TMP2,(R1)      ;SET LINE SELECT IN SCR
4340  005524  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
4341  005532  011203                          MOV     (R2),R3         ;GET WAS DATA
4342  005534  126767  173446  022502          CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
4343  005542  001401                          BEQ     5$              ;BR IF IT IS
4344  005544  005004                          CLR     R4              ;MAKE S/B DATA = 000000
4345  005546  020304          5$:             CMP     R3,R4           ;CORRECT DATA IN CAR ?
4346  005550  001412                          BEQ     6$              ;BR IF YES
4347
4348  005552  004767  016712                  JSR     PC,SUER4        ;GO SET UP ERROR IN F0
4349  005556  004567  017044                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
4350  005562  001206                          $TMP2
4351  005564  033720                          EM46+63
4352  005566  012767  005444  173314          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
4353  005574  104046                          ERROR   46              ;INCORRECT DATA READ FROM CAR
4354
4355  005576  105267  173404          6$:     INCB    $TMP2           ;GENERATE NEXT LINE NO.
4356  005602  122767  000020  173376          CMPB    #20,$TMP2       ;DONE ALL LINES ?
4357  005610  001707                          BEQ     11$             ;BR IF YES
4358  005612  000740                          BR      4$              ;GO CHECK NEXT LINE
```

```
                        ;;*****************************************************************
                        ;*TEST 21       "BCR" MEMORY PATTERNS TEST / O'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
```

```
4415  005656  116711  173326          3$:   MOVB   $TMP3,(R1)      ;SELECT A LINE TO CLEAR
4416  005662  005012                        CLR    (R2)            ;CLR BCR FOR THAT LINE
4417  005664  105267  173320                INCB   $TMP3           ;GENERATE NEW LINE NO.
4418  005670  126727  173314  000020        CMPB   $TMP3,#20       ;DONE CLEARING ALL LINES ?
4419  005676  001367                        BNE    3$              ;BR IF NOT
4420
4421  005700  116711  022342                MOVB   LINEA,(R1)      ;SET LINE SELECT BITS
4422  005704  112761  000000  000016        MOVB   #0,SSR(R1)      ;SCOPE SYNC
4423  005712  016712  173266                MOV    $TMP1,(R2)      ;LOAD BCR WITH TEST PATTERN
4424
4425  005716  105067  173264                CLRB   $TMP2           ;INIT A LINE COUNTER
4426  005722  016704  173256          4$:   MOV    $TMP1,R4        ;SET UP S/B DATA
4427  005726  116711  173254                MOVB   $TMP2,(R1)      ;SELECT A LINE TO CHECK
4428  005732  112761  000000  000017        MOVB   #0,SSR+1(R1)    ;SCOPE SYNC
4429  005740  011203                        MOV    (R2),R3         ;GET WAS DATA
4430  005742  126767  173240  022274        CMPB   $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
4431  005750  001401                        BEQ    5$              ;BR IF IT IS
4432  005752  005004                        CLR    R4              ;MAKE S/B DATA = 000000
4433  005754  020304                  5$:   CMP    R3,R4           ;CORRECT DATA IN BCR ?
4434  005756  001412                        BEQ    6$              ;BR IF YES
4435
4436  005760  004767  016504                JSR    PC,SUER4        ;GO SET UP ERROR IN F0
4437  005764  004567  016636                JSR    R5,SUNUM        ;GO SET UP LINE NO.IN MSG BUFFER
4438  005770  001206                        $TMP2
4439  005772  034120                        EM47+56
4440  005774  012767  005652  173106        MOV    #2$,$LPERR      ;SET UP ERROR LOOP RETURN
4441  006002  104047                        ERROR  47              ;INCORRECT DATA READ FROM BCR
4442
4443  006004  105267  173176          6$:   INCB   $TMP2           ;GENERATE NEXT LINE NO.
4444  006010  122767  000020  173170        CMPB   #20,$TMP2       ;DONE ALL LINES ?
4445  006016  001707                        BEQ    11$             ;BR IF YES
4446  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 ?
```

```
4503  006102  001367                        BNE     3$              ;BR IF NOT
4504
4505  006104  116711  022136                MOVB    LINEA,(R1)      ;SET LINE SELECT IN SCR
4506  006110  112761  000000  000016        MOVB    #0,SSR(R1)      ;SCOPE SYNC
4507  006116  005012                        CLR     (R2)            ;CLEAR THE CAR UNDER TEST
4508
4509  006120  105067  173062                CLRB    $TMP2           ;INIT A LINE COUNTER
4510  006124  005004                4$:     CLR     R4              ;MAKE S/B DATA = 000000
4511  006126  116711  173054                MOVB    $TMP2,(R1)      ;SELECT A LINE TO CHECK
4512  006132  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
4513  006140  011203                        MOV     (R2),R3         ;GET WAS DATA
4514  006142  126767  173040  022074        CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
4515  006150  001401                        BEQ     5$              ;BR IF IT IS
4516  006152  010504                        MOV     R5,R4           ;MAKE S/B DATA = 177777
4517  006154  020304                5$:     CMP     R3,R4           ;CORRECT DATA IN CAR ?
4518  006156  001412                        BEQ     6$              ;BR IF YES
4519
4520  006160  004767  016304                JSR     PC,SUER4        ;GO SET UP ERROR IN F0
4521  006164  004567  016436                JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
4522  006170  001206                        $TMP2
4523  006172  033720                        EM46+63
4524  006174  012767  006056  172706        MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
4525  006202  104046                        ERROR   46              ;INCORRECT DATA READ FROM CAR
4526
4527  006204  105267  172776                INCB    $TMP2           ;GENERATE NEXT LINE NO.
4528  006210  122767  000020  172770        CMPB    #20,$TMP2       ;DONE ALL LINES ?
4529  006216  001711                        BEQ     1$              ;BR IF YES
4530  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 ?
```

```
4587  006302  001367                        BNE     3$              ;BR IF NOT
4588
4589  006304  116711  021736                MOVB    LINEA,(R1)      ;SET LINE SELECT BITS
4590  006310  112761  000000  000016        MOVB    #0,SSR(R1)      ;SCOPE SYNC
4591  006316  005012                        CLR     (R2)            ;CLEAR THE BCR UNDER TEST
4592
4593  006320  105067  172662                CLRB    $TMP2           ;INIT A LINE COUNTER
4594  006324  005004                4$:     CLR     R4              ;MAKE S/B DATA = 000000
4595  006326  116711  172654                MOVB    $TMP2,(R1)      ;SELECT A LINE TO CHECK
4596  006332  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
4597  006340  011203                        MOV     (R2),R3         ;GET WAS DATA
4598  006342  126767  172640  021674        CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
4599  006350  001401                        BEQ     5$              ;BR IF IT IS
4600  006352  010504                        MOV     R5,R4           ;MAKE S/B DATA = 177777
4601  006354  020304                5$:     CMP     R3,R4           ;CORRECT DATA IN BCR ?
4602  006356  001412                        BEQ     6$              ;BR IF YES
4603
4604  006360  004767  016104                JSR     PC,SUER4        ;GO SET UP ERROR IN F0
4605  006364  004567  016236                JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
4606  006370  001206                        $TMP2
4607  006372  034120                        EM47+56
4608  006374  012767  006256  172506        MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
4609  006402  104047                        ERROR   47              ;INCORRECT DATA READ FROM BCR
4610
4611  006404  105267  172576                INCB    $TMP2           ;GENERATE NEXT LINE NO.
4612  006410  122767  000020  172570        CMPB    #20,$TMP2       ;DONE ALL LINES ?
4613  006416  001711                        BEQ     1$              ;BR IF YES
4614  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 THE
                                            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                CLRB    $TMP4           ;INIT A LINE COUNTER
006546  016704  172434        5$:     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  105267  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 THE 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
```

```
4784  006716  116713  021317            MOVB    DHTLVL,(R3)
4785  006722  012705  000001            MOV     #1,R5           ;INIT BIT TEST MARKER
4786  006726  030567  020666    1$:     BIT     R5,INTMSK       ;TEST THIS BIT ??
4787  006732  001003                    BNE     3$              ;BR IF YES
4788  006734  006305            2$:     ASL     R5              ;SHIFT THE MARKER
4789  006736  001437                    BEQ     6$              ;BR IF TESTED ALL REQUIRED BITS
4790  006740  000772                    BR      1$              ;GO TEST FOR THIS ONE
4791
4792  006742  012706  001100    3$:     MOV     #STACK,SP       ;RESET SP FOR ERROR LOOPING
4793  006746  004767  020172            JSR     PC,CHPS2        ;GO LOCK OUT INTRS
4794  006752  012711  004000            MOV     #BIT11,(R1)     ;CLEAR THE DH11 INTERFACE
4795  006756  010504                    MOV     R5,R4           ;SET UP S/B DATA
4796  006760  050511                    BIS     R5,(R1)         ;SET THE TEST I.E. BIT
4797  006762  004767  020142            JSR     PC,CHPS1        ;GO CLEAR PSW
4798  006766  000240                    NOP                     ;WAIT A BIT TO ALLOW INTR
4799  006770  000761                    BR      2$              ;OK - GO DO NEXT I.E. BIT
4800
4801  006772  004767  020162    4$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
4802  006776  011103                    MOV     (R1),R3         ;GET THE WAS DATA
4803  007000  004767  015402            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4804  007004  104011                    ERROR   11              ;DH11 RCVR SHOULD NOT HAVE INTERRUPTED
4805  007006  012716  006734            MOV     #2$,(SP)        ;SET UP TO RETURN
4806  007012  000002                    RTI                     ;RETURN TO TEST NEXT BIT
4807
4808  007014  004767  020140    5$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
4809  007020  011103                    MOV     (R1),R3         ;GET THE WAS DATA
4810  007022  004767  015360            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4811  007026  104012                    ERROR   12              ;XMITTER SHOULD NOT HAVE INTERRUPTED
4812  007030  012716  006734            MOV     #2$,(SP)        ;SET UP TO RETURN
4813  007034  000002                    RTI                     ;RETURN TO TEST NEXT BIT
4814
4815  007036  012706  001100    6$:     MOV     #STACK,SP       ;RESET THE SP JUST IN CASE
4816  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
```

```
 4873  007060  016703  020142                MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
 4874  007064  012723  007216                MOV     #3$,(R3)+       ;GO TO 3$ IF RCVR INTRS
 4875  007070  116723  021144                MOVB    DHRLVL,(R3)+
 4876  007074  105723                        TSTB    (R3)+           ;UPDATE POINTER
 4877  007076  012723  007172                MOV     #2$,(R3)+       ;GO TO 3$ ON XMITTR INTRS
 4878  007102  116713  021133                MOVB    DHTLVL,(R3)
 4879  007106  012711  004000          1$:   MOV     #BIT11,(R1)     ;CLR THE DH11
 4880  007112  012706  001100                MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
 4881  007116  004767  020022                JSR     PC,CHPS2        ;GO LOCK OUT INTRS
 4882  007122  012711  001000                MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
 4883  007126  052711  000100                BIS     #BIT06,(R1)     ;SET CHAR AVAILABLE I.E. BIT
 4884  007132  052711  000200                BIS     #BIT07,(R1)     ;SET THE CHAR AVAIL BIT TO FORCE INTR
 4885  007136  004767  017766                JSR     PC,CHPS1        ;GO CLEAR PSW
 4886  007142  000240                        NOP                     ;GIVE IT A LITTLE TIME
 4887
 4888  007144  004767  020010                JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4889  007150  011103                        MOV     (R1),R3         ;GET THE WAS DATA
 4890  007152  005011                        CLR     (R1)            ;CLEAR OUT THE SCR
 4891  007154  005011                        CLR     (R1)
 4892  007156  012704  001300                MOV     #1300,R4        ;SET UP S/B DATA
 4893  007162  004767  015220                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4894  007166  104013                        ERROR   13              ;TIMED OUT AWAITING CHAR AVAIL INTR
 4895  007170  000412                        BR      3$              ;GO EXIT TEST
 4896
 4897  007172  004767  017762          2$:   JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4898  007176  011103                        MOV     (R1),R3         ;GET WAS DATA
 4899  007200  012704  001300                MOV     #1300,R4        ;SET UP S/B DATA
 4900  007204  005011                        CLR     (R1)            ;CLR OUT SCR REG
 4901  007206  005011                        CLR     (R1)
 4902  007210  004767  015172                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4903  007214  104012                        ERROR   12              ;UNEXPECTED XMITTR INTR
 4904
 4905  007216  012706  001100          3$:   MOV     #STACK,SP       ;RESET THE SP
 4906  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
```

```
4963  007306  052711  040000          BIS     #BIT14,(R1)     ;SET SILO OVFLW I.E. BIT
4964  007312  052711  010000          BIS     #BIT12,(R1)     ;SET THE SILO FULL BIT TO FORCE INTR
4965  007316  004767  017606          JSR     PC,CHPS1        ;GO CLEAR PSW
4966  007322  000240                  NOP                     ;GIVE IT A LITTLE TIME
4967
4968  007324  004767  017630          JSR     PC,SAPS         ;SAVE THE ERROR PSW
4969  007330  011103                  MOV     (R1),R3         ;GET THE WAS DATA
4970  007332  005011                  CLR     (R1)            ;CLEAR OUT THE SCR
4971  007334  005011                  CLR     (R1)
4972  007336  012704  051000          MOV     #51000,R4       ;SET UP S/B DATA
4973  007342  004767  015040          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4974  007346  104043                  ERROR   43              ;TIMED OUT AWAITING SILO OVFLW INTR
4975  007350  000412                  BR      3$              ;GO EXIT TEST
4976
4977  007352  004767  017602  2$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
4978  007356  011103                  MOV     (R1),R3         ;GET WAS DATA
4979  007360  012704  051000          MOV     #51000,R4       ;SET UP S/B DATA
4980  007364  005011                  CLR     (R1)            ;CLR OUT SCR REG
4981  007366  005011                  CLR     (R1)
4982  007370  004767  015012          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4983  007374  104012                  ERROR   12              ;UNEXPECTED XMITTR INTR
4984
4985  007376  012706  001100  3$:     MOV     #STACK,SP       ;RESET THE SP
4986  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
```

```
5043  007466  052711  020000            BIS     #BIT13,(R1)     ;SET XMITTR I.E. BIT
5044  007472  052711  002000            BIS     #BIT10,(R1)     ;SET THE NON EX MEM BIT TO FORCE INTR
5045  007476  004767  017426            JSR     PC,CHPS1        ;GO CLEAR PSW
5046  007502  000240                    NOP                     ;GIVE IT A LITTLE TIME
5047
5048  007504  004767  017450            JSR     PC,SAPS         ;SAVE THE ERROR PSW
5049  007510  011103                    MOV     (R1),R3         ;GET THE WAS DATA
5050  007512  005011                    CLR     (R1)            ;CLEAR OUT THE SCR
5051  007514  005011                    CLR     (R1)
5052  007516  012704  023000            MOV     #23000,R4       ;SET UP S/B DATA
5053  007522  004767  014660            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5054  007526  104044                    ERROR   44              ;TIMED OUT AWAITING NON EX MEM INTR
5055  007530  000412                    BR      3$              ;GO EXIT TEST
5056
5057  007532  004767  017422    2$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
5058  007536  011103                    MOV     (R1),R3         ;GET WAS DATA
5059  007540  012704  023000            MOV     #23000,R4       ;SET UP S/B DATA
5060  007544  005011                    CLR     (R1)            ;CLR OUT SCR REG
5061  007546  005011                    CLR     (R1)
5062  007550  004767  014632            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5063  007554  104011                    ERROR   11              ;UNEXPECTED RCVR INTR
5064
5065  007556  012706  001100    3$:     MOV     #STACK,SP       ;RESET THE SP
5066  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
```

```
5201                                    CLOCK SIGNAL OR BYTE COUNT DECODER ON M7278 SH3 (XMIT FINISHED PULSE L)
5202                                2.  IF GROUP OF 8 LINES <15:08> OR <07:00> FAIL TO INTERRUPT ON TIME, SUSPECT
5203                                    THE BUFFERED CLOCK SIGNALS (TOP AND BOT) ON THE M7288 SH3
5204                                3.  IF ONLY ONE LINE FAILS TO INTERRUPT ON TIME, SUSPECT LOSS OF CLOCK IN
5205                                    LINE MULTIPLEXORS M7288 SH4-SH11.
5206                                4.  IF LINE INTERRUPTED OK BUT "SCR", "BAR", "CAR", OR "BCR" WAS INCORRECT
5207                                    REFER TO KEY LOGIC SIGNALS BELOW.
5208
5209
5210                                KEY LOGIC:
5211                                **********
5212
5213                                XMIT DONE FAILED TO SET:
5214
5215                                        M7289   SH6     SCR15 H FR2
5216                                        M7278   SH3     XMIT FINISHED PULSE L   AR1
5217
5218                                "BAR" BIT FAILED TO CLEAR:
5219
5220                                        M7278   SH3     CLR BAR <15:00> L
5221                                                SH5     BAR <15:12> H   E54, E55 (7474)
5222                                                SH6     BAR <11:08> H   E62, E63 (7474)
5223                                                SH7     BAR <07:04> H   E70, E71 (7474)
5224                                                SH8     BAR <03:00> H   E78, E79 (7474)
5225
5226                                "CAR" REG NOT INCREMENTED:
5227
5228                                        M7277   SH6     END CYCLE PULSE DLY H   FL2
5229                                                SH5 AND SH6 "CAR" MEMORY LOGIC
5230                                        M796            END CYCLE H P2
5231
5232                                "BCR" REG NOT INCREMENTED:
5233
5234                                        M796    END CYCLE L     N2
5235                                        M7278   SH3 AND SH4 "BCR" MEMORY LOGIC
5236
5237                                %
5238  007750  012767  010004  171132          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
5239  007756  016703  017244                  MOV     DHVCT,R3        ;GET THE FIRST VECTOR ADDRESS
5240  007762  062703  000004                  ADD     #4,R3           ;POINT TO XMITTR ENTRY
5241  007766  012723  010150                  MOV     #4$,(R3)+       ;GO TO 4$ ON XMITTR INTR
5242  007772  116713  020243                  MOVB    DHTLVL,(R3)
5243  007776  004767  014532          1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO. TO TEST
5244  010002  000552                  BR      8$              ;BR IF TESTED ALL SELECTED LINES
5245
5246  010004  012706  001100          2$:     MOV     #STACK,SP       ;RESET SP FOR ERROR LOOPS
5247  010010  010102                  MOV     R1,R2           ;SET UP REGADR
5248  010012  012704  120000                  MOV     #120000,R4      ;SET UP S/B DATA
5249  010016  156704  020222                  BISB    LINE,R4
5250  010022  004767  014660                  JSR     PC,CLCABC       ;GO CLEAR CAR AND BCR MEMORIES
5251  010026  004767  017112                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
5252  010032  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE DH11 INTERFACE
5253  010036  156711  020202                  BISB    LINE,(R1)       ;SELECT A LINE NO.
5254  010042  012761  177777  000010          MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO -1
5255  010050  012761  033500  000004          MOV     #33500,LPR(R1)  ;SET UP LINE PARAMETERS
5256  010056  056761  017154  000012          BIS     LINMSK,BAR(R1)  ;ACTIVATE SELECTED LINE
```

```
5257  010064  052711  020000            BIS     #BIT13,(R1)     ;ENABLE INTERRUPT ON XMIT DONE
5258  010070  004767  017034            JSR     PC,CHPS1        ;GO CLEAR PSW
5259
5260  010074  012767  000001  020170    MOV     #1,TIMEA        ;INIT TIMER A
5261  010102  005067  020166            CLR     TIMEB           ;INIT TIMER B
5262  010106  000240            3$:     NOP                     ;DO NOTHING WAIT
5263  010110  004767  016662            JSR     PC,TIMEIT       ;CALL TIMER
5264  010114  000774                    BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC AROUND
5265                                                            ;THIS BRANCH IF TIMEOUT OCCURS
5266
5267  010116  004767  017036            JSR     PC,SAPS         ;SAVE THE ERROR PSW
5268  010122  011103                    MOV     (R1),R3         ;GET THE WAS DATA
5269  010124  042703  000200            BIC     #BIT07,R3       ;WE'RE NOT INTERESTED IN THIS BIT
5270  010130  004767  014252            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5271  010134  004567  014466            JSR     R5,SUNUM        ;GO SET LINE NO. IN ERROR MSG
5272  010140  030244                    LINE
5273  010142  031444                    EM15+43
5274  010144  104015                    ERROR   15              ;TIMEOUT WHILE AWAITING XMIT INTR
5275  010146  000713                    BR      1$              ;GO TEST NEXT LINE
5276
5277  010150  005711            4$:     TST     (R1)            ;DID XMIT DONE SET ??
5278  010152  100411                    BMI     5$              ;BR IF YES
5279
5280  010154  004767  017000            JSR     PC,SAPS         ;SAVE THE ERROR PSW
5281  010160  011103                    MOV     (R1),R3         ;GET THE WAS DATA
5282  010162  004767  014220            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5283  010166  004767  000174            JSR     PC,9$           ;GO SET UP( SOME ERROR STUFF
5284  010172  104014                    ERROR   14              ;XMIT DONE FAILED TO SET
5285  010174  000700                    BR      1$              ;GO TEST NEXT LINE
5286
5287  010176  016103  000012    5$:     MOV     BAR(R1),R3      ;GET WAS DATA FROM "BAR"
5288  010202  001413                    BEQ     6$              ;BR IF BAR BIT GOT CLEARED
5289
5290  010204  004767  016750            JSR     PC,SAPS         ;SAVE THE ERROR PSW
5291  010210  062702  000012            ADD     #BAR,R2         ;SET UP REGADR
5292  010214  005004                    CLR     R4              ;SET UP S/B DATA
5293  010216  004767  014164            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5294  010222  004767  000140            JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
5295  010226  104014                    ERROR   14              ;BAR BIT FAILED TO CLEAR
5296  010230  000662                    BR      1$              ;GO TEST NEXT LINE
5297
5298  010232  016103  000006    6$:     MOV     CAR(R1),R3      ;GET THE WAS DATA FROM CAR
5299  010236  022703  000001            CMP     #1,R3           ;DID IT GET INCREMENTED ?
5300  010242  001414                    BEQ     7$              ;BR IF YES
5301
5302  010244  004767  016710            JSR     PC,SAPS         ;SAVE THE ERROR PSW
5303  010250  012704  000001            MOV     #1,R4           ;SET UP S/B DATA
5304  010254  062702  000006            ADD     #CAR,R2         ;SET UP REGADR
5305  010260  004767  014122            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5306  010264  004767  000076            JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
5307  010270  104014                    ERROR   14              ;CAR REG NOT  INCREMENTED PROPERLY
5308  010272  000641                    BR      1$              ;GO TEST NEXT LINE
5309
5310  010274  016103  000010    7$:     MOV     BCR(R1),R3      ;GET WAS DATA FROM BCR
5311  010300  001636                    BEQ     1$              ;BR IF BCR GOT INCREMENTED TO 000000
5312
```

```
5313  010302  004767  016652          JSR     PC,SAPS         ;SAVE THE ERROR PSW
5314  010306  005004                  CLR     R4              ;SET UP S/B DATA
5315  010310  062702  000010          ADD     #BCR,R2         ;SET UP REGADR
5316  010314  004767  014066          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5317  010320  004767  000042          JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
5318  010324  104014                  ERROR   14              ;BCR REG NOT INCREMENTED PROPERLY
5319  010326  000623                  BR      1$              ;GO TEST NEXT LINE
5320
5321  010330  012711  004000  8$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
5322  010334  016703  016666          MOV     DHVCT,R3        ;GET THE VECTOR ADDR
5323  010340  062703  000004          ADD     #4,R3           ;POINT TO XMIT VECTOR
5324  010344  010313                  MOV     R3,(R3)         ;RESTORE TRAP CATCHER
5325  010346  062723  000002          ADD     #2,(R3)+
5326  010352  005013                  CLR     (R3)
5327  010354  004767  016550          JSR     PC,CHPS1        ;GO CLEAR PSW
5328  010360  012706  001100          MOV     #STACK,SP       ;RESET THE STACK POINTER
5329  010364  000405                  BR      TST33           ;;GO TO NEXT TEST
5330
5331  010366  004567  014234  9$:     JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG.
5332  010372  030244                  LINE
5333  010374  031376                  EM14+44
5334  010376  000207                  RTS     PC              ;RETURN TO REPORT ERROR
```

```
5335                                ;;*****************************************************************
5336                                ;*TEST 33       TRANSMITTR NPR LOGIC TEST 2
5337                                ;;*****************************************************************
5338  010400  000004                TST33:  SCOPE
5339                                .REM    %
5340                                TEST ABSTRACT:
5341                                **************
5342
5343                                        THIS TEST IS SIMILAR TO TEST 32 EXCEPT THAT ALL LOCATIONS IN THE "BCR"
5344                                AND "CAR" MEMORIES ARE TESTED TO VERIFY THAT TRANSMISSION ON THE SELECTED
5345                                LINE DID NOT DISTURB ANY UNSELECTED LOCATIONS IN THE MEMORIES. IF ALSO OPERATES
5346                                IN "FLAG" MODE RATHER THAN USING INTERRUPTS. THE TEST SEQUENCE IS AS
5347                                FOLLOWS:
5348
5349                                        1.  SELECT A LINE # TO TEST (AS DEFINED BY "LINSEL:")
5350                                        2.  CLEAR BOTH THE "CAR" AND "BCR" MEMORIES
5351                                        3.  LOAD THE "BCR" MEMORY WITH ALL ONES (BYTE COUNT = -1)
5352                                        4.  ACTIVATE THE XMITTER ON THE SELECTED LINE
5353                                        5.  ACTIVATE TIMER TO WAIT FOR "XMIT DONE"
5354                                        6.  IF "XMIT DONE" FAILS TO SET ON TIME - REPORT ERROR AND
5355                                            REPEAT 1 THRU 5 UNTIL ALL SELECTED LINES TESTED
5356                                        7.  IF "XMIT DONE" SETS CHECK ALL LOCATIONS IN THE "BCR" MEMORY
5357                                            REPORT ANY UNSELECTED LINES NOT CONTAINING -1 AND THE SELECTED
5358                                            LINE IF IT DOES NOT CONTAIN 0
5359                                        8.  CHECK ALL LOCATIONS IN THE "CAR" MEMORY AND REPORT ANY UNSELECTED
5360                                            LOCATIONS NOT CONTAINING 0 AND THE SELECTED LINE IF IT DOES NOT
5361                                            CONTAIN +1.
5362                                        9.  REPEAT STEPS 1 THRU 8 UNTIL ALL SELECTED LINES TESTED.
5363
5364                                ERRORS:
5365                                *******
5366
5367                                1.  [ERROR 50] CALLED IF XMIT DONE TIMEOUT ERROR DETECTED.
5368                                2.  [ERROR 51] CALLED IF "BCR" MEMORY ERROR DETECTED
5369                                3.  [ERROR 51] CALLED IF "CAR" MEMORY ERROR DETECTED
5370
5371                                SYNC:   M7277   SH3     INIT A H        EF2
5372                                *****
5373
5374                                DEBUG:
5375                                ******
5376
5377                                1.  ASSUMING TEST 32 RAN ERROR FREE THE PROBLEM IS MOST LIKELY THE:
5378                                        M7278 MODULE IF "BCR" ERRORS
5379                                        M7277 MODULE IF "CAR" ERRORS
5380
5381                                KEY LOGIC:      (SAME AS TEST 32)
5382                                **********
5383
5384                                %
5385  010402  012767  010416  170500        MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
5386  010410  004767  014120        1$:     JSR     PC,$ELINE       ;GO SELECT A LINE TO TEST
5387  010414  000544                  BR      TST34           ;;BR IF DONE ALL SELECTED LINES
5388  010416  052711  004000        2$:     BIS     #BIT11,(R1)     ;CLEAR THE DH11
5389  010422  004767  014260              JSR     PC,CLCABC       ;GO CLEAR "CAR" AND "BCR" MEMORIES
5390  010426  004767  014316              JSR     PC,LDBCR        ;GO LOAD "BCR" MEMORY WITH ALL ONES
```

```
5391  010432  156711  017606                BISB    LINE,(R1)           ;SELECT THE LINE
5392  010436  012761  033500  000004        MOV     #33500,LPR(R1)      ;SET UP PARAMETERS
5393  010444  016761  016566  000012        MOV     LINMSK,BAR(R1)      ;ACTIVATE XMIT ON SELECTED LINE
5394
5395  010452  012767  000001  017612        MOV     #1,TIMEA            ;INIT TIMER A
5396  010460  005067  017610                CLR     TIMEB               ;INIT TIMER B
5397  010464  005711                3$:     TST     (R1)                ;XMITTR DONE YET
5398  010466  100423                BMI     4$                          ;BR IF YES
5399  010470  004767  016302                JSR     PC,TIMEIT           ;CALL THE TIMER
5400  010474  000773                BR      3$                          ;TIMER ROUTINE WILL MOVE RETURN PC
5401                                                                    ;AROUND THIS BRANCH IF TIME OUT OCCURS
5402
5403  010476  004767  016456                JSR     PC,SAPS             ;SAVE THE ERROR PSW
5404  010502  011103                MOV     (R1),R3                     ;GET THE WAS DATA
5405  010504  012704  100000                MOV     #BIT15,R4           ;SET UP S/B DATA
5406  010510  156704  017530                BISB    LINE,R4
5407  010514  010102                MOV     R1,R2                       ;MAKE REGADR = DEVADR
5408  010516  004767  013664                JSR     PC,SUER2A           ;GO SET UP ERROR INFO
5409  010522  004567  014100                JSR     R5,SUNUM            ;SET LINE NO. IN MSG
5410  010526  030244                LINE
5411  010530  034176                EM50+53
5412  010532  104050                ERROR   50                          ;TIMED OUT AWAITING XMIT DONE ON SEL LINE
5413  010534  000725                BR      1$                          ;GO TRY THE NEXT LINE
5414
5415  010536  005067  170456        4$:     CLR     $TMP7               ;INIT A LINE COUNTER
5416  010542  116711  170452        5$:     MOVB    $TMP7,(R1)          ;SELECT LINE NO. IN "SCR"
5417  010546  012704  177777                MOV     #-1,R4              ;SET UP S/B DATA
5418  010552  016103  000010                MOV     BCR(R1),R3          ;GET THE WAS BYTE COUNT
5419  010556  126767  017462  170434        CMPB    LINE,$TMP7          ;WAS THIS THE ACTIVE LINE ??
5420  010564  001001                BNE     6$                          ;BR IF NOT
5421  010566  005004                CLR     R4                          ;CHANGE S/B DATA TO 000000
5422  010570  020304                6$:     CMP     R3,R4               ;WAS BYTE COUNT CORRECT ??
5423  010572  001416                BEQ     7$                          ;BR IF YES
5424
5425  010574  005067  170402                CLR     $TMP0               ;SAVE THE ACTIVE LINE NO.
5426  010600  116767  017440  170374        MOVB    LINE,$TMP0
5427  010606  016767  170406  170370        MOV     $TMP7,$TMP1         ;SAVE THE LINE NO. BEING CHECKED
5428  010614  010102                MOV     R1,R2                       ;SET UP REGADR = BCR REG ADDR
5429  010616  062702  000010                ADD     #BCR,R2
5430  010622  004767  013642                JSR     PC,SUER4            ;GO SET UP ERROR INFO
5431  010626  104051                ERROR   51                          ;BYTE COUNT INCORRECT
5432
5433  010630  005004                7$:     CLR     R4                  ;SET UP S/B DATA
5434  010632  016103  000006                MOV     CAR(R1),R3          ;GET THE WAS DATA
5435  010636  126767  017402  170354        CMPB    LINE,$TMP7          ;IS THIS THE ACTIVE LINE
5436  010644  001001                BNE     8$                          ;BR IF NOT
5437  010646  005204                INC     R4                          ;BUMP THE CAR ADDRESS FOR ACTIVE LINE
5438  010650  020304                8$:     CMP     R3,R4               ;CAR CONTENTS CORRECT ??
5439  010652  001416                BEQ     9$                          ;BR IF YES
5440
5441  010654  005067  170322                CLR     $TMP0               ;SET UP ACT LINE NO.
5442  010660  116767  017360  170314        MOVB    LINE,$TMP0
5443  010666  016767  170326  170310        MOV     $TMP7,$TMP1         ;SAVE THE LINE NO. BEING CHECKED
5444  010674  010102                MOV     R1,R2                       ;SET UP REGADR
5445  010676  062702  000006                ADD     #CAR,R2
5446  010702  004767  013562                JSR     PC,SUER4            ;SET UP THE ERROR INFO
```

```
 5447  010706  104051                          ERROR   51              ;CAR REG INCORRECT
 5448
 5449  010710  005267  170304          9$:     INC     $TMP7           ;GENERATE NEW LINE NO.
 5450  010714  022767  000020  170276          CMP     #20,$TMP7       ;TESTED ALL LINES
 5451  010722  001707                          BEQ     5$              ;BR IF NOT
 5452  010724  000631                          BR      1$              ;GO SELECT NEXT ACTIVE LINE
```

```
                        ;;*****************************************************************
                        ;*TEST 34       TEST THAT CHARACTER AVAILABLE CAN CAUSE RCVR INTERRUPT
                        ;;*****************************************************************
010726  000004           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
```

```
5621  011234  001365                          BNE     2$              ;BR UNTIL WE'VE LOADED THE TEST COUNT
5622
5623  011236  011203                          MOV     (R2),R3         ;SET THE WAS COUNT
5624  011240  042703  140377                  BIC     #140377,R3      ;CLR JUNK BITS
5625  011244  000304                          SWAB    R4              ;SET UP S/B DATA
5626  011246  020304                          CMP     R3,R4           ;TEST COUNT = SILO COUNTER ?
5627  011250  001406                          BEQ     4$              ;BR IF YES
5628
5629  011252  004767  013124          JSR     PC,SUER2                ;GO SET UP ERROR INFO
5630  011256  012767  011176  167624          MOV     #1$,$LPERR      ;SET UP ERROR LOOP RETURN
5631  011264  104006                          ERROR   6               ;SSR FAILED TO UP-COUNT CORRECTLY
5632
5633  011266  005267  167726          4$:     INC     $TMP7           ;INCREMENT TO NEXT COUNT TO TEST
5634  011272  022767  000100  167720          CMP     #100,$TMP7      ;MAXIMUM COUNT ??
5635  011300  001336                          BNE     1$              ;BR IF NOT
5636
```

```
                                ;;*****************************************************************
                                ;*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
```

```
5693
5694  011410  011203                        MOV     (R2),R3         ;GET THE WAS DATA
5695  011412  042703  140377                BIC     #140377,R3      ;CLR JUNK BITS
5696  011416  010504                        MOV     R5,R4           ;SET UP S/B DATA
5697  011420  000304                        SWAB    R4
5698  011422  020304                        CMP     R3,R4           ;DID IT DOWN COUNT OK ??
5699  011424  001406                        BEQ     6$              ;BR IF YES
5700
5701  011426  004767  012750                JSR     PC,SUER2        ;GO SET UP ERROR INFO
5702  011432  012767  011320  167450        MOV     #1$,$LPERR      ;SET UP ERROR LOOP RETURN
5703  011440  104006                        ERROR   6               ;SILO STATUS REG. DOWN-COUNTED INCORRECTLY
5704
5705  011442  005267  167552          6$:   INC     $TMP7           ;UPDATE COUNT
5706  011446  022767  000101  167544        CMP     #101,$TMP7      ;TESTED ALL COUNTS ??
5707  011454  001321                        BNE     1$              ;BR IF NOT
5708
5709
```

```
                                ;;*****************************************************************
                                ;*TEST 37       TEST SILO ALARM LEVEL FOR  COUNTS 0,1,2,4,8,16, AND 32
                                ;;*****************************************************************
011456  000004                 T$T37:  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
```

```
5766  011512  005205                    INC     R5              ;LOAD ONE MORE THAN FILL LEVEL
5767  011514  052712  100000    2$:     BIS     #BIT15,(R2)     ;SET SILO MAINT. TO LOAD A CHAR
5768  011520  012703  001000            MOV     #1000,R3        ;INIT STALL TIMER
5769  011524  005303            3$:     DEC     R3              ;WAIT FOR SILO TO SETTLE
5770  011526  001376                    BNE     3$              ;BR TIL R3 GOES TO 000000
5771  011530  042712  100000            BIC     #BIT15,(R2)     ;CLR THE SILO MAINT BIT
5772  011534  005305                    DEC     R5              ;COUNT ONE LOADED
5773  011536  105711                    TSTB    (R1)            ;CHAR AVAIL SET YET
5774  011540  100412                    BMI     4$              ;BR IF IT IS
5775  011542  005705                    TST     R5              ;SHOULD IT BE ??
5776  011544  001363                    BNE     2$              ;BR IF NOT
5777
5778  011546  004767  015406            JSR     PC,SAPS         ;SAVE THE ERROR PSW
5779  011552  004767  000042            JSR     PC,5$           ;GO SET UP S/B DATA
5780  011556  004767  012624            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5781  011562  104006                    ERROR   6               ;SILO ALARM LEVEL FAILED AT SELECTED COUNT
5782  011564  000426                    BR      6$              ;GO CHECK NEXT COUNT
5783
5784  011566  005705            4$:     TST     R5              ;SHOULD IT HAVE BEEN SET (CHAR AVAIL)
5785  011570  001424                    BEQ     6$              ;BR IF YES
5786
5787  011572  004767  015362            JSR     PC,SAPS         ;SAVE THE ERROR PSW
5788  011576  022705  000001            CMP     #1,R5           ;IS IT OFF BY ONLY ONE ??
5789  011602  001417                    BEQ     6$              ;BR IF YES - WE'LL ALLOW HIM THIS
5790  011604  004767  000010            JSR     PC,5$           ;GO SET UP S/B DATA
5791  011610  004767  012572            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5792  011614  104006                    ERROR   6               ;SILO ALARM LEVEL FAILED
5793  011616  000411                    BR      6$              ;GO CHECK NEXT COUNT
5794
5795  011620  011203            5$:     MOV     (R2),R3         ;GET WAS DATA
5796  011622  016704  167372            MOV     $TMP7,R4        ;SET UP THE S/B DATA
5797  011626  005204                    INC     R4
5798  011630  000304                    SWAB    R4
5799  011632  105004                    CLRB    R4
5800  011634  156704  167360            BISB    $TMP7,R4
5801  011640  000207                    RTS     PC              ;RETURN TO SET UP AND REPORT ERROR
5802
5803  011642  005767  167352    6$:     TST     $TMP7           ;COUNT AT ZERO
5804  011646  001002                    BNE     7$              ;BR IF NOT
5805  011650  000261                    SEC                     ;SET THE "C" BIT
5806  011652  000401                    BR      8$              ;GO SET UP COUNT
5807  011654  000241            7$:     CLC                     ;CLEAR THE "C" BIT
5808  011656  006167  167336    8$:     ROL     $TMP7           ;SHIFT POWER OF TWO BIT
5809  011662  032767  000100  167330    BIT     #BIT6,$TMP7     ;DONE ALL POWERS ??
5810  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 "$TMP7" 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

                                %
```

```
5867  011674  012767  011724  167206          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
5868  011702  004767  012626          1$:     JSR     PC,$ELINE       ;GO SELECT A LINE TO TEST
5869  011706  000534                  BR      TST41                   ;;BR IF TESTED ALL SELECTED LINES
5870  011710  012767  002100  167302          MOV     #2100,$TMP7     ;INIT TI START WITH LOWEST SPEED
5871  011716  012767  177777  016352          MOV     #-1,TIMEC       ;INIT RELATIVE TIME CHECKER
5872  011724  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
5873  011730  156711  016310          3$:     BISB    LINE,(R1)       ;SELECT IT IN THE SCR
5874  011734  012761  177775  000010          MOV     #-3,BCR(R1)     ;SET BYTE COUNT TO XFER 3 CHARS
5875  011742  005061  000006                  CLR     CAR(R1)         ;GET TEST DATA STARTING AT LOC. 0
5876  011746  016761  167246  000004          MOV     $TMP7,LPR(R1)   ;SELECT A XMIT SPEED
5877  011754  016761  015256  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE TRANSMITTER
5878
5879  011762  012767  000001  016302          MOV     #1,TIMEA        ;INIT TIMER A
5880  011770  005067  016300                  CLR     TIMEB           ;INIT TIMER B
5881  011774  005711                  4$:     TST     (R1)            ;XMITTR DONE SET YET ?
5882  011776  100437                  BMI     5$                      ;BR IF YES
5883  012000  004767  014772          JSR     PC,TIMEIT               ;CALL THE TIMER
5884  012004  000773                  BR      4$                      ;TIMER ROUTINE WILL MOVE RETURN PC
5885                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
5886
5887  012006  016767  167206  167166          MOV     $TMP7,$TMP0     ;SAVE AND SET UP THE SPEED CODE
5888  012014  000367  167162          SWAB    $TMP0
5889  012020  006267  167156          ASR     $TMP0
5890  012024  006267  167152          ASR     $TMP0
5891  012030  042767  177760  167144          BIC     #177760,$TMP0
5892  012036  011103                  MOV     (R1),R3                 ;GET THE WAS DATA
5893  012040  042703  000200          BIC     #BIT07,R3               ;CLEAR UNINTERESTING BITS
5894  012044  010102                  MOV     R1,R2                   ;MAKE REGADR = DEVADR
5895  012046  012704  100000          MOV     #BIT15,R4               ;SET UP S/B DATA
5896  012052  156704  016166          BISB    LINE,R4
5897  012056  004767  012324          JSR     PC,SUER2A               ;GO SET UP ERROR INFO
5898  012062  004567  012540          JSR     R5,SUNUM                ;GO SET LINE NO. IN MSG
5899  012066  030244                  LINE
5900  012070  034176                  EM50+53
5901  012072  104053                  ERROR   53                      ;TIMED OUT WAITING FOR XMIT DONE
5902  012074  000426                  BR      8$                      ;GO TEST NEXT SPEED
5903
5904  012076  016703  016172          5$:     MOV     TIMEB,R3        ;GET THE WAS COUNT
5905  012102  016704  016170          MOV     TIMEC,R4                ;GET LASTR CHECK COUNT
5906  012106  020304                  CMP     R3,R4                   ;COMPARE RELATIVE TIMES
5907  012110  103420                  BLO     8$                      ;BR IF THIS SPEED FASTER THAN LAST
5908                                                                  ;SPEED TESTED
5909
5910  012112  004767  015042          7$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
5911  012116  016702  167076          MOV     $TMP7,R2                ;GET SPEED CODE AND RIGHT JUSTIFY
5912  012122  000302                  SWAB    R2
5913  012124  006202                  ASR     R2
5914  012126  006202                  ASR     R2
5915  012130  042702  177760          BIC     #177760,R2              ;STRIP AWAY ALL JUNK
5916  012134  004767  012246          JSR     PC,SUER2A               ;GO SET UP ERROR INFO
5917  012140  004567  012462          JSR     R5,SUNUM                ;GO PUT LINE NO. IN MSG
5918  012144  030244                  LINE
5919  012146  031541                  EM17+41
5920  012150  104017                  ERROR   17                      ;TRANSMITTER SPEED INCORRECT
5921
5922  012152  016767  016116  016116  8$:     MOV     TIMEB,TIMEC     ;SET UP NEW CHECK TIMER COUNT
```

```
 5923  012160  062767  002100  167032          ADD     #2100,$TMP7     ;GENERATE NEXT SPEED
 5924  012166  022767  035600  167024          CMP     #35600,$TMP7    ;DONE ALL SPEEDS ?
 5925  012174  001253                          BNE     2$              ;BR IF NOT
 5926  012176  000641                          BR      1$              ;GO TEST NEXT LINE
 5927
```

```
                                ;;*********************************************************************
                                ;*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
```

```
5984  012336  000367  166640              SWAB    $TMP0
5985  012342  042767  177760  166632      BIC     #177760,$TMP0
5986  012350  011103                      MOV     (R1),R3         ;GET THE WAS DATA
5987  012352  010102                      MOV     R1,R2           ;MAKE REGADR = DEVADR
5988  012354  012704  100200              MOV     #BIT15+BIT07,R4 ;SET UP S/B DATA
5989  012360  156704  015660              BISB    LINE,R4
5990  012364  004767  012016              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5991  012370  004567  012232              JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
5992  012374  030244                      LINE
5993  012376  032020                      EM22+51
5994  012400  104054                      ERROR   54              ;TIMED OUT WAITING FOR CHAR AVAIL
5995  012402  000426                      BR      8$              ;GO TEST NEXT SPEED
5996
5997  012404  016703  015664      5$:     MOV     TIMEB,R3        ;GET THE WAS COUNT
5998  012410  016704  015662              MOV     TIMEC,R4        ;GET THE CHECK COUNT
5999  012414  020304                      CMP     R3,R4           ;COMPARE RELATIVE TIMES
6000  012416  103420                      BLO     8$              ;BR IF TIME INDICATES THIS SPEED FASTER
6001                                                              ;THAN LAST SPEED
6002
6003  012420  004767  014534      7$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
6004  012424  016702  166570              MOV     $TMP7,R2        ;GET SPEED CODE AND RIGHT JUSTIFY
6005  012430  006302                      ASL     R2
6006  012432  006302                      ASL     R2
6007  012434  000302                      SWAB    R2
6008  012436  042702  177760              BIC     #177760,R2      ;STRIP AWAY ALL JUNK
6009  012442  004767  011740              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6010  012446  004567  012154              JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
6011  012452  030244                      LINE
6012  012454  031700                      EM20+36
6013  012456  104020                      ERROR   20              ;RECEIVER SPEED INCORRECT
6014
6015  012460  016767  015610  015610  8$: MOV     TIMEB,TIMEC     ;SET UP NEW CHECK TIMER COUNT
6016  012466  062767  002100  166524      ADD     #2100,$TMP7     ;GENERATE NEXT SPEED
6017  012474  022767  035600  166516      CMP     #35600,$TMP7    ;DONE ALL SPEEDS ?
6018  012502  001253                      BNE     2$              ;BR IF NOT
6019  012504  000641                      BR      1$              ;GO TEST NEXT LINE
6020
```

```
 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
```

```
6078                                        %
6079  012510  012767  012524  166372        MOV     #2$,$LPERR      ;SET UP ERROR RETURN
6080  012516  004767  012012                1$:     JSR     PC,$ELINE       ;GO SELECT A LINE NO. TO TEST
6081  012522  000535                        BR      TST43           ;;BR IF DONE ALL SELECTED LINES
6082  012524  012711  004000                2$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
6083  012530  010102                        MOV     R1,R2           ;MAKE REGADR = DEVADR
6084  012532  116711  015506                MOVB    LINE,(R1)       ;SELECT A LINE
6085  012536  005061  000006                CLR     CAR(R1)         ;SET UP CURRENT ADDRESS
6086  012542  012761  177677  000010        MOV     #-65.,BCR(R1)   ;SET UP BYTE COUNT
6087  012550  012761  033500  000004        MOV     #33500,LPR(R1)  ;SET UP LPR: 9600 BAUD,5 BIT CHARS
6088  012556  016761  014454  000012        MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
6089
6090  012564  012767  000001  015500        MOV     #1,TIMEA        ;INIT TIMERS
6091  012572  005067  015476                CLR     TIMEB
6092  012576  032711  040000                3$:     BIT     #BIT14,(R1)     ;STOREGE OVERFLOW YET ??
6093  012602  001024                        BNE     4$              ;BR IF YES YOU SHOULD GET IT
6094  012604  004767  014166                JSR     PC,TIMEIT       ;CALL TIMER
6095  012610  000772                        BR      3$              ;BR IF NO TIME OUT
6096
6097  012612  004767  014342                JSR     PC,SAPS         ;GO SAVE PSW
6098  012616  012704  040000                MOV     #BIT14,R4       ;SET UP S/B DATA
6099  012622  156704  015416                BISB    LINE,R4
6100  012626  011103                        MOV     (R1),R3         ;SET UP WAS DATA
6101  012630  042703  137760                BIC     #137760,R3      ;CLEAR UNINTERESTING BITS
6102  012634  004767  011546                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6103  012640  004567  011762                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
6104  012644  030244                        LINE
6105  012646  034663                        EM57+44
6106  012650  104057                        ERROR   57              ;STORAGE OVERFLOW FAILED TO SET
6107  012652  000721                        BR      1$              ;GO TRY NEXT LINE
6108
6109  012654  016167  000002  166336  4$:   MOV     NRC(R1),$TMP7   ;READ THE SILO
6110  012662  016167  000002  166330        MOV     NRC(R1),$TMP7   ;READ IT AGAIN
6111  012670  012705  001000                MOV     #1000,R5        ;INIT STALL COUNTER
6112  012674  005305                41$:    DEC     R5              ;COUNT TIMER
6113  012676  001376                        BNE     41$             ;BR IF NO TIMEOUT
6114  012700  032711  040000                BIT     #BIT14,(R1)     ;DID OVERFLOW GO AWAY ?
6115  012704  001420                        BEQ     5$              ;BR IF YES
6116
6117  012706  004767  014246                JSR     PC,SAPS         ;GO SAVE THE PSW
6118  012712  005004                        CLR     R4              ;SET UP S/B DATA
6119  012714  156704  015324                BISB    LINE,R4
6120  012720  011103                        MOV     (R1),R3         ;SET UP WAS DATA
6121  012722  042703  137760                BIC     #137760,R3      ;CLEAR UNINTERESTING BITS
6122  012726  004767  011454                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6123  012732  004567  011670                JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
6124  012736  030244                        LINE
6125  012740  034663                        EM57+44
6126  012742  104057                        ERROR   57              ;STORAGE BIT FAILED TO CLEAR
6127  012744  000664                        BR      1$              ;GO TRY NEXT LINE
6128
6129  012746  122761  000077  000017  5$:   CMPB    #77,SSR+1(R1)   ;WAS IT REALLY 65. ??
6130  012754  001660                        BEQ     1$
6131
6132  012756  004767  014176                JSR     PC,SAPS         ;GO SAVE PSW
6133  012762  012704  037400                MOV     #37400,R4       ;SET UP S/B DATA
```

```
6134  012766  062702  000016          ADD     #SSR,R2         ;SET UP REGADR
6135  012772  016103  000016          MOV     SSR(R1),R3      ;SAVE WAS DATA
6136  012776  004767  011404          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6137  013002  004567  011620          JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
6138  013006  030244                  LINE
6139  013010  034663                  EM57+44
6140  013012  104057                  ERROR   57              ;READING SILO FAILED TO DEC SSR OR
6141                                                          ;STORAGE OVFL SET AT WRONG COUNT
6142  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
```

```
6199  013026  004767  011502          1$:   JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
6200  013032  000511                        BR      TST44           ;;BR IF DONE ALL SELECTED LINES
6201  013034  012705  030260                MOV     #TDATA2,R5      ;GET POINTER TO DATA TABLE
6202  013040  005002                        CLR     R2              ;INIT R2 TO START AT CHAR LENGTH OF 5 BITS
6203  013042  012567  166152          2$:   MOV     (R5)+,$TMP7     ;PUT TEST CHAR IN XMIT BUFFER
6204  013046  012711  004000          3$:   MOV     #BIT11,(R1)     ;CLEAR THE DH11
6205  013052  156711  015166                BISB    LINE,(R1)       ;SELECT THE LINE
6206  013056  012761  177777  000010        MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO -1
6207  013064  012761  001220  000006        MOV     #$TMP7,CAR(R1)  ;SET CURRENT ADDRESS REG
6208  013072  012761  033500  000004        MOV     #33500,LPR(R1)  ;SET BAUD RATE TO 9600
6209  013100  050261  000004                BIS     R2,LPR(R1)      ;SELECT CHAR LENGTH
6210  013104  056761  014126  000012        BIS     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
6211
6212  013112  012767  000001  015152        MOV     #1,TIMEA        ;INIT TIMER A
6213  013120  005067  015150                CLR     TIMEB           ;INIT TIMER B
6214  013124  105711                  4$:   TSTB    (R1)            ;RCVR DONE YET ??
6215  013126  100424                        BMI     5$              ;BR IF YES
6216  013130  004767  013642                JSR     PC,TIMEIT       ;CALL THE TIMER
6217  013134  000773                        BR      4$              ;TIMER ROUTINE WILL MOVE RETURN PC
6218                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS
6219
6220  013136  004767  014016                JSR     PC,SAPS         ;SAVE THE ERROR PSW
6221  013142  011103                        MOV     (R1),R3         ;GET THE SCR
6222  013144  042703  177560                BIC     #177560,R3      ;CLEAR UNINTERESTING BITS
6223  013150  012704  000200                MOV     #200,R4         ;SET UP S/B DATA
6224  013154  156704  015064                BISB    LINE,R4
6225  013160  004767  011222                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6226  013164  004567  011436                JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
6227  013170  030244                        LINE
6228  013172  032020                        EM22+51
6229  013174  104055                        ERROR   55              ;CHAR AVAIL FAILED TO SET ON TIME
6230  013176  000422                        BR      6$              ;GO TEST NEXT CHAR LENGTH
6231
6232  013200  016103  000002          5$:   MOV     NRC(R1),R3      ;GET THE WAS DATA
6233  013204  012704  000200                MOV     #200,R4         ;SET UP THE S/B DATA IN R4
6234  013210  156704  015030                BISB    LINE,R4
6235  013214  000304                        SWAB    R4
6236  013216  156704  165776                BISB    $TMP7,R4
6237  013222  020304                        CMP     R3,R4           ;WAS THE RCVD DATA CORRECT ??
6238  013224  001407                        BEQ     6$              ;BR IF YES
6239
6240  013226  004767  011150                JSR     PC,SUER2        ;GO SET UP THE ERROR INFO
6241  013232  004567  011370                JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
6242  013236  030244                        LINE
6243  013240  032061                        EM23+36
6244  013242  104023                        ERROR   23              ;DATA COMPARE ERROR
6245
6246  013244  005202          6$:   INC     R2              ;DO NEXT CHAR LENGTH ON SELECTED LINE
6247  013246  022702  000004                CMP     #4,R2           ;HAVE WE DONE ALL FOUR CHAR LENGTHS ??
6248  013252  001273                        BNE     2$              ;BR IF NOT
6249  013254  000664                        BR      1$              ;GO DO NEXT LINE
```

```
6250                                    ;;*****************************************************************
6251                                    ;*TEST 44       SINGLE LINE DATA TEST - ALL SELECTED LINES
6252                                    ;;*****************************************************************
6253  013256  000004                    T$T44:  SCOPE
6254                                    .REM    %
6255                                    TEST ABSTRACT:
6256                                    **************
6257
6258                                            THIS TEST TRANSMITS AND RECEIVES A BINARY COUNT PATTERN
6259                                    (000 - 377) ON ALL SELECTED LINES. THE TEST SEQUENCE IS AS
6260                                    FOLLOWS:
6261
6262                                            1. SET UP THE ERROR LOOP RETURN
6263                                            2. GO SELECT A LINE NO. TO TEST - IF DONE ALL SELECTED
6264                                               LINES THEN GO TO THE NEXT TEST .
6265                                            3. CLEAR THE DH11 AND PRIME THE SELECTED LINE TO XMIT
6266                                               TO XMIT 256. CHARS AT 9600. BAUD - 8 BIT CHARS.
6267                                            4. SET UP R5 TO POINT TO RCVR CORE BUFFER.
6268                                            5. ACTIVATE THE SELECTED XMITTER.
6269                                            6. WAIT FOR "CHAR AVAIL" TO SET BEFORE READING THE SILO.
6270                                               IF RCVR TIMEOUT REPORT ERROR AND RESTART AT STEP 2.
6271                                            7. IF NO TIMEOUT READ THE SILO AND STORE THE WORD IN THE RCVR
6272                                               CORE BUFFER - WHEN THE BUFFER IS FULL GO TO STEP8 IF NOT
6273                                               THEN GO TO STEP 6.
6274                                            8. COMPARE THE XMIT AND RCVR CORE IMAGE BUFFERS AND REPORT
6275                                               ALL DATA COMPARE ERRORS.
6276                                            9. CHECK THE "BAR", "BCR", AND "CAR" REGISTERS FOR CORRECT
6277                                               CONTENTS - REPORT ALL ERRORS.
6278                                            10.GO TO STEP 2
6279
6280
6281                                    ERRORS:
6282                                    *******
6283
6284                                    1. [ERROR  22] IS CALLED TO REPORT "DATA AVAIL" TIMEOUT
6285                                    2. [ERROR  37] "          "          " DATA COMPARE ERRORS
6286                                    3. [ERROR  40] "          "          " "BAR" REG NOT CLEARED
6287                                    4. [ERROR  10] "          "          " "BCR" REG NOT ALL ZEROES
6288                                    5. [ERROR   7] "          "          " "CAR" REG NOT UPDATED CORRECTLY
6289
6290                                    SYNC:   M7277   SH3     INIT A H        EF2
6291                                    *****
6292
6293                                    DEBUG:
6294                                    *******
6295
6296                                    1. IF THE FAULT AFFECTS ONE OR MORE LINES IN AN 8 LINE GROUP <15:08>
6297                                       OR <07:00>, SWAP THE M7280 MODULES. IF THE FAULT SHIFTS SO THAT
6298                                       THE ERROR INDICATES DIFFERENT LINES THE PROBLEM IS MOST LIKELY
6299                                       THE M7280 THE SYMPTOM SHIFTED TO.
6300
6301                                    2. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS ON
6302                                       ONE LINE THE FAULT IS MOST LIKELY A "UART" CHIP.
6303
6304                                    3. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS
6305                                       ON ALL LINES SUSPECT THE DATA PATHS EXTERNAL TO, THE M7280 MODULES.
```

```
6306
6307                                    4. IF THE FAULT CAUSES NO DATA ERRORS BUT GIVES "BAR", "BCR", OR "CAR" ERRORS
6308                                       SUSPECT THE M7278 OR M7277 MODULES.
6309
6310                                    KEY LOGIC       (REFER TO TEST 43)
6311                                    ***********
6312
6313                                    %
6314  013260  012767  013302  165622            MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
6315  013266  004767  011242            1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
6316  013272  000401                            BR      11$             ;;BR IF ALL SELECTED LINES DONE
6317  013274  000402                            BR      2$              ;GO TEST IT
6318  013276  000167  000424            11$:    JMP     @$              ;EXIT TEST
6319  013302  016701  013716            2$:     MOV     DHADR,R1        ;RESET DEVADR
6320  013306  012711  004000                    MOV     #BIT11,(R1)     ;CLEAR THE DH11
6321  013312  156711  014726                    BISB    LINE,(R1)       ;SET SELECT BITS IN SCR
6322  013316  012761  037244  000006            MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDRESS REG
6323  013324  012761  177400  000010            MOV     #-400,BCR(R1)   ;SET UP BYTE COUNT
6324  013332  012761  033503  000004            MOV     #33503,LPR(R1)  ;SET LINE PARAMETERS
6325  013340  012705  036244                    MOV     #RBUF,R5        ;SET UP POINTER TO INPUT DATA BUFFER
6326  013344  052711  001000                    BIS     #BIT09,(R1)     ;SET MAINT MODE BIT
6327  013350  016761  013662  000012            MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
6328
6329  013356  012767  000002  014706            MOV     #2,TIMEA        ;INIT TIMER A
6330  013364  005067  014704                    CLR     TIMEB           ;INIT TIMER B
6331  013370  105711                    3$:     TSTB    (R1)            ;RCVR DONE YET ??
6332  013372  100425                            BMI     4$              ;BR IF YES
6333  013374  004767  013376                    JSR     PC,TIMEIT       ;CALL THE TIMER
6334  013400  000773                            BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
6335                                                                    ;AROUND THIS BRANCH IF TIME OUT OCCURS
6336
6337  013402  004767  013552                    JSR     PC,SAPS         ;SAVE THE ERROR PSW
6338  013406  010102                            MOV     R1,R2           ;SET UP REGADR
6339  013410  011203                            MOV     (R2),R3         ;GET THE WAS DATA
6340  013412  042703  176400                    BIC     #176400,R3      ;CLEAR JUNK BITS
6341  013416  012704  001200                    MOV     #1200,R4        ;SET UP S/B DATA
6342  013422  156704  014616                    BISB    LINE,R4
6343  013426  004767  010754                    JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6344  013432  004567  011170                    JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
6345  013436  030244                            LINE
6346  013440  032020                            EM22+51
6347  013442  104022                            ERROR   22              ;CHAR AVAIL TIMEOUT
6348  013444  000710                            BR      1$              ;GO TRY NEXT LINE
6349
6350  013446  016125  000002            4$:     MOV     NRC(R1),(R5)+   ;SAVE THE RECEIVED DATA
6351  013452  022705  037244                    CMP     #RBUF+1000,R5   ;INPUT BUFFER FULL ??
6352  013456  001344                            BNE     3$              ;BR IF NOT
6353
6354  013460  012702  037244                    MOV     #TBUF,R2        ;SET UP POINTER TO OUTPUT BUFFER
6355  013464  012701  036244                    MOV     #RBUF,R1        ;SET UP POINTER TO INPUT BUFFER
6356  013470  111204                    5$:     MOVB    (R2),R4         ;SET UP S/B DATA IN R4
6357  013472  042704  177400                    BIC     #177400,R4
6358  013476  000304                            SWAB    R4
6359  013500  156704  014540                    BISB    LINE,R4
6360  013504  152704  000200                    BISB    #200,R4
6361  013510  000304                            SWAB    R4
```

```
6362  013512  011103                        MOV     (R1),R3         ;GET THE WAS DATA
6363  013514  020304                        CMP     R3,R4           ;DATA CORRECT ??
6364  013516  001407                        BEQ     6$              ;BR IF YES
6365
6366  013520  004767  010656                JSR     PC,SUER2        ;GO SET UP ERROR INFO
6367  013524  004567  011076                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
6368  013530  030244                        LINE
6369  013532  033276                        EM37+33
6370  013534  104037                        ERROR   37              ;DATA COMPARE ERROR
6371
6372
6373  013536  005202                6$:     INC     R2              ;UPDATE DATA BUFFER POINTERS
6374  013540  062701  000002                ADD     #2,R1           ;
6375  013544  022701  037244                CMP     #RBUF+1000,R1   ;COMPARED ALL 256. CHARS ??
6376  013550  001347                        BNE     5$              ;BR IF NOT
6377
6378  013552  016701  013446                MOV     DHADR,R1        ;RESET DEVADR
6379  013556  010102                        MOV     R1,R2           ;SET UP REGADR
6380  013560  062702  000012                ADD     #BAR,R2
6381  013564  005712                        TST     (R2)            ;WAS THE "BAR" ALL ZEROES ??
6382  013566  001413                        BEQ     7$              ;BR IF YES
6383
6384  013570  004767  013364                JSR     PC,SAPS         ;SAVE THE ERROR PSW
6385  013574  011203                        MOV     (R2),R3         ;GET THE WAS DATA
6386  013576  005004                        CLR     R4              ;SET UP S/B DATA
6387  013600  004767  010602                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6388  013604  004567  011016                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
6389  013610  030244                        LINE
6390  013612  033341                        EM40+40
6391  013614  104040                        ERROR   40              ;"BAR" REG NOT ALL ZEROES
6392
6393  013616  010102                7$:     MOV     R1,R2           ;SET UP REGADR
6394  013620  062702  000010                ADD     #BCR,R2
6395  013624  005712                        TST     (R2)            ;BYTE COUNT REG ALL ZEROES ?
6396  013626  001413                        BEQ     71$             ;;BR IF BYTE COUNT ZERO
6397
6398  013630  004767  013324                JSR     PC,SAPS         ;SAVE THE ERROR PSW
6399  013634  011203                        MOV     (R2),R3         ;GET THE WAS DATA
6400  013636  005004                        CLR     R4              ;SET UP THE S/B DATA
6401  013640  004767  010542                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6402  013644  004567  010756                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
6403  013650  030244                        LINE
6404  013652  031146                        EM10+44
6405  013654  104010                        ERROR   10              ;BYTE COUNT NOT ALL ZEROES
6406
6407  013656  010102                71$:    MOV     R1,R2           ;SET UP REGADR
6408  013660  062702  000006                ADD     #CAR,R2
6409  013664  022712  037644                CMP     #TBUF+400,(R2)  ;DID "CAR" INCREMENT PROPERLY ?
6410  013670  001414                        BEQ     72$             ;BR IF YES
6411
6412  013672  004767  013262                JSR     PC,SAPS         ;SAVE THE ERROR PSW
6413  013676  011203                        MOV     (R2),R3         ;GET THE WAS DATA
6414  013700  012704  037644                MOV     #TBUF+400,R4    ;SET UP S/B DATA
6415  013704  004767  010476                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6416  013710  004567  010712                JSR     R5,SUNUM        ;GO PUT LINE NO IN MESSAGE
6417  013714  030244                        LINE
```

```
6418  013716  031077                        EM7+47
6419  013720  104007                        ERROR   7               ;"CAR" NOT UPDATED CORRECTLY
6420
6421  013722  000167  177340        72$:    JMP     1$              ;GO DO NEXT LINE
6422
6423  013726  000240                8$:     NOP                     ;EXIT POINT
```

```
                                ;;*****************************************************************
                                ;*TEST 45       BASIC PARITY LOGIC TEST - ALL SELECTED LINES - ODD PARITY
                                ;;*****************************************************************
013730  000004                  T$T45:  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$:     JSR     PC,$ELINE       ;GO SELECT A LINE NO.
013744  000506                  BR      T$T46           ;;BR IF ALL SELECTED LINES DONE
013746  012711  004000          2$:     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          4$:   MOV     NRC(R1),R3 ;GET THE WAS DATA
014124  020304                        CMP     R3,R4     ;CORRECT DATA RECEIVED ??
014126  001704                        BEQ     1$        ;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                  ;ST46:  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
```

```
6563  014322  016761  012706  000012          MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL SELECTEDLINES
6564  014330  116767  164546  164650          MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
6565  014336  042767  177400  164642          BIC     #177400,$TMP2
6566  014344  004767  012560                  JSR     PC,CHPS1        ;GO CLEAR PSW
6567  014350  000167  000236                  JMP     7$              ;GO WAIT FOR INTERRUPTS
6568
6569  014354  012706  001100          21$:    MOV     #STACK,SP       ;RESTORE THE SP
6570  014360  004767  012544                  JSR     PC,CHPS1        ;GO CLEAR PSW
6571  014364  004767  012360                  JSR     PC,RESTRP       ;RESTORE TRAP CATCHER
6572  014370  000556                          BR      TST47           ;;GO TO NEXT TEST
6573
6574                          ;RECEIVER INTERRUPT SERVICE ROUTINE
6575
6576  014372  005067  164614          3$:     CLR     $TMP4
6577  014376  105711                          TSTB    (R1)            ;CHAR AVAIL SET
6578  014400  100404                          BMI     4$              ;BR IF YES
6579
6580  014402  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
6581  014406  104041                          ERROR   41              ;RCVR FALSE INTERRUPT - CHAR AVAIL NOT SET
6582  014410  000700                          BR      2$              ;GO TRY NEXT SUB TEST
6583
6584  014412  032711  040000          4$:     BIT     #BIT14,(R1)     ;SILO OVERFLOW ??
6585  014416  001404                          BEQ     5$              ;BR IF NOT
6586
6587  014420  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
6588  014424  104042                          ERROR   42              ;SILO OVERFLOW ERROR
6589  014426  000671                          BR      2$              ;GO TRY NEXT SUB TEST
6590
6591  014430  016103  000002          5$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
6592  014434  010302                          MOV     R3,R2           ;EXTRACT AND SAVE LINE NO.
6593  014436  000302                          SWAB    R2
6594  014440  042702  177760                  BIC     #177760,R2
6595  014444  006302                          ASL     R2              ;GENERATE TABLE OFFSETR
6596  014446  005267  164540                  INC     $TMP4
6597  014452  022767  000101  164532          CMP     #101,$TMP4
6598  014460  001744                          BEQ     3$
6599  014462  036267  030170  012544          BIT     $LNSEL(R2),LINSEL       ;IS THIS ONE OF THE SELECTED LINES?
6600  014470  001002                          BNE     51$             ;IF SO,GO ANALYZE THE CHARACTER
6601  014472  104000                          ERROR                   ;INDICATE SOME KIND OF ERROR
6602  014474  000755                          BR      5$              ;CHECK THE NEXT SILO ENTRY
6603  014476  010267  164510          51$:    MOV     R2,$TMP4
6604  014502  006267  164504                  ASR     $TMP4
6605  014506  026203  036244                  CMP     RBUF(R2),R3     ;CORRECT DATA RECEIVED ??
6606  014512  001426                          BEQ     6$              ;BR IF YES
6607
6608  014514  004767  012440                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
6609  014520  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
6610  014524  016204  036244                  MOV     RBUF(R2),R4     ;SET UP S/B DATA
6611  014530  062701  000002                  ADD     #NRC,R1         ;SET UP WAS ADDRESS
6612  014534  062702  036244                  ADD     #RBUF,R2        ;SET UP S/B ADDRESS
6613  014540  004767  007642                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6614  014544  004567  010056                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
6615  014550  001212                          $TMP4
6616  014552  033066                          EM34+51
6617  014554  004567  010046                  JSR     R5,SUNUM        ;PUT SUBTEST NO. IN MESSAGE
6618  014560  001220                          $TMP7
```

```
6619  014562  033104                          EM34+67
6620  014564  104034                          ERROR   34              ;PARITY DATA COMPARE ERROR
6621  014566  000611                          BR      2$              ;GO TRY NEXT SUBTEST
6622
6623  014570  105262  036244          6$:     INCB    RBUF(R2)        ;GENERATE NEW RCVD DATA
6624  014574  005262  027504                  INC     MULPTB(R2)      ;COUNT ONE BYTES RECEIVED
6625  014600  001003                          BNE     61$             ;BR IF NOT DONE
6626  014602  046267  027442  012672          BIC     LINBIT(R2),LINACT ;FLAG THIS LINE DONE
6627  014610  000002                  61$:    RTI                     ;RETURN TO WAIT ROUTINE
6628
6629                                  ;WAIT ROUTINE
6630
6631  014612  012767  000002  013452  7$:     MOV     #2,TIMEA        ;INIT TIMER A
6632  014620  005067  013450                  CLR     TIMEB           ;INIT TIMER B
6633  014624  005761  000012          8$:     TST     BAR(R1)         ;ALL LINES DONE XMITTING ??
6634  014630  001413                          BEQ     9$              ;BR IF YES
6635  014632  004767  012140                  JSR     PC,TIMEIT       ;CALL THE TIMER
6636  014636  000772                          BR      8$              ;TIMER ROUTINE WILL MOVE RETURN PC
6637                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
6638
6639  014640  016167  000012  164342          MOV     BAR(R1),$TMP3   ;SAVE THE ACTIVE LINES FLAG
6640  014646  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
6641  014652  104035                          ERROR   35              ;TIMED OUT WAITING FOR TRANSMITTERS TO FINISH
6642  014654  000167  177332                  JMP     2$              ;GO TRY NEXT SUBTEST
6643
6644
6645  014660  012767  000001  013404  9$:     MOV     #1,TIMEA        ;INIT TIMER A
6646  014666  005067  013402                  CLR     TIMEB           ;INIT TIMER B
6647  014672  005767  012604          10$:    TST     LINACT          ;ALL CHARS RECEIVED ?
6648  014676  001411                          BEQ     11$             ;BR IF YES
6649  014700  004767  012072                  JSR     PC,TIMEIT       ;CALL THE TIMER
6650  014704  000772                          BR      10$             ;TIMER ROUTINE WILL MOVE RETURN PC
6651                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
6652
6653  014706  016767  012570  164274          MOV     LINACT,$TMP3    ;SET UP ACTIVE LINE PARAMETER
6654  014714  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
6655  014720  104035                          ERROR   35              ;SILO EMPTY TIMEOUT
6656  014722  000167  177264          11$:    JMP     2$              ;GO TRY NEXT SUB TEST
```

```
6657                                    ;;*****************************************************************
6658                                    ;*TEST 47       AUTO ECHO TEST 1 - ALL SELECTED LINES
6659                                    ;;*****************************************************************
6660  014726  000004                    TST47:  SCOPE
6661                                    .REM    %
6662                                    TEST ABSTRACT:
6663                                    **************
6664
6665                                            THIS TEST VERIFIES THAT ALL SELECTED LINES CAN'TURN AROUND
6666                                    A SINGLE TEST CHARACTER (377) AND (000) IN AUTO ECHO MODE. THE TEST SEQUENCE
6667                                    IS AS FOLLOWS:
6668
6669                                            1. SET UP THE ERROR LOOP RETURN
6670                                            2. RETRIEVE THE AUTO-ECHO TEST DATA FROM "AETAB:" AND
6671                                               UPDATE THE POINTER.
6672                                            3. GO SELECT A LINE NO. TO TEST - GO TO STEP 10 IF DONE
6673                                               ALL SELECTED LINES.
6674                                            4. CLEAR THE "CAR" AND "BCR" MEMORIES.
6675                                            5. PRIME THE SELECTED LINE TO XMIT ONE CHAR WITH A.E. ENABLED.
6676                                            6. ACTIVATE THE SELECTED TRANSMITTER.
6677                                            7. WAIT FOR "CHAR AVAIL" - IF TIMEOUT REPORT ERROR AND RESTART
6678                                               AT STEP 2.
6679                                            8. IF NO TIMEOUT - READ SILO AND COMPARE AUTO ECHO DATA RECEIVED
6680                                               REPORT DATA COMPARE ERRORS AND RESTART AT STEP 2
6681                                            9. IF NO ERRORS REPEAT STEPS 7 AND 8 SIXTY-FOUR TIMES THEN TURN
6682                                               OFF A.E. ENABLE AND READ LAST CHAR FROM SILO.. CHECK LAST
6683                                               CHAR FOR DATA COMPARE ERRORS - REPORT ERRORS IF ANY - AND RE-
6684                                               START AT  TEP 2.
6685                                           10. CHANGE A.E. TABLE POINTER TO POINT TO "AETABO:" (O'S DATA)
6686                                               AND REPEAT STEPS 2 THRU 9.
6687
6688                                    ERRORS:
6689                                    ********
6690
6691                                    1. [ERROR 24] IS CALLED TO REPORT ALL ERRORS
6692
6693                                    SYNC:   M7277   SH4     LOAD BAR LB+HB L        CN2
6694                                    *****
6695
6696                                    DEBUG:
6697                                    ******
6698
6699                                    1. IF ALL LINES FAIL, SUSPECT EITHER THE M7277 OR M7289
6700
6701                                    2. IF ONLY ONE LINE FAILS SUSPECT THE M7288
6702
6703                                    3. LOOP ON THE FAILING LINE AND TRACK BACK THROUGH THE KEY LOGIC.
6704
6705
6706                                    KEY LOGIC:
6707                                    **********
6708
6709                                            M7277   SH3     AE GO L         CS2
6710                                                            7402 "OR" GATE CHIPS E38 OR E41
6711                                                            74157 MUX CHIPS  E39 OR 342
6712                                                    SH4     E35 - PIN 2 STUCK LOW
```

```
6713
6714                                        M7289   SH3     AE GO L          EK1
6715                                                        AE SCAN MUX  E22 PIN 10
6716                                                SH4     SAMPLE STATUS H E21-12
6717
6718                                        M7288   SH5, 7, 9, 11
6719                                                        AE ENABLE "NN" H CONTROL FLOPS
6720                                                        74174 CHIPS PIN 15
6721
6722                                    %
6723  014730  012767  015012  164152          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
6724  014736  005067  164246                  CLR     $TMP3           ;INIT I/O DATA FLAG
6725  014742  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE DH11
6726  014746  012705  027342                  MOV     #AETAB,R5       ;GET POINTER TO AUTO ECHO DATA TABLE
6727  014752  005067  012262          7$:     CLR     LMSK1           ;INIT BIT TEST MARKER
6728  014756  000261                          SEC                     ;SET "C" BIT FOR MARKER
6729  014760  000401                          BR      13$             ;GO SHIFT MASK
6730  014762  000241                  1$:     CLC                     ;INIT THE "C" BIT
6731  014764  006167  012250          13$:    ROL     LMSK1           ;SHIFT BIT MARKER
6732  014770  001407                          BEQ     12$             ;BR IF DONE ALL LINES
6733  014772  012504                          MOV     (R5)+,R4        ;SET UP THE S/B DATA
6734  014774  036767  012240  012232          BIT     LMSK1,LINSEL    ;TEST THIS LINE ?
6735  015002  001767                          BEQ     1$              ;BR IF NOT
6736  015004  004767  007524          11$:    JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
6737  015010                          12$:
6738  015010  000522                          BR      6$              ;;BR IF ALL SELECTED LINES TESTED
6739  015012  004767  007670          2$:     JSR     PC,CLCABC       ;GO CLEAR "CAR" AND "BCR" MEMORIES
6740  015016  116711  013222                  MOVB    LINE,(R1)       ;SET SELECT BITS IN SCR REG
6741  015022  012761  177777  000010          MOV     #-1,BCR(R1)     ;SET UP TO XFER ONE CHAR
6742  015030  010561  000006                  MOV     R5,CAR(R1)      ;SET UP THE BUS ADDRESS REG
6743  015034  162761  000002  000006          SUB     #2,CAR(R1)      ;CORRECT BUS ADDRESS
6744  015042  012767  000100  164150          MOV     #100,$TMP7      ;COUNT 64 CHARS TO BE RECEIVED IN AUTO ECHO
6745  015050  005067  164142                  CLR     $TMP6           ;INIT CHAR COUNTER
6746  015054  012761  133503  000004          MOV     #133503,LPR(R1) ;SET UP LINE PARAMETER REG
6747  015062  016761  012150  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE LINE
6748
6749  015070  012767  000002  013174          MOV     #2,TIMEA        ;INIT TIMER A
6750  015076  005067  013172                  CLR     TIMEB           ;INIT TIMERB
6751  015102  105711                  3$:     TSTB    (R1)            ;CHAR AVAIL SET ??
6752  015104  100427                          BMI     4$              ;BR IF YES
6753  015106  004767  011664                  JSR     PC,TIMEIT       ;CALL THE TIMER
6754  015112  000773                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
6755                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
6756
6757  015114  004767  012040                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
6758  015120  005061  000004                  CLR     LPR(R1)         ;TURN OFF AUTO ECHO MODE
6759  015124  010102                          MOV     R1,R2           ;MAKE REGADR = DEVADR
6760  015126  011103                          MOV     (R1),R3         ;GET THE WAS DATA
6761  015130  042703  100000                  BIC     #BIT15,R3       ;CLEAR JUNK BIT
6762  015134  012704  000200                  MOV     #200,R4         ;SET UP S/B DATA
6763  015140  156704  013100                  BISB    LINE,R4
6764  015144  004767  007236                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6765  015150  004567  007452                  JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
6766  015154  030244                          LINE
6767  015156  032216                          EM24+35
6768  015160  104024                          ERROR   24              ;DATA AVAIL FAILED TO SET ON TIME
```

```
6769  015162  000677                          BR      1$              ;GO TRY NEXT LINE
6770
6771  015164  005267  164026          4$:     INC     $TMP6           ;COUNT ONE CHAR RECVD
6772  015170  016103  000002                  MOV     NRC(R1),R3      ;GET THE WAS DATA
6773  015174  020304                          CMP     R3,R4           ;WAS CHAR AUTO ECHOED CORRECTLY ?
6774  015176  001417                          BEQ     5$              ;BR IF YES
6775
6776  015200  004767  011754                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
6777  015204  005061  000004                  CLR     LPR(R1)         ;DISABLE AUTO ECHO
6778  015210  010102                          MOV     R1,R2           ;SET UP REGADR
6779  015212  062702  000002                  ADD     #NRC,R2
6780  015216  004767  007164                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6781  015222  004567  007400                  JSR     R5,SUNUM        ;PUT LINE NO. IN ERROR MSG
6782  015226  030244                          LINE
6783  015230  032216                          EM24+35
6784  015232  104024                          ERROR   24              ;CHAR AUTO ECHOED INCORRECTLY
6785  015234  000652                          BR      1$              ;GO TRY NEXT LINE
6786
6787  015236  005367  163756          5$:     DEC     $TMP7           ;COUNT ONE CHAR READ OUT OF 64
6788  015242  003317                          BGT     3$              ;BR IF NOT LAST ONE
6789  015244  100646                          BMI     1$              ;;BR IF LAST ONE READ
6790  015246  042761  100000  000004          BIC     #BIT15,LPR(R1)  ;DISABLE AUTO ECHO
6791  015254  000712                          BR      3$              ;GO READ LAST CHAR
6792
6793  015256  005167  163726          6$:     COM     $TMP3           ;TOGGLE I/O FLAG
6794  015262  001406                          BEQ     TST50           ;;BR IF DONE BOTH I/O DATA
6795  015264  005067  012754                  CLR     LINE            ;INIT LINE NO TO 00
6796  015270  012705  027402                  MOV     #AETAB0,R5      ;SET POINTER TO 0'S TABLE
6797  015274  000167  177452                  JMP     7$              ;REPEAT TEST FOR ZERO PATTERNS
6798
```

```
                                ;;*****************************************************************
                                ;*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     #AE†AB,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
```

```
6855  015416  012761  133503  000004          MOV     #133503,LPR(R1) ;DO IT AT 9600 BAUD/8 BITS
6856  015424  116767  163452  163554          MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
6857  015432  042767  177400  163546          BIC     #177400,$TMP2
6858  015440  046767  011572  012034          BIC     LINMSK,LINACT   ;MAKE THIS LINE APPEAR INACTIVE
6859  015446  016761  011562  000012          MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL SELECTED TRANSMITTERS
6860
6861  015454  012767  000002  012610  21$:    MOV     #2,TIMEA        ;INIT TIMER A
6862  015462  005067  012606                  CLR     TIMEB           ;INIT TIMER B
6863  015466  016103  000002          3$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
6864  015472  100414                  BMI     4$              ;BR IF YES
6865  015474  004767  011276                  JSR     PC,TIMEIT       ;CALL THE TIMER
6866  015500  000772                  BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
6867                                                          ;AROUND THIS BRANCH IF TIME OUT OCCURS
6868
6869  015502  016167  000004  163472          MOV     LPR(R1),$TMP0   ;SAVE THE CURRENT "LPR"
6870  015510  004567  007112                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
6871  015514  030244                  LINE
6872  015516  032707                  EM32+35
6873  015520  104032                  ERROR   32              ;AUTO ECHO TIMEOUT
6874  015522  000700                  BR      1$              ;GO TRY NEXT LINE
6875
6876  015524  010304          4$:     MOV     R3,R4           ;EXTRACT LINE NUMBER OF RCVD CHAR
6877  015526  000304                  SWAB    R4
6878  015530  042704  177760                  BIC     #177760,R4
6879  015534  010402                  MOV     R4,R2           ;SAVE IT IN R2
6880  015536  006302                  ASL     R2              ;GENERATE TABLE INDEX IN R2
6881  015540  126704  012500                  CMPB    LINE,R4         ;IS THIS THE A.E. TEST LINE ??
6882  015544  001432                  BEQ     5$              ;BRANCH IF YES.
6883  015546  036267  030170  011460          BIT     $LNSEL(R2),LINSEL
6884  015554  001737                  BEQ     21$
6885
6886  015556  026203  036244                  CMP     RBUF(R2),R3     ;RECVD DATA CORRECT ??
6887  015562  001447                  BEQ     6$              ;BR IF IT WAS
6888
6889  015564  004767  011370                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
6890  015570  010467  163420                  MOV     R4,$TMP5        ;SAVE THE LINE NUMBER
6891  015574  016204  036244                  MOV     RBUF(R2),R4     ;SET UP S/B DATA
6892  015600  062702  036244                  ADD     #RBUF,R2        ;SET UP S/B ADDRESS
6893  015604  012701  177703                  MOV     #177703,R1      ;SET UP THE WAS ADDRESS
6894  015610  004767  006572                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6895  015614  004567  007006                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
6896  015620  001214                  $TMP5
6897  015622  032552                  EM31+45
6898  015624  104031                  ERROR   31              ;NON-ECHO DATA COMPARE ERROR
6899  015626  000167  177472                  JMP     1$              ;GO TRY NEXT LINE
6900
6901
6902  015632  020367  163362          5$:     CMP     R3,$TMP7        ;CHAR ECHOED OK ??
6903  015636  001427                  BEQ     7$              ;BR IF YES
6904
6905  015640  004767  011314                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
6906  015644  012702  001220                  MOV     #$TMP7,R2       ;SAVE THE S/B ADDRESS
6907  015650  016704  163344                  MOV     $TMP7,R4        ;SAVE THE S/B DATA
6908  015654  012701  177703                  MOV     #177703,R1      ;SAVE THE WAS ADDRESS
6909  015660  004767  006522                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
6910  015664  004567  006736                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MESSAGE
```

```
 6911  015670  030244                          LINE
 6912  015672  032552                          EM31+45
 6913  015674  104031                          ERROR   31              ;AUTO ECHO LINE DATA ERROR
 6914  015676  000167  177422                  JMP     1$              ;GO TRY NEXT LINE
 6915
 6916  015702  105262  036244          6$:     INCB    RBUF(R2)        ;GENERATE NEXT EXPECTED DATA ON THIS LINE
 6917  015706  001262                          BNE     21$             ;BR IF ITS NOT BACK TO 000
 6918  015710  046267  027442  011564          BIC     LINBIT(R2),LINACT       ;INDICATE THIOS LINE DONE 256 BYTES
 6919  015716  005767  011560          7$:     TST     LINACT          ;ALL LINES INACTIVE
 6920  015722  001254                          BNE     21$             ;BR IF NOT
 6921  015724  042761  100000  000004          BIC     #BIT15,LPR(R1)  ;TURN OFF THE A.E. BIT
 6922  015732  105761  000017                  TSTB    SSR+1(R1)       ;SILO EMPTY ??
 6923  015736  001002                          BNE     8$              ;BR IF NOT
 6924  015740  000167  177360                  JMP     1$              ;GO TEST NEXT LINE
 6925  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
```

```
6982  016112  000771                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
6983                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
6984
6985  016114  016167  000004  163060          MOV     LPR(R1),$TMP0   ;SAVE THE "LPR" REG
6986  016122  104036                          ERROR   36              ;DATA AVAILABLE TIMEOUT
6987  016124  000467                          BR      TST52           ;;EXIT TEST ON ERROR
6988
6989  016126  016103  000002          4$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
6990  016132  010302                          MOV     R3,R2           ;BUILD AND SAVE LINE NO.
6991  016134  000302                          SWAB    R2
6992  016136  042702  177760                  BIC     #177760,R2
6993  016142  010267  163050                  MOV     R2,$TMP6        ;SAVE THE LINE NO.
6994  016146  006302                          ASL     R2              ;GENERATE TABLE OFFSET
6995  016150  005267  163036                  INC     $TMP4
6996  016154  022767  000101  163030          CMP     #101,$TMP4
6997  016162  001745                          BEQ     3$
6998  016164  036267  030170  011042          BIT     $LNSEL(R2),LINSEL  ;IS THIS ONE OF THE SELECTED LINES?
6999  016172  001002                          BNE     41$             ;IF SO,GO ANALYZE THE CHARACTER
7000  016174  104000                          ERROR                   ;INDICATE SOME KIND OF  ERROR
7001  016176  000753                          BR      4$              ;GO GET THE NEXT CHARACTER
7002  016200  005262  036204          41$:    INC     RCNT(R2)        ;COUNT THE CHARACTER
7003  016204  020362  027342                  CMP     R3,AETAB(R2)    ;IS THE DATA CORRECT ??
7004  016210  001420                          BEQ     5$              ;BR IF YES
7005
7006  016212  004767  010742                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
7007  016216  016204  027342                  MOV     AETAB(R2),R4    ;GET THE S/B DATA
7008  016222  062702  027342                  ADD     #AETAB,R2       ;GENERATE S/B ADDRESS
7009  016226  062701  000002                  ADD     #NRC,R1         ;GENERATE THE WAS ADDRESS
7010  016232  004767  006150                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
7011  016236  004567  006364                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
7012  016242  001216                          $TMP6
7013  016244  032552                          EM31+45
7014  016246  104031                          ERROR   31              ;DATA COMPARE ERROR
7015  016250  000415                          BR      TST52           ;;EXIT TEST ON ERROR
7016
7017  016252  022762  000100  036204  5$:     CMP     #100,RCNT(R2)   ;DONE 64. CHARS ON THIS LINE ?
7018  016260  001306                          BNE     3$              ;BR IF NOT
7019  016262  016711  162730                  MOV     $TMP6,(R1)      ;SELECT LINE IN SCR REG
7020  016266  042761  100000  000004          BIC     #BIT15,LPR(R1)  ;TURN OFF A.E. BIT
7021  016274  046267  027442  011200          BIC     LINBIT(R2),LINACT       ;ALL LINES INACTIVE ??
7022  016302  001275                          BNE     3$              ;BR IF NOT
```

```
7023                                    ;;*********************************************************************
7024                                    ;*TEST 52       BREAK BIT TEST - ALL SELECTED LINES
7025                                    ;;*********************************************************************
7026  016304  000004                    TST52:  SCOPE
7027                                    .REM    %
7028                                    TEST ABSTRACT:
7029                                    **************
7030
7031                                            THIS TEST VERIFIES THAT THE "BREAK" FEATURE WORKS PROPERLY
7032                                    FOR ALL SELECTED LINES. THE TEST SEQUENCE IS AS FOLLOWS:
7033
7034                                            1. SET UP THE ERROR LOOP RETURN
7035                                            2. RETRIEVE THE CORRECT S/B DATA FROM THE "BREAK" DATA
7036                                               TABLE AND UPDATE THE POINTER.
7037                                            3. GO SELECT A LINE TO TEST - GO TO THE NEXT TEST IF DONE ALL
7038                                               SELECTED LINES
7039                                            4. RESET THE DH11 AND CLEAR THE "CAR" AND "BCR" MEMORIES.
7040                                            5. PRIME SELECTED LINE TO OUTPUT TWO "NULL" CHARS TO CLEAR UART
7041                                            6. ACTIVATE THE SELECTED LINE
7042                                            7. WAIT FOR SILO TO RECEIVE TWO NULLS - IF TIMEOUT REPORT ERROR
7043                                               AND RESTART AT STEP 2
7044                                            8. IF NO TIMEOUT CLEAR THE SELECTED DH11 AND RESELECT LINE NO.
7045                                            9. PRIME SELECTED LINE TO OUT PUT 256. CHARS.
7046                                            10.SET THE SELECTED LINE'S BREAK BIT
7047                                            11.ACTIVATE THE SELECTED LINE
7048                                            12.WAIT FOR "BAR" REG TO CLEAR -F TIMEOUT REPORT ERROR AND
7049                                               RESTART AT STEP 2
7050                                            13.IF NO TIMEOUT VERIFY THAT THE SILO RECEIVED ONLY ONE CHAR-
7051                                               IF NOT REPORT ERROR AND RESTART AT STEP 2
7052                                            14.IF SILO RECEIVED ONLY ONE CHAR VERIFY THAT IT WAS A "BREAK"
7053                                               CHAR - IF NOT REPORT ERROR - AND RESTART AT STEP 2
7054
7055                                    ERRORS:
7056                                    ********
7057
7058                                    1. [ERROR  25] IS CALLED TO REPORT ALL ERRORS
7059
7060                                    SYNC:   M7277   SH4     LOAD BCR H      FU1
7061                                    *****
7062
7063                                    DEBUG:
7064                                    *******
7065
7066                                    1. IF ALL LINES FAILED SUSPECT THAT THE M7277 IS NOT GENERATING THE
7067                                       BREAK CONTROL REG LOAD SIGNAL.
7068
7069                                    2. IF ONLY ONE LINE FAILS SUSPECT THE BREAK CONTROL LOGIC ON THE M7278
7070
7071                                    KEY SIGNALS:
7072                                    ************
7073
7074                                            M7277   SH4     LOAD BCR H      FU1
7075
7076                                            M7278   SH5 THRU SH8
7077
7078                                                            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
```

```
7135  016572  016761  010440  000014          MOV     LINMSK,BKR(R1)  ;SET BREAK BIT FOR ACTIVE LINE
7136  016600  016761  010432  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
7137
7138  016606  012767  000005  011456          MOV     #5,TIMEA        ;INIT TIMER A
7139  016614  005067  011454                  CLR     TIMEB           ;INIT TIMER B
7140  016620  005761  000012          5$:     TST     BAR(R1)         ;BAR BIT CLEARED ??
7141  016624  001426                  BEQ     6$              ;BR IFD YES
7142  016626  004767  010144                  JSR     PC,TIMEIT       ;CALL THE TIMER
7143  016632  000772                  BR      5$              ;TIMER ROUTINE WILL MOVE RETURN PC
7144                                                          ;AROUND THIS BRANCH IF TIME OUT OCCURS
7145
7146  016634  004767  010320                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
7147  016640  010467  162340                  MOV     R4,$TMP1        ;SAVE THE S/B DATA
7148  016644  010102                  MOV     R1,R2           ;SET UP REGADR
7149  016646  062702  000012                  ADD     #BAR,R2
7150  016652  011203                  MOV     (R2),R3         ;GET THE WAS DATA
7151  016654  005004                  CLR     R4              ;SET UP S/B DATA
7152  016656  004767  005524                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
7153  016662  004567  005740                  JSR     R5,SUNUM        ;PUT LINE NO IN MESSAGE
7154  016666  030244                  LINE
7155  016670  032255                  EM25+34
7156  016672  016704  162306                  MOV     $TMP1,R4        ;RESTORE THE S/B DATA
7157  016676  104025                  ERROR   25              ;BAR BIT FAILED TO CLEAR
7158  016700  000613                  BR      1$              ;GO TRY NEXT LINE
7159
7160  016702  122761  000001  000017  6$:     CMPB    #1,SSR+1(R1)    ;ONE CHAR RECEIVED ?
7161  016710  001430                  BEQ     7$              ;BR IF YES
7162
7163  016712  004767  010242                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
7164  016716  010467  162262                  MOV     R4,$TMP1        ;SAVE THE S/B DATA
7165  016722  010102                  MOV     R1,R2           ;SET UP REGADR
7166  016724  062702  000016                  ADD     #SSR,R2
7167  016730  011203                  MOV     (R2),R3         ;GET THE WAS DATA
7168  016732  042703  100377                  BIC     #100377,R3      ;CLEAR JUNK
7169  016736  012704  000001                  MOV     #1,R4           ;SET UP S/B DATA
7170  016742  000304                  SWAB    R4
7171  016744  004767  005436                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
7172  016750  004567  005652                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MESSAGE
7173  016754  030244                  LINE
7174  016756  032255                  EM25+34
7175  016760  016704  162220                  MOV     $TMP1,R4        ;RESTORE THE S/B DATA
7176  016764  104025                  ERROR   25              ;FAILED TO RECEIVE THE ONE CHAR
7177  016766  000167  177336                  JMP     1$              ;GO TRY NEXT LINE
7178
7179
7180  016772  016103  000002          7$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
7181  016776  020304                  CMP     R3,R4           ;WAS IT A BREAK CHAR ?
7182  017000  001002                  BNE     8$              ;BR IF NOT CORRECT
7183  017002  000167  177322                  JMP     1$              ;GO TEST NEXT LINE
7184
7185  017006  004767  010146          8$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
7186  017012  010102                  MOV     R1,R2           ;SET UP REGADR
7187  017014  062702  000002                  ADD     #NRC,R2
7188  017020  004767  005362                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
7189  017024  004567  005576                  JSR     R5,SUNUM        ;PUT LINE NO IN MESSAGE
7190  017030  030244                  LINE
```

```
 7191  017032  032255                        EM25+34
 7192  017034  104025                        ERROR   25              ;INCORRECT DATA RECEIVED
 7193  017036  000167  177266                JMP     1$              ;GO TRY NEXT LINE
 7194  017042  000240                  9$:   NOP                     ;EXIT THIS TEST
 7195
```

```
                        ;;*****************************************************************
                        ;*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$:     JSR     PC,$ELINE
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
```

```
7252  017104  012761  177400  000010          MOV     #-400,BCR(R1)   ;XMIT 256(10) CHARS
7253  017112  012761  073503  000004          MOV     #73503,LPR(R1)  ;SET UP THE LINE PARAMETERS
7254  017120  016761  010112  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
7255
7256  017126  012767  000001  011136          MOV     #1,TIMEA        ;INIT TIME A
7257  017134  005067  011134                  CLR     TIMEB           ;INIT TIME B
7258  017140  005761  000012          3$:     TST     BAR(R1)         ;WAIT FOR XMITTR TO FINISH
7259  017144  001423                  BEQ     4$              ;BR IF XMITTR FINISHED
7260  017146  004767  007624                  JSR     PC,TIMEIT       ;CALL TIMER
7261  017152  000772                  BR      3$              ;TIMER WILL MOVE RETURN PC AROUND
7262                                                          ;THIS BRANCH IF TIMEOUT OCCURS
7263
7264  017154  004767  010000                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
7265  017160  016103  000012                  MOV     BAR(R1),R3      ;GET THE WAS DATA
7266  017164  010102                  MOV     R1,R2           ;SET UP REGADR
7267  017166  062702  000012                  ADD     #BAR,R2
7268  017172  005004                  CLR     R4              ;SET UP NEW S/B DATA
7269  017174  004767  005206                  JSR     PC,SUER2A       ;GO SET UP THE ERROR INFO
7270  017200  004567  005422                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
7271  017204  030244                  LINE
7272  017206  032320                  EM26+37
7273  017210  104026                  ERROR   26              ;BAR BIT FAILED TO CLEAR ON TIME
7274  017212  000720                  BR      1$              ;GO TRY NEXT LINE
7275
7276  017214  105711          4$:     TSTB    (R1)            ;CHAR AVAIL SET ??
7277  017216  100316                  BPL     1$              ;BR IF NOT IT SHOULDN'T BE
7278
7279  017220  004767  007734                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
7280  017224  010102                  MOV     R1,R2           ;SET UP REGADR
7281  017226  011103                  MOV     (R1),R3         ;GET WAS DATA
7282  017230  042703  100000                  BIC     #BIT15,R3       ;CLEAR JUNK BIT
7283  017234  116704  011004                  MOVB    LINE,R4         ;SET UP S/B DATA
7284  017240  004767  005142                  JSR     PC,SUER2A       ;GO SETUP ERROR INFO
7285  017244  004567  005356                  JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
7286  017250  030244                  LINE
7287  017252  032320                  EM26+37
7288  017254  104026                  ERROR   26              ;HALF DUPLEX FAILED TO BLIND RECVR
7289  017256  000676                  BR      1$              ;GO SELECT NEXT LINE
```

```
7290                                    ;;*****************************************************************
7291                                    ;*TEST 54       VERIFY THAT OVERRUN CAN SET PROPERLY - ALL SELECTED LINES
7292                                    ;;*****************************************************************
7293  017260  000004                    TST54:  SCOPE
7294                                    .REM    %
7295                                    TEST ABSTRACT:
7296                                    ***************
7297
7298                                            THIS TEST VERIFIES THAT "OVERRUN" SETS PROPERLY FOR ALL LINES
7299                                    THAT ARE SELECTED FOR TEST WHEN THE OVERRUN CONDITION IS FORCED BY THE
7300                                    PROGRAM. THE TEST SEQUENCE IS AS FOLLOWS:
7301
7302                                            1. SET UP THE ERROR LOOP RETURN
7303                                            2. SELECT A LINE NO. TO TEST - IF DONE ALL LINES GO TO
7304                                               THE NEXT TEST.
7305                                            3. PRIME THE SELECTED LINE TO XMIT 68. CHARS
7306                                            4. ACTIVATE THE SELECTED LINE
7307                                            5. WAIT FOR "XMIT DONE" TO SET - IF TIMEOUT REPORT ERROR
7308                                               AND RESTART AT STEP 2
7309                                            6. IF NO TIMEOUT READ 65. CHARS FROM THE SILO AND VERIFY THAT
7310                                               "OVERRUN" IS SET ON THE LAST WORD READ
7311                                            7. IF NOT REPORT ERROR AND RESTART AT STEP 2
7312
7313                                    ERRORS:
7314                                    ********
7315
7316                                    1. [ERROR  50] IS CALLED TO REPORT "XMIT DONE " TIMEOUTS
7317                                    2. [ERROR  56] IS CALLEDT TO REPORT "OVERRUN" ERROR
7318
7319                                    SYNC:   M7277   SH3     INIT A H        EF2
7320                                    *****
7321
7322                                    DEBUG:
7323                                    ******
7324
7325                                    1. IF FAULT APPEARS ON ONLY ONE LINE SUSPECT UART MODULE
7326                                       FOR THE APPROPRIATE LINE IN QUESTION.
7327
7328                                    2. IF FAULT APPEARS ON ALL LINES SUSPECT THE M7279 MODULE
7329
7330                                    KEY LOGIC:
7331                                    ***********
7332
7333                                            M7279   SH1     MASTER OR H     E12-9
7334                                                    SH2     MEMORY CHIP (3341) E13-11
7335
7336                                            M7280   SH2     UC1 OR 2 MASTER OR      EN2
7337                                                    SH2-5   UART PIN 15 (BUF OR LINE NN)
7338
7339                                    %
7340
7341  017262  012767  017276  161620            MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
7342  017270  004767  005240            1$:     JSR     PC,$ELINE       ;GO SELECT A LINE # TO TEST
7343  017274  000512                            BR      TST55           ;;BR IF DONE ALL SELECTED LINES
7344  017276  012711  004000            2$:     MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
7345  017302  116711  010736                    MOVB    LINE,(R1)       ;SELECT THE LINE TO TEST
```

```
7346  017306  012761  037244  000006          MOV     #TBUF,CAR(R1)   ;SET UP CURRENT ADDRESS
7347  017314  012761  177674  000010          MOV     #-68.,BCR(R1)   ;SET UP BYTE COUNT REG
7348  017322  012761  033503  000004          MOV     #33503,LPR(R1)  ;DO IT AT 9600 BAUD - 8 BITS
7349  017330  016761  007702  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
7350
7351  017336  012767  000001  010726          MOV     #1,TIMEA        ;INIT TIMERS A AND B
7352  017344  005067  010724                  CLR     TIMEB
7353  017350  005711                  3$:     TST     (R1)            ;TRANSMITTER DONE ??
7354  017352  100425                          BMI     4$              ;BR IF YES
7355  017354  004767  007416                  JSR     PC,TIMEIT       ;CALL TIMER
7356  017360  000773                          BR      3$              ;BR IF NO TIMEOUT
7357
7358  017362  004767  007572                  JSR     PC,SAPS         ;GO SAVE PSW
7359  017366  011103                          MOV     (R1),R3         ;GET THE WAS DATA
7360  017370  042703  077760                  BIC     #77760,R3       ;CLEAR UNINTERESTING BITS
7361  017374  116704  010644                  MOVB    LINE,R4         ;SET UP S/B DATA
7362  017400  052704  100000                  BIS     #BIT15,R4
7363  017404  010102                          MOV     R1,R2           ;SET UP REGADR
7364  017406  004767  004774                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
7365  017412  004567  005210                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE HEADER
7366  017416  030244                          LINE
7367  017420  034176                          EM50+53
7368  017422  104050                          ERROR   50              ;REPORT XMIT DONE TIME OUT
7369  017424  000721                          BR      1$              ;GO TRY NEXT LINE
7370
7371  017426  012767  000101  161550  4$:     MOV     #65.,$TMP1      ;SET UP TO READ 65. WORDS FROM SILO
7372  017434  116704  010604                  MOVB    LINE,R4         ;SET UP S/B DATA
7373  017440  000304                          SWAB    R4
7374  017442  152704  000101                  BISB    #65.,R4
7375  017446  052704  140000                  BIS     #BIT15+BIT14,R4 ;PUT IN OVERRUN AND VALID DATA BITS
7376  017452  016103  000002          5$:     MOV     NRC(R1),R3      ;GET WAS DATA FROM SILO
7377  017456  005367  161522                  DEC     $TMP1           ;COUNT ONE WORD READ
7378  017462  001373                          BNE     5$              ;BR TIL 65. READ
7379  017464  020304                          CMP     R3,R4           ;WAS DATA AND OVERRUN CORRECT ??
7380  017466  001700                          BEQ     1$              ;BR IF YES TRY NEXT SELECTED LINE
7381
7382  017470  004767  007464                  JSR     PC,SAPS         ;GO SAVE PSW
7383  017474  010102                          MOV     R1,R2           ;SET UP REGADR
7384  017476  062702  000002                  ADD     #NRC,R2
7385  017502  004767  004700                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
7386  017506  004567  005114                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG HDR
7387  017512  030244                          LINE
7388  017514  034614                          EM56+42
7389  017516  104056                          ERROR   56              ;OVERRUN OR DATA INCORRECT
7390  017520  000663                          BR      1$              ;GO TEST NEXT SELECTED LINE
7391
```

```
                                ;;*******************************************************************
                                ;*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
```

```
7448  017714  117703  010312              MOVB    @DHMCSR,R3          ;READ CONTROL STATUS REGISTER
7449  017720  042703  177760              BIC     #177760,R3          ;CLEAR UNWANTED BITS
7450  017724  020103                      CMP     R1,R3               ;IF LINE NUMBER=SELECTED LINE NUMBER,
7451  017726  001002                      BNE     MUX11C              ;EXCEPT LINE ENABLE FUNCTION FLIP FLOP
7452  017730  012705  000001              MOV     #LINENA,R5
7453                                                                  ;TO BE SET
7454  017734  020504              MUX11C: CMP     R5,R4               ;COMPARE EXPECTED AND RECEIVED
7455  017736  001403                      BEQ     MUX11D              ;RESULTS
7456  017740  030504                      BIT     R5,R4               ;TEST TO SEE IF THE LINE ENABLE BIT WAS SET
7457  017742  001001                      BNE     MUX11D              ;IF SO, EVERYTHING IS OK
7458  017744  104060                      ERROR   60                  ;INDICATE A MODEM CONTROL ERROR
7459  017746  052777  000400  010256  MUX11D: BIS     #STEP,@DHMCSR       ;EXAMINE NEXT LINE
7460  017754  005302                      DEC     R2
7461  017756  001353                      BNE     MUX11B
7462  017760  005005                      CLR     R5
7463  017762  010177  010244              MOV     R1,@DHMCSR
7464  017766  010103              MUX11E: MOV     R1,R3               ;SET LINE COUNTER TO SELECTED LINE
7465  017770  005077  010240              CLR     @DHMLSR             ;CLEAR LINE ENABLE FLIP FLOP
7466  017774  105227  000000              INCB    #0                  ;DELAY FOR CABLE
7467  020000  001375                      BNE     .-4                 ;DITTO
7468  020002  017704  010226              MOV     @DHMLSR,R4          ;READ LINE STATUS REGISTER
7469                                                                  ;WAS LINE ENABLE FUNCTION FLIP FLOP
7470  020006  001704                      BEQ     DOIT11              ;CLEARED
7471  020010  104060                      ERROR   60                  ;MODEM CONRTROL ERROR
7472  020012  000702                      BR      DOIT11              ;DO THE NEXT LINE THAT IS SELECTED
7473
```

```
7474                                    ;;*****************************************************************
7475                                    ;*TEST 56       MODEM CONTROL DIAGNOSTIC CONTINUED
7476                                    ;;*****************************************************************
7477  020014  000004                    TST56:  SCOPE
7478                                    .REM    %
7479                                    TEST ABSTRACT:
7480                                    ***************
7481
7482                                            THIS TEST VERIFIES THAT CLEAR TO SEND AND CARRIER ARE SET
7483                                    IF  "LINE ENABLE" AND TERMINAL ARE SET FOR THE SELECTED LINE.
7484
7485                                    ERRORS:
7486                                    ********
7487
7488                                            IF ANY ERRORS OCCUR, RUN THE  MODEM CONTROL DIAGNOSTIC,
7489                                     DZDHK.
7490
7491                                    %
7492  020016  005077  010212            DOIT15: CLR     @DHMLSR                 ;CLEAR LINE STATUS REGISTER
7493  020022  004767  004506                    JSR     PC,SELINE               ;GO SELECT A LINE.
7494  020026  000471                            BR      TST57           ;;BR IF DONE ALL SELECTED LINES
7495  020030  005077  010176            MUX15:  CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
7496  020034  042767  000340  157734            BIC     #340,PS                 ;ENABLE INTERRUPTS
7497  020042  116701  010176                    MOVB    LINE,R1
7498  020046  012777  002000  010156            MOV     #CLRMUX,@DHMCSR         ;RESET ALL THE LINE STATUS REGISTERS
7499  020054  010167  161126                    MOV     R1,$TMP2                ;SAVE LINE NBUMBER FOR ERROR MESSAGE
7500  020060  012702  000020            MUX15A: MOV     #16.,R2                 ;16 LINES
7501  020064  010177  010142                    MOV     R1,@DHMCSR              ;SELECT A LINE
7502  020070  012777  000003  010136            MOV     #LINENA+TRMRDY,@DHMLSR  ;SET LINE ENABLE +TRMRDY
7503  020076  005077  010130                    CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
7504  020102  005005                    MUX15B: CLR     R5                      ;CLEAR EXPECTED RESULT
7505  020104  017704  010124                    MOV     @DHMLSR,R4              ;READ LINE STATUS
7506  020110  117703  010116                    MOVB    @DHMCSR,R3              ;READ LINE NUMBER
7507  020114  042703  177760                    BIC     #177760,R3              ;CLEAR UNWANTED BITS
7508  020120  020103                            CMP     R1,R3                   ;IF RECEIVED LINE=SELECTED LINE
7509  020122  001002                            BNE     MUX15C                  ;EXPECT LINE ENABLE AND
7510  020124  012705  000143                    MOV     #LINENA+TRMRDY+CO+CS,R5
7511                                                                            ;CLEAR TO SEND AND CARRIER ARE SET
7512  020130  020405                    MUX15C: CMP     R4,R5                   ;COMPARE EXPECTED AND
7513  020132  001401                            BEQ     MUX15D                  ;RECEIVED RESULTS
7514  020134  104060                            ERROR   60                      ;MODEM CONTROL ERROR
7515  020136  052777  000400  010066    MUX15D: BIS     #STEP,@DHMCSR           ;UPDATE LINE COUNTER
7516  020144  005302                            DEC     R2                      ;CONTINUE IF ALL CHECKS
7517  020146  001355                            BNE     MUX15B                  ;ARE NOT DONE FOR THIS LINE
7518  020150  012705  000001                    MOV     #LINENA,R5              ;EXPECT LINE ENABLE
7519  020154  010103                    MUX15E: MOV     R1,R3                   ;ON SELECTED LINE
7520  020156  010177  010050                    MOV     R1,@DHMCSR              ;SELECT LINE
7521  020162  042777  000002  010044            BIC     #TRMRDY,@DHMLSR         ;CLEAR TERMINAL
7522  020170  105227  000000                    INCB    #0                      ;DELAY FOR CABLE
7523  020174  001375                            BNE     .-4                     ;DITTO
7524  020176  017704  010032                    MOV     @DHMLSR,R4              ;READ LINE STATUS REGISTER
7525  020202  020504                            CMP     R5,R4                   ;ONLY LINE ENABLE SHOULD BE
7526  020204  001704                            BEQ     DOIT15                  ;SET ON THIS LINE
7527  020206  104060                            ERROR   60                      ;MODEM COMNTROL ERROR
7528  020210  000702                            BR      DOIT15                  ;GO DO THE NEXT LINE
7529
```

```
7530                                    ;;*****************************************************************
7531                                    ;*TEST 57       MODEM CONTROL DIAGNOSTIC CONTINUED
7532                                    ;;*****************************************************************
7533  020212  000004                    TST57:  SCOPE
7534                                    .REM    %
7535                                    TEST ABSTRACT:
7536                                    ****************
7537
7538                                            THIS TEST VERIFIES THAT RING IS SET IF "LINE ENABLE" AND
7539                                    REQUEST TO SEND ARE SET FOR THE SELECTED LINE.
7540
7541                                    ERRORS:
7542                                    *********
7543
7544                                            IF ANY ERRORS OCCUR, RUN THE  MODEM CONTROL DIAGNOSTIC,
7545                                     DZDHK.
7546
7547                                    %
7548  020214  005077  010014            DOIT16: CLR     @DHMLSR                 ;CLEAR LINE STATUS REGISTER
7549  020220  004767  004310                    JSR     PC,SELINE               ;GO SELECT A LINE.
7550  020224  000466                            BR      TST60           ;;BR IF DONE ALL SELECTED LINES
7551  020226  005077  010000            MUX16:  CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
7552  020232  042767  000340  157536            BIC     #340,PS                 ;ENABLE INTERRUPTS
7553  020240  116701  010000                    MOVB    LINE,R1
7554  020244  010167  160736                    MOV     R1,$TMP2                ;SAVE LINE NBUMBER FOR ERROR MESSAGE
7555  020250  012702  000020            MUX16A: MOV     #16.,R2                 ;16 LINES
7556  020254  010177  007752                    MOV     R1,@DHMCSR              ;SELECT A LINE
7557  020260  012777  000005  007746            MOV     #LINENA+RS,@DHMLSR      ;SET LINE ENABLE +RS
7558  020266  005077  007740                    CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
7559  020272  005005                    MUX16B: CLR     R5                      ;CLEAR EXPECTED RESULT
7560  020274  017704  007734                    MOV     @DHMLSR,R4              ;READ LINE STATUS
7561  020300  117703  007726                    MOVB    @DHMCSR,R3              ;READ LINE NUMBER
7562  020304  042703  177760                    BIC     #177760,R3              ;CLEAR UNWANTED BITS
7563  020310  020103                            CMP     R1,R3                   ;IF RECEIVED LINE=SELECTED LINE
7564  020312  001002                            BNE     MUX16C                  ;EXPECT LINE ENABLE AND
7565  020314  012705  000205                    MOV     #LINENA+RS+RING,R5
7566                                                                            ;RING IS SET
7567  020320  020405                    MUX16C: CMP     R4,R5                   ;COMPARE EXPECTED AND
7568  020322  001401                            BEQ     MUX16D                  ;RECEIVED RESULTS
7569  020324  104060                            ERROR   60                      ;MODEM CONTROL ERROR
7570  020326  052777  000400  007676    MUX16D: BIS     #STEP,@DHMCSR           ;UPDATE LINE COUNTER
7571  020334  005302                            DEC     R2                      ;CONTINUE IF ALL CHECKS
7572  020336  001355                            BNE     MUX16B                  ;ARE NOT DONE FOR THIS LINE
7573  020340  012705  000001                    MOV     #LINENA,R5              ;EXPECT LINE ENABLE
7574  020344  010103                    MUX16E: MOV     R1,R3                   ;ON SELECTED LINE
7575  020346  010177  007660                    MOV     R1,@DHMCSR              ;SELECT LINE
7576  020352  042777  000004  007654            BIC     #RS,@DHMLSR             ;CLEAR REQUEST TO SEND
7577  020360  105227  000000                    INCB    #0                      ;DELAY FOR CABLE
7578  020364  001375                            BNE     .-4                     ;DITTO
7579  020366  017704  007642                    MOV     @DHMLSR,R4              ;READ LINE STATUS REGISTER
7580  020372  020504                            CMP     R5,R4                   ;ONLY LINE ENABLE SHOULD BE
7581  020374  001707                            BEQ     DOIT16                  ;SET ON THIS LINE
7582  020376  104060                            ERROR   60                      ;MODEM CONTROL ERROR
7583  020400  000705                            BR      DOIT16                  ;GO DO THE NEXT LINE
7584
```

```
                                ;;*****************************************************************
                                ;*TEST 60       MODEM CONTROL DIAGNOSTIC CONTINUED
                                ;;*****************************************************************
020402  000004                  ST60:  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
```

```
7641  020572  000004                        ENDA:   SCOPE
7642  020574  012767  000240  000054                MOV     #240,$EOP       ;NOP THE SCOPE AT THE BEGINNING OF EOP
7643  020602  005267  007434                        INC     DHNUM           ;GENERATE NEW DH11 NUMBER
7644  020606  062767  000002  007434                ADD     #2,ADPTR        ;UPDATE THE TABLE POINTERS
7645  020614  062767  000002  007430                ADD     #2,VCPTR
7646  020622  062767  000002  007424                ADD     #2,BRPTR
7647  020630  006367  006374                        ASL     SELMSK          ;SHIFT MARKER TO TEST NEXT DH11
7648  020634  001410                                BEQ     $EOP            ;BR IF TESTED ALL SELECTED DH11'S
7649  020636  036767  006366  006366                BIT     SELMSK,DHSEL    ;IS THIS DH11 SELECTED ?
7650  020644  001752                                BEQ     ENDA            ;BR IF NOT
7651  020646  105067  160230                        CLRB    $TSTNM          ;INIT TEST NUMBER
7652  020652  000167  162062                        JMP     RSTRTA          ;GO TEST THIS DH11
7653                                        .SBTTL  END OF PASS ROUTINE
7654
7655                                        ;;**************************************************************************
7656                                        ;*INCREMENT THE PASS NUMBER ($PASS)
7657                                        ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
7658                                        ;*IF THERES A MONITOR GO TO IT
7659                                        ;*IF THERE ISN'T JUMP TO START2
7660
7661  020656                                $EOP:
7662  020656  000004                                SCOPE
7663  020660  005067  160216                        CLR     $TSTNM          ;;ZERO THE TEST NUMBER
7664  020664  005067  160332                        CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
7665  020670  005267  160344                        INC     $PASS           ;;INCREMENT THE PASS NUMBER
7666  020674  042767  100000  160336                BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
7667  020702  005327                                DEC     (PC)+           ;;LOOP?
7668  020704  000001                        $EOPCT: .WORD   1
7669  020706  003022                                BGT     $DOAGN          ;;YES
7670  020710  012737                                MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
7671  020712  000001                        $ENDCT: .WORD   1
7672  020714  020704                                $EOPCT
7673  020716  104401  020763                        TYPE    ,$ENDMG         ;;TYPE "END PASS #"
7674  020722  016746  160312                        MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
7675  020726  104405                                TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
7676  020730  104401  020760                        TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
7677  020734  013700  000042                $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
7678  020740  001405                                BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
7679  020742  000005                                RESET                   ;;CLEAR THE WORLD
7680  020744  004710                        $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
7681  020746  000240                                NOP                     ;;SAVE ROOM
7682  020750  000240                                NOP                     ;;FOR
7683  020752  000240                                NOP                     ;;ACT11
7684  020754                                $DOAGN:
7685  020754  000137                                JMP     @(PC)+          ;;RETURN
7686  020756  002634                        $RTNAD: .WORD   START2
7687  020760     377     377     000        $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
7688  020763     015  042412  042116        $ENDMG: .ASCIZ  <15><12>/END PASS #/
7689  020770  050040  051501  020123
7690  020776  000043
7691                                        .SBTTL  SCOPE HANDLER ROUTINE
7692
7693                                        ;;**************************************************************************
7694                                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
7695                                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
7696                                        ;*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$:;*****END OF CODE FOR THE XOR TESTER*****
021060  032777  000400  160052         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
021066  001404                         BEQ     2$              ;;BR 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
```

```
7753  021252  016777  157624  157662  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
7754  021260  016716  157622                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
7755  021264  000002                          RTI                     ;;FIXES PS
7756  021266  000010                  $MXCNT: 10                      ;;MAX. NUMBER OF ITERATIONS
7757                                  .SBTTL  ERROR HANDLER ROUTINE
7758
7759                                  ;;**********************************************************************
7760                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
7761                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
7762                                  ;*AND GO TO $ERRTYP ON ERROR
7763                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
7764                                  ;*SW15=1         HALT ON ERROR
7765                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
7766                                  ;*SW09=1         LOOP ON ERROR
7767                                  ;*CALL
7768                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
7769
7770  021270                          $ERROR:
7771  021270  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
7772  021272  105267  157605          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
7773  021276  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
7774  021300  016777  157576  157634        MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
7775  021306  005267  157600                INC     $ERTTL          ;;INC THE ERROR COUNT
7776  021312  011667  157600                MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
7777  021316  162767  000002  157572        SUB     #2,$ERRPC
7778  021324  117767  157566  157562        MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
7779  021332  032777  020000  157600        BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
7780  021340  001004                        BNE     20$             ;;SKIP TYPEOUTS
7781  021342  004767  000106                JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
7782  021346  104401  001227                TYPE    ,$CRLF
7783  021352                          20$:
7784  021352  122767  000001  157672        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
7785  021360  001007                        BNE     2$              ;;NO,SKIP APT ERROR REPORT
7786  021362  116767  157526  000004        MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
7787  021370  004767  001166                JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
7788  021374     000                  21$:    .BYTE   0
7789  021375     000                          .BYTE   0
7790  021376  000777                  22$:    BR      22$             ;;APT ERROR LOOP
7791  021400  005777  157534          2$:     TST     @SWR            ;;HALT ON ERROR
7792  021404  100002                        BPL     3$              ;;SKIP IF CONTINUE
7793  021406  000000                        HALT                    ;;HALT ON ERROR!
7794  021410  104407                        CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
7795  021412  032777  001000  157520  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
7796  021420  001402                        BEQ     4$              ;;BR IF NO
7797  021422  016716  157462                MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
7798  021426  005767  157572          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
7799  021432  001402                        BEQ     5$              ;;BR IF NONE
7800  021434  016716  157564                MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
7801  021440                          5$:
7802  021440  022737  020744  000042        CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
7803  021446  001001                        BNE     6$              ;;BRANCH IF NO
7804  021450  000000                        HALT                    ;;YES
7805  021452                          6$:
7806  021452  000002                        RTI                     ;;RETURN
7807                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
7808
```

```
                                ;;**********************************************************************
                                ;*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
```

```
 7865                                   ;*                                      ;;1=TYPE LEADING ZEROS
 7866                                   ;*                                      ;;0=SUPPRESS LEADING ZEROS
 7867                                   ;*
 7868                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 7869                                   ;*$TYPOS OR $TYPOC
 7870                                   ;*CALL:
 7871                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 7872                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
 7873                                   ;*
 7874                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 7875                                   ;*CALL:
 7876                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 7877                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
 7878
 7879  021610  017646  000000           $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 7880  021614  116667  000001  000211           MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 7881  021622  112667  000207                   MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 7882  021626  062716  000002                   ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 7883  021632  000406                           BR      $TYPON
 7884  021634  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 7885  021642  112767  000006  000165           MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 7886  021650  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 7887  021656  010346                           MOV     R3,-(SP)        ;;SAVE R3
 7888  021660  010446                           MOV     R4,-(SP)        ;;SAVE R4
 7889  021662  010546                           MOV     R5,-(SP)        ;;SAVE R5
 7890  021664  116704  000145                   MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 7891  021670  005404                           NEG     R4
 7892  021672  062704  000006                   ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 7893  021676  110467  000132                   MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 7894  021702  116704  000125                   MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 7895  021706  016605  000012                   MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 7896  021712  005003                           CLR     R3              ;;CLEAR THE OUTPUT WORD
 7897  021714  006105                   1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 7898  021716  000404                           BR      3$              ;;GO DO MSB
 7899  021720  006105                   2$:     ROL     R5              ;;FORM THIS DIGIT
 7900  021722  006105                           ROL     R5
 7901  021724  006105                           ROL     R5
 7902  021726  010503                           MOV     R5,R3
 7903  021730  006103                   3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 7904  021732  105367  000076                   DECB    $OMODE          ;;TYPE THIS DIGIT?
 7905  021736  100016                           BPL     7$              ;;BR IF NO
 7906  021740  042703  177770                   BIC     #177770,R3      ;;GET RID OF JUNK
 7907  021744  001002                           BNE     4$              ;;TEST FOR 0
 7908  021746  005704                           TST     R4              ;;SUPPRESS THIS 0?
 7909  021750  001403                           BEQ     5$              ;;BR IF YES
 7910  021752  005204                   4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 7911  021754  052703  000060                   BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 7912  021760  052703  000040           5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 7913  021764  110367  000040                   MOVB    R3,8$           ;;SAVE FOR TYPING
 7914  021770  104401  022030                   TYPE    ,8$             ;;GO TYPE THIS DIGIT
 7915  021774  105367  000032           7$:     DECB    $OCNT           ;;COUNT BY 1
 7916  022000  003347                           BGT     2$              ;;BR IF MORE TO DO
 7917  022002  002402                           BLT     6$              ;;BR IF DONE
 7918  022004  005204                           INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 7919  022006  000744                           BR      2$              ;;GO DO THE LAST DIGIT
 7920  022010  012605                   6$:     MOV     (SP)+,R5        ;;RESTORE R5
```

```
7921  022012  012604                          MOV     (SP)+,R4        ;;RESTORE R4
7922  022014  012603                          MOV     (SP)+,R3        ;;RESTORE R3
7923  022016  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
7924  022024  012616                          MOV     (SP)+,(SP)
7925  022026  000002                          RTI                     ;;RETURN
7926  022030     000                  B$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
7927  022031     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
7928  022032     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
7929  022033     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
7930  022034  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
7931                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
7932
7933                                  ;;**************************************************************
7934                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
7935                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
7936                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
7937                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
7938                                  ;*REPLACED WITH SPACES.
7939                                  ;*CALL:
7940                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
7941                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
7942
7943  022036                          $TYPDS:
7944  022036  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
7945  022040  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
7946  022042  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
7947  022044  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
7948  022046  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
7949  022050  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
7950  022054  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
7951  022060  100004                          BPL     1$              ;;BR IF INPUT IS POS.
7952  022062  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
7953  022064  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
7954  022072  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
7955  022074  012703  022252                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
7956  022100  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
7957  022104  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
7958  022106  016001  022242                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
7959  022112  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
7960  022114  002402                          BLT     4$              ;;BR IF DONE
7961  022116  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
7962  022120  000774                          BR      3$
7963  022122  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
7964  022124  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
7965  022126  001002                          BNE     5$              ;;FALL THROUGH IF 0
7966  022130  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
7967  022132  100407                          BMI     7$              ;;BR IF YES
7968  022134  106316                  5$:     ASLB    (SP)            ;;MSD?
7969  022136  103003                          BCC     6$              ;;BR IF NO
7970  022140  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
7971  022146  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
7972  022152  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
7973  022156  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
7974  022160  005720                          TST     (R0)+           ;;JUST INCREMENTING
7975  022162  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
7976  022166  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
```

```
                                        BGT     8$              ;;GO TO EXIT
                                        MOV     R5,R2           ;;GET THE LSD
                                        BR      6$              ;;GO CHANGE TO ASCII
                                8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
                                        BPL     9$              ;;BR IF NO
                                        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
                                9$:     CLRB    (R3)            ;;SET THE TERMINATOR
                                        MOV     (SP)+,R5        ;;POP STACK INTO R5
                                        MOV     (SP)+,R3        ;;POP STACK INTO R3
                                        MOV     (SP)+,R2        ;;POP STACK INTO R2
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                        TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
                                        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
                                        MOV     (SP)+,(SP)
                                        RTI                     ;;RETURN TO USER
                                $DTBL:  10000.
                                        1000.
                                        100.
                                        10.
                                $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
```

```
 8033  022352  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
 8034  022354  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 8035  022360  000002                        RTI                     ;;RETURN
 8036  022362  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 8037  022366  001430                        BEQ     8$
 8038  022370  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 8039  022374  001006                        BNE     5$
 8040  022376  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
 8041  022400  104401                        TYPE                    ;;TYPE A CR AND LF
 8042  022402  001227                        $CRLF
 8043  022404  105067  000130                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 8044  022410  000755                        BR      2$              ;;GET NEXT CHARACTER
 8045  022412  004767  000056        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 8046  022416  126726  156534        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 8047  022422  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
 8048  022424  016746  156524                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 8049                                                                ;;AND THE NULL CHAR.
 8050  022430  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 8051  022434  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 8052  022436  004767  000032                JSR     PC,$TYPEC       ;;GO TYPE A NULL
 8053  022442  105367  000072                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 8054  022446  000770                        BR      7$              ;;LOOP
 8055
 8056                                  ;HORIZONTAL TAB PROCESSOR
 8057
 8058  022450  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 8059  022454  004767  000014        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 8060  022460  132767  000007  000052        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 8061  022466  001372                        BNE     9$              ;;TAB STOP
 8062  022470  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
 8063  022472  000724                        BR      2$              ;;GET NEXT CHARACTER
 8064  022474  105777  156450        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 8065  022500  100375                        BPL     $TYPEC
 8066  022502  116677  000002  156442        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 8067  022510  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 8068  022516  001003                        BNE     1$              ;;BRANCH IF NO
 8069  022520  105067  000014                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 8070  022524  000406                        BR      $TYPEX          ;;EXIT
 8071  022526  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 8072  022534  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
 8073  022536  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
 8074  022540  000000        $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
 8075  022542  000207        $TYPEX: RTS     PC
 8076
 8077                                  .SBTTL  APT COMMUNICATIONS ROUTINE
 8078
 8079                                  ;;**************************************************************
 8080  022544  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
 8081  022552  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
 8082  022560  000403                        BR      $ATYC
 8083  022562  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
 8084  022570                        $ATYC:
 8085  022570  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 8086  022572  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8087  022574  105767  000206                TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
 8088  022600  001450                        BEQ     5$              ;;IF NOT:  BR
```

```
8089  022602  122767  000001  156442          CMPB    #APTENV,$ENV      ;;OPERATING UNDER APT?
8090  022610  001031                          BNE     3$                ;;IF NOT:  BR
8091  022612  132767  000100  156433          BITB    #APTSPOOL,$ENVM   ;;SHOULD SPOOL MESSAGES?
8092  022620  001425                          BEQ     3$                ;;IF NOT:  BR
8093  022622  017600  000004                  MOV     @4(SP),R0         ;;GET MESSAGE ADDR.
8094  022626  062766  000002  000004          ADD     #2,4(SP)                  ;;BUMP RETURN ADDR.
8095  022634  005767  156372          1$:     TST     $MSGTYPE          ;;SEE IF DONE W/ LAST XMISSION?
8096  022640  001375                          BNE     1$                ;;IF NOT:  WAIT
8097  022642  010067  156400                  MOV     R0,$MSGAD         ;;PUT ADDR IN MAILBOX
8098  022646  105720                  2$:     TSTB    (R0)+             ;;FIND END OF MESSAGE
8099  022650  001376                          BNE     2$
8100  022652  166700  156370                  SUB     $MSGAD,R0         ;;SUB START OF MESSAGE
8101  022656  006200                          ASR     R0                ;;GET MESSAGE LNGTH IN WORDS
8102  022660  010067  156364                  MOV     R0,$MSGLGT        ;;PUT LENGTH IN MAILBOX
8103  022664  012767  000004  156340          MOV     #4,$MSGTYPE       ;;TELL APT TO TAKE MSG.
8104  022672  000413                          BR      5$
8105  022674  017667  000004  000016  3$:     MOV     @4(SP),4$         ;;PUT MSG ADDR IN JSR LINKAGE
8106  022702  062766  000002  000004          ADD     #2,4(SP)                  ;;BUMP RETURN ADDRESS
8107  022710  016746  155062                  MOV     177776,-(SP)      ;;PUSH 177776 ON STACK
8108  022714  004767  177342                  JSR     PC,$TYPE          ;;CALL TYPE MACRO
8109  022720  000000                  4$:     .WORD   0
8110  022722                          5$:
8111  022722  105767  000062          10$:    TSTB    $FFLG             ;;SHOULD REPORT FATAL ERROR?
8112  022726  001416                          BEQ     12$               ;;IF NOT:  BR
8113  022730  005767  156316                  TST     $ENV              ;;RUNNING UNDER APT?
8114  022734  001413                          BEQ     12$               ;;IF NOT:  BR
8115  022736  005767  156270          11$:    TST     $MSGTYPE          ;;FINISHED LAST MESSAGE?
8116  022742  001375                          BNE     11$               ;;IF NOT:  WAIT
8117  022744  017667  000004  156262          MOV     @4(SP),$FATAL     ;;GET ERROR #
8118  022752  062766  000002  000004          ADD     #2,4(SP)                  ;;BUMP RETURN ADDR.
8119  022760  005267  156246                  INC     $MSGTYPE          ;;TELL APT TO TAKE ERROR
8120  022764  105067  000020          12$:    CLRB    $FFLG             ;;CLEAR FATAL FLAG
8121  022770  105067  000013                  CLRB    $LFLG             ;;CLEAR LOG FLAG
8122  022774  105067  000006                  CLRB    $MFLG             ;;CLEAR MESSAGE FLAG
8123  023000  012601                          MOV     (SP)+,R1          ;;POP STACK INTO R1
8124  023002  012600                          MOV     (SP)+,R0          ;;POP STACK INTO R0
8125  023004  000207                          RTS     PC                ;;RETURN
8126  023006     000          $MFLG:  .BYTE   0                 ;;MESSG. FLAG
8127  023007     000          $LFLG:  .BYTE   0                 ;;LOG FLAG
8128  023010     000          $FFLG:  .BYTE   0                 ;;FATAL FLAG
8129        023012                          .EVEN
8130        000200                  APTSIZE=200
8131        000001                  APTENV=001
8132        000100                  APTSPOOL=100
8133        000040                  APTCSUP=040
8134                                .SBTTL  TTY INPUT ROUTINE
8135
8136                                ;;**********************************************************************
8137                                .ENABL  LSB
8138
8139                                ;;**********************************************************************
8140                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
8141                                ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
8142                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
8143                                ;*WHEN OPERATING IN TTY FLAG MODE.
8144  023012  022767  000176  156120  $CKSWR: CMP     #SWREG,SWR        ;;IS THE SOFT-SWR SELECTED?
```

```
8145  023020  001074                        BNE     15$             ;;BRANCH IF NO
8146  023022  105777  156116                TSTB    @$TKS           ;;CHAR THERE?
8147  023026  100071                        BPL     15$             ;;IF NO, DON'T WAIT AROUND
8148  023030  117746  156112                MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
8149  023034  042716  177600                BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
8150  023040  022726  000007                CMP     #7,(SP)+        ;;IS IT A CONTROL G?
8151  023044  001062                        BNE     15$             ;;NO, RETURN TO USER
8152  023046  126727  156062  000001        CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
8153  023054  001456                        BEQ     15$             ;;BRANCH IF YES
8154
8155  023056  104401  023665                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
8156  023062  104401  023672        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
8157  023066  016746  155104                MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
8158  023072  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8159  023074  104401  023703                TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
8160  023100  005046                19$:    CLR     -(SP)           ;;CLEAR COUNTER
8161  023102  005046                        CLR     -(SP)           ;;THE NEW SWR
8162  023104  105777  156034        7$:     TSTB    @$TKS           ;;CHAR THERE?
8163  023110  100375                        BPL     7$              ;;IF NOT TRY AGAIN
8164
8165  023112  117746  156030                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
8166  023116  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
8167
8168
8169
8170  023122  021627  000025        9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
8171  023126  001005                        BNE     10$             ;;BRANCH IF NOT
8172  023130  104401  023660                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
8173  023134  062706  000006        20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
8174  023140  000757                        BR      19$             ;;LET'S TRY IT AGAIN
8175
8176
8177  023142  021627  000015        10$:    CMP     (SP),#15        ;;IS IT A <CR>?
8178  023146  001022                        BNE     16$             ;;BRANCH IF NO
8179  023150  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
8180  023154  001403                        BEQ     11$             ;;BRANCH IF YES
8181  023156  016677  000002  155754        MOV     2(SP),@SWR      ;;SAVE NEW SWR
8182  023164  062706  000006        11$:    ADD     #6,SP           ;;CLEAR UP STACK
8183  023170  104401  001227        14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
8184  023174  126727  155735  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
8185  023202  001003                        BNE     15$             ;;BRANCH IF NOT
8186  023204  012777  000100  155732        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
8187  023212  000002                15$:    RTI                     ;;RETURN
8188  023214  004767  177254        16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
8189  023220  021627  000060                CMP     (SP),#60        ;;CHAR < 0?
8190  023224  002420                        BLT     18$             ;;BRANCH IF YES
8191  023226  021627  000067                CMP     (SP),#67        ;;CHAR > 7?
8192  023232  003015                        BGT     18$             ;;BRANCH IF YES
8193  023234  042726  000060                BIC     #60,(SP)+       ;;STRIP-OFF ASCII
8194  023240  005766  000002                TST     2(SP)           ;;IS THIS THE FIRST CHAR
8195  023244  001403                        BEQ     17$             ;;BRANCH IF YES
8196  023246  006316                        ASL     (SP)            ;;NO, SHIFT PRESENT
8197  023250  006316                        ASL     (SP)            ;;    CHAR OVER TO MAKE
8198  023252  006316                        ASL     (SP)            ;;    ROOM FOR NEW ONE.
8199  023254  005266  000002        17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
8200  023260  056616  177776                BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
```

```
8201  023264  000707                        BR      7$              ;;GET THE NEXT ONE
8202  023266  104401  001226        18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
8203  023272  000720                        BR      20$             ;;SIMULATE CONTROL-U
8204                                .DSABL  LSB
8205
8206
8207                                ;;**************************************************************
8208                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
8209                                ;*CALL:
8210                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
8211                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
8212                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
8213                                ;
8214
8215  023274  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
8216  023276  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
8217  023304  105777  155634        1$:     TSTB    @$TKS           ;;WAIT FOR
8218  023310  100375                        BPL     1$              ;;A CHARACTER
8219  023312  117766  155630  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
8220  023320  042766  177600  000004        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
8221  023326  026627  000004  000023        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
8222  023334  001013                        BNE     3$              ;;BRANCH IF NO
8223  023336  105777  155602        2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
8224  023342  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
8225  023344  117746  155576                MOVB    @$TKB,-(SP)     ;;GET CHARACTER
8226  023350  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
8227  023354  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
8228  023360  001366                        BNE     2$              ;;IF NOT DISCARD IT
8229  023362  000750                        BR      1$              ;;YES, RESUME
8230  023364  026627  000004  000140  3$:   CMP     4(SP),#140      ;;IS IT UPPER CASE?
8231  023372  002407                        BLT     4$              ;;BRANCH IF YES
8232  023374  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
8233  023402  003003                        BGT     4$              ;;BRANCH IF YES
8234  023404  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
8235  023412  000002                4$:     RTI                     ;;GO BACK TO USER
8236                                ;;**************************************************************
8237                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
8238                                ;*CALL:
8239                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
8240                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
8241                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
8242
8243  023414  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
8244  023416  005046                        CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
8245  023420  012703  023650        1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
8246  023424  022703  023660        2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
8247  023430  101456                        BLOS    4$              ;;BR IF YES
8248  023432  104410                        RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
8249  023434  112613                        MOVB    (SP)+,(R3)      ;;GET CHARACTER
8250  023436  122713  000177        10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
8251  023442  001022                        BNE     5$              ;;BR IF NO
8252  023444  005716                        TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
8253  023446  001007                        BNE     6$              ;;BR IF NO
8254  023450  112767  000134  000170        MOVB    #'\,9$          ;;TYPE A BACK SLASH
8255  023456  104401  023646                TYPE    ,9$
8256  023462  012716  177777                MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
```

```
8257  023466  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
8258  023470  020327  023650                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
8259  023474  103434                          BLO     4$              ;;BR IF YES
8260  023476  111367  000144                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
8261  023502  104401  023646                  TYPE    ,9$             ;;GO TYPE
8262  023506  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
8263  023510  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
8264  023512  001406                          BEQ     7$              ;;BR IF NO
8265  023514  112767  000134  000124          MOVB    #'\,9$          ;;TYPE A BACK SLASH
8266  023522  104401  023646                  TYPE    ,9$
8267  023526  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
8268  023530  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
8269  023534  001003                          BNE     8$              ;;BR IF NO
8270  023536  104401  023660                  TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
8271  023542  000726                          BR      1$              ;;GO START OVER
8272  023544  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
8273  023550  001011                          BNE     3$              ;;BRANCH IF NO
8274  023552  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
8275  023554  104401  001227                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
8276  023560  104401  023650                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
8277  023564  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
8278  023566  104401  001226          4$:     TYPE    ,$QUES          ;;TYPE A '?'
8279  023572  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
8280  023574  111367  000046          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
8281  023600  104401  023646                  TYPE    ,9$
8282  023604  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
8283  023610  001305                          BNE     2$              ;;LOOP IF NOT RETURN
8284  023612  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
8285  023616  104401  001230                  TYPE    ,$LF            ;;TYPE A LINE FEED
8286  023622  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
8287  023624  012603                          MOV     (SP)+,R3        ;;RESTORE R3
8288  023626  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
8289  023630  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
8290  023636  012766  023650  000004          MOV     #$TTYIN,4(SP)
8291  023644  000002                          RTI                     ;;RETURN
8292  023646     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
8293  023647     000                          .BYTE   0               ;;TERMINATOR
8294  023650  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
8295  023660  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
8296  023665     136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
8297  023672  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
8298  023700  020075     000
8299  023703     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
8300  023710  036440  000040
8301                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
8302
8303                                  ;;**************************************************************
8304                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
8305                                  ;*CHANGE IT TO BINARY.
8306                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
8307                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
8308                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
8309                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
8310                                  ;*CALL:
8311                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
8312                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
```

```
8313                                    ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
8314
8315  023714  011646                    $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
8316  023716  016666  000004  000002            MOV     4(SP),2(SP)     ;;INPUT NUMBER
8317  023724  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8318  023726  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8319  023730  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8320  023732  104411                    1$:     RDLIN                   ;;READ AN ASCIZ LINE
8321  023734  012600                            MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
8322  023736  010067  000100                    MOV     R0,5$           ;;AND SAVE IT
8323  023742  005001                            CLR     R1              ;;CLEAR DATA WORD
8324  023744  005002                            CLR     R2
8325  023746  112046                    2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
8326  023750  001420                            BEQ     3$              ;;IF ZERO GET OUT
8327  023752  122716  000060                    CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
8328  023756  003026                            BGT     4$              ;;IS AN OCTAL DIGIT
8329  023760  122716  000067                    CMPB    #'7,(SP)
8330  023764  002423                            BLT     4$
8331  023766  006301                            ASL     R1              ;;*2
8332  023770  006102                            ROL     R2
8333  023772  006301                            ASL     R1              ;;*4
8334  023774  006102                            ROL     R2
8335  023776  006301                            ASL     R1              ;;*8
8336  024000  006102                            ROL     R2
8337  024002  042716  177770                    BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
8338  024006  062601                            ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
8339  024010  000756                            BR      2$              ;;LOOP
8340  024012  005726                    3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
8341  024014  010166  000012                    MOV     R1,12(SP)       ;;SAVE THE RESULT
8342  024020  010267  000026                    MOV     R2,$HIOCT
8343  024024  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
8344  024026  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
8345  024030  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
8346  024032  000002                            RTI                     ;;RETURN
8347  024034  005726                    4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
8348  024036  105010                            CLRB    (R0)            ;;SET A TERMINATOR
8349  024040  104401                            TYPE                    ;;TYPE UP THRU THE BAD CHAR.
8350  024042  000000                    5$:     .WORD   0
8351  024044  104401  001226                    TYPE    ,$QUES          ;;"?" "CR" & "LF"
8352  024050  000730                            BR      1$              ;;TRY AGAIN
8353  024052  000000                    $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
8354                                    .SBTTL  TRAP DECODER
8355
8356                                    ;;**************************************************************
8357                                    ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
8358                                    ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
8359                                    ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
8360                                    ;*GO TO THAT ROUTINE.
8361
8362  024054  010046                    $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
8363  024056  016600  000002                    MOV     2(SP),R0        ;;GET TRAP ADDRESS
8364  024062  005740                            TST     -(R0)           ;;BACKUP BY 2
8365  024064  111000                            MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
8366  024066  006300                            ASL     R0              ;;POSITION FOR INDEXING
8367  024070  016000  024110                    MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
8368  024074  000200                            RTS     R0              ;;GO TO ROUTINE
```

```
8369
8370
8371                                    ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
8372
8373  024076  011646                    $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
8374  024100  016666  000004  000002            MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
8375  024106  000002                            RTI                     ;;RESTORE THE PSW
8376
8377                                    .SBTTL  TRAP TABLE
8378
8379                                    ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
8380                                    ;*BY THE "TRAP" INSTRUCTION.
8381
8382                                    ;       ROUTINE
8383                                    ;       -------
8384  024110  024076                    $TRPAD: .WORD   $TRAP2
8385  024112  022262                            $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
8386  024114  021634                            $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
8387  024116  021610                            $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
8388  024120  021650                            $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
8389  024122  022036                            $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
8390
8391  024124  023062                            $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
8392
8393  024126  023012                            $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
8394  024130  023274                            $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
8395  024132  023414                            $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
8396  024134  023714                            $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
8397                                    .SBTTL  POWER DOWN AND UP ROUTINES
8398
8399                                    ;;**************************************************************
8400                                    ;POWER DOWN ROUTINE
8401  024136  012737  024302  000024    $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
8402  024144  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;;PRIO:7
8403  024152  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8404  024154  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8405  024156  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8406  024160  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8407  024162  010446                            MOV     R4,-(SP)        ;;PUSH R4 ON STACK
8408  024164  010546                            MOV     R5,-(SP)        ;;PUSH R5 ON STACK
8409  024166  017746  154746                    MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
8410  024172  010667  000110                    MOV     SP,$SAVR6       ;;SAVE SP
8411  024176  012737  024210  000024            MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
8412  024204  000000                            HALT
8413  024206  000776                            BR      .-2             ;;HANG UP
8414
8415                                    ;;**************************************************************
8416                                    ;POWER UP ROUTINE
8417  024210  012737  024302  000024    $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
8418  024216  016706  000064                    MOV     $SAVR6,SP       ;;GET SP
8419  024222  005067  000060                    CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
8420  024226  005267  000054            1$:     INC     $SAVR6          ;;WAIT FOR THE INC
8421  024232  001375                            BNE     1$              ;;OF  WORD
8422  024234  012677  154700                    MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
8423  024240  012605                            MOV     (SP)+,R5        ;;POP STACK INTO R5
8424  024242  012604                            MOV     (SP)+,R4        ;;POP STACK INTO R4
```

```
8425  024244  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
8426  024246  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
8427  024250  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
8428  024252  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
8429  024254  012737  024136  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
8430  024262  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
8431  024270  104401                        TYPE                    ;REPORT THE POWER FAILURE
8432  024272  024310                $PWRMG: .WORD   $POWER           ;;POWER FAIL MESSAGE POINTER
8433  024274  012716                        MOV     (PC)+,(SP)      ;;RESTART AT RSTRTA
8434  024276  002740                $PWRAD: .WORD   RSTRTA           ;;RESTART ADDRESS
8435  024300  000002                        RTI
8436  024302  000000                $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
8437  024304  000776                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
8438  024306  000000                $SAVR6: 0                       ;;PUT THE SP HERE
8439  024310  005015  047520  042527 $POWER: .ASCIZ  <15><12>"POWER"
8440  024316  000122
8441                                  .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
```

```
8498  024512  010167  154446                  MOV     R1,$REG1
8499  024516  010267  154444                  MOV     R2,$REG2
8500  024522  010367  154442                  MOV     R3,$REG3
8501  024526  010467  154440                  MOV     R4,$REG4
8502  024532  000207                          RTS     PC              ;RETURN TO PATTERNS TEST
8503
8504                                  ;THIS ROUTINE IS CALLED TO SELECT A NEW LINE NO. BASED ON THE
8505                                  ;VALUE OF THE LINE SELECTION PARAMETER
8506
8507                                          ;CALLING SEQUENCE:
8508
8509                                          ;JSR     PC,SELINE      ;CALL THE ROUTINE
8510                                          ;BR      1$             ;EXIT BRANCH-ROUTINE MOVES THE RETURN
8511                                                                  ;PC AROUND THIS BR IF MORE LINES ARE
8512                                                                  ;YET TO BE TESTED
8513
8514  024534  105767  003505          SELINE: TSTB    LINE+1         ;FIRST TIME THROUGH FOR ANY TEST ?
8515  024540  001010                          BNE     1$             ;BR IF NOT
8516  024542  105167  003477                  COMB    LINE+1         ;SET ENTRY FLAG
8517  024546  012767  000001  002462          MOV     #1,LINMSK      ;INIT SELECT TEST MASK TO TEST LINE 00
8518  024554  105067  003464                  CLRB    LINE           ;START WITH LINE #00
8519  024560  000405                          BR      2$             ;GO TEST FOR LINE #00
8520  024562  105267  003456          1$:     INCB    LINE           ;GENERATE NEW LINE NO.
8521  024566  006367  002444                  ASL     LINMSK         ;SHIFT SELECT MASK TO TEST NXT LINE
8522  024572  001407                          BEQ     3$             ;RETURN TO EXIT BRANCH - ALL LINES DONE
8523  024574  036767  002436  002432  2$:     BIT     LINMSK,LINSEL  ;IS THE LINE SELECTED FOR TEST ??
8524  024602  001767                          BEQ     1$             ;BR IF NOT
8525  024604  062716  000002                  ADD     #2,(SP)        ;MOVE RETURN PC AROUND EXIT BRANCH
8526  024610  000402                          BR      4$             ;RETURN TO TEST SELECTED LINE
8527  024612  005067  003426          3$:     CLR     LINE           ;INIT ENTRY FLAG AND LINE NO. TO 000
8528  024616  142777  000017  002400  4$:     BICB    #17,@DHADR     ;INIT LINE SELECT BITS IN "SCR"
8529  024624  000207                          RTS     PC             ;RETURN TO CALLING TEST
8530
8531                                  ;THIS ROUTINE IS CALLED TO CONVERT EITHER THE "DH" NUMBER OR THE
8532                                  ;"LINE" NUMBER TO TWO ASCII CHARACTERS AND MOVE THEM INTO A
8533                                  ;PARTICULAR MESSAGE BUFFER FOR ERROR REPORTING
8534
8535                                          ;CALLING SEQUENCE
8536
8537                                          ;JSR     R5,SUNUM       ;CALL TO THIS ROUTINE
8538                                          ;ADDR1                  ;ADDRESS OF THE NUMBER TO BE CONVERTED
8539                                          ;ADDR2                  ;ADDRESS OF THE MSG BUFFER SLOT
8540
8541  024626                          SUNUM:
8542  024626  010046                          MOV     R0,-(SP)       ;;PUSH R0 ON STACK
8543  024630  010146                          MOV     R1,-(SP)       ;;PUSH R1 ON STACK
8544  024632  010246                          MOV     R2,-(SP)       ;;PUSH R2 ON STACK
8545  024634  012500                          MOV     (R5)+,R0       ;GET ADDRESS OF NUMBER
8546  024636  012501                          MOV     (R5)+,R1       ;GET MSG BUFFER ADDR
8547  024640  111000                          MOVB    (R0),R0        ;GET NO. TO BE CONVERTED
8548  024642  010002                          MOV     R0,R2          ;SAVE IT IN R2
8549  024644  006202                          ASR     R2             ;SHIFT MSD TO LSD POSITION
8550  024646  006202                          ASR     R2
8551  024650  006202                          ASR     R2
8552  024652  042702  177770                  BIC     #177770,R2     ;CLR JUNK BITS
8553  024656  062702  000060                  ADD     #60,R2         ;MAKE IT ASCII
```

```
8554  024662  110221                        MOVB    R2,(R1)+        ;PUT IT IN MSG BUFFER
8555  024664  042700  177770                BIC     #177770,R0      ;CLR JUNK FROM LSD
8556  024670  062700  000060                ADD     #60,R0          ;MAKE IT ASCII
8557  024674  110011                        MOVB    R0,(R1)         ;PUT LSD IN THE BUFFER
8558  024676  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
8559  024700  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
8560  024702  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
8561  024704  000205                        RTS     R5              ;RETURN TO CALLER
8562
8563                                  ;THIS ROUTINE IS CALLED TO CLEAR THE "CAR" AND "BCR" MEMORIES
8564                                  ;IT ASSUMES THAT THE ADDRESS OF THE "SCR" IS IN R1
8565
8566  024706  005067  154306          CLCABC: CLR     $TMP7           ;INIT A COUNTER
8567  024712  116711  154302          1$:     MOVB    $TMP7,(R1)      ;SELECT A LINE
8568  024716  005061  000006                  CLR     CAR(R1)         ;CLEAR A CAR LOCATION
8569  024722  005061  000010                  CLR     BCR(R1)         ;CLEAR A BCR LOCATION
8570  024726  005267  154266                  INC     $TMP7           ;GENERATE NEW LINE NO.
8571  024732  022767  000020  154260        CMP     #20,$TMP7       ;DONE ALL LINES ?
8572  024740  001364                        BNE     1$              ;BR IF NOT
8573  024742  142711  000017                BICB    #17,(R1)        ;SET "SCR" TO SELECT LINE 00
8574  024746  000207                        RTS     PC              ;RETURN TO CALLER
8575
8576                                  ;THIS ROUTINE IS CALLED TO LOAD THE "BCR" MEMORY WITH ALL ONES
8577                                  ;IT ASSUMES THAT THE ADDRESS OF THE SCR IS IN R1
8578
8579  024750  005067  154244          LDBCR:  CLR     $TMP7           ;INIT A COUNTER
8580  024754  116711  154240          1$:     MOVB    $TMP7,(R1)      ;SELECT A LINE
8581  024760  012761  177777  000010        MOV     #-1,BCR(R1)     ;LOAD BCR LOC. WITH 177777
8582  024766  005267  154226                INC     $TMP7           ;GENERATE NEXT LINE NO.
8583  024772  022767  000020  154220        CMP     #20,$TMP7       ;DONE ALL LINES ?
8584  025000  001365                        BNE     1$              ;BR IF NOT
8585  025002  142711  000017                BICB    #17,(R1)        ;SET "SCR" TO SELECT LINE 00
8586  025006  000207                        RTS     PC              ;RETURN TO CALLER
8587
8588                                  ;THIS ROUTINE CALLED TO SET UP FOR PARITY TESTS
8589
8590  025010  012767  000020  154174  SUPPAR: MOV     #20,$TMP4       ;SET UP FOR 16. LINES
8591  025016  105011                        CLRB    (R1)            ;INIT SCR TO START AT LINE 00
8592  025020  005002                        CLR     R2              ;INIT INDEX REGISTER FOR RBUF (EVEN)
8593  025022  012703  000200                MOV     #200,R3         ;SET UP CONSTANT
8594  025026  012704  000001                MOV     #1,R4           ;INIT INDEX REG FOR RBUF (ODD)
8595  025032  012761  037244  000006  1$:    MOV     #TBUF,CAR(R1)   ;LOAD BUS ADDRESS REWG
8596  025040  016761  154152  000010        MOV     $TMP6,BCR(R1)   ;LOAD BYTE COUNT REG
8597  025046  016761  154142  000004        MOV     $TMP5,LPR(R1)   ;LOAD LINE PARAMETERS
8598  025054  105062  036244                CLRB    RBUF(R2)        ;INIT DATA BYTE IN RBUF TO START AT 000
8599  025060  110364  036244                MOVB    R3,RBUF(R4)     ;SET CONSTANT IN HIGH BYTE
8600  025064  005211                        INC     (R1)            ;SELECT NEXT LINE
8601  025066  005203                        INC     R3              ;GENERATE NEW CONSTANT
8602  025070  062702  000002                ADD     #2,R2           ;UPDATE POINTERS TO RBUF (EVEN/ODD)
8603  025074  062704  000002                ADD     #2,R4
8604  025100  005367  154106                DEC     $TMP4           ;COUNT ONE LINE SETUP
8605  025104  001352                        BNE     1$              ;BR TILL ALL 16. SET UP
8606  025106  012704  027504                MOV     #MULPTB,R4      ;SET UP TABLE POINTER
8607  025112  016724  154100          2$:   MOV     $TMP6,(R4)+     ;SET UP BYTE COUNT ENTRY
8608  025116  022704  027544                CMP     #MULPTB+40,R4   ;SET UP ALL COUNTS ?
8609  025122  001373                        BNE     2$              ;BR IF NOT
```

```
8610  025124  105011                          CLRB    (R1)            ;INIT SCR TO SELECT LINE 00
8611  025126  000207                          RTS     PC              ;RETURN TO PARITY TEST
8612
8613                                  ;THIS ROUTINE AUTOSIZES THE SYSTEM TO DETERMINE THE ADDRESSES AND
8614                                  ;VECTORS OF THE DH11'S AND MODEM CONTROL'S.
8615
8616  025130  010046                  AUTOSZ: MOV     R0,-(SP)
8617  025132  005003                          CLR     R3
8618  025134  012702  027764                  MOV     #DHADRS,R2      ;POINT TO BEGINNING OF TABLE
8619  025140  005022                  25$:    CLR     (R2)+           ;CLEAR AUTOSIZER TABLES.
8620  025142  005203                          INC     R3
8621  025144  020327  000102                  CMP     R3,#102         ;HAVE WE CLEARED ALL ENTRIES?
8622  025150  001373                          BNE     25$             ;BRANCH IF NOT.
8623  025152  013746  000004                  MOV     @#4,-(SP)       ;SAVE TRAP VECTOR.
8624  025156  012737  025264  000004          MOV     #4$,@#4         ;SETUP FOR NON-EXISTENT MEMORY TRAP.
8625  025164  012703  030066                  MOV     #DMADRS,R3      ;SETUP DM ADDRESS TABLE POINTER.
8626  025170  012702  027764                  MOV     #DHADRS,R2      ;SET UP DH ADDRESS TABLE POINTER.
8627
8628  025174  012701  160020                  MOV     #160020,R1      ;R1=FIRST ADDRESS TO BE TESTED.
8629
8630  025200  005711                  1$:     TST     (R1)            ;SEE IF ADDRESS IN R1 RESPONDS.
8631  025202  005761  000016                  TST     16(R1)          ;CHECK TO SEE IF DEVICE IS MODULO 20.
8632  025206  052711  004000                  BIS     #4000,(R1)      ;IF IT IS, CONTINUE
8633                                                                  ;AND CHECK TO SEE
8634  025212  052711  001000                  BIS     #1000,(R1)      ;IF THIS ADDRESS CONTAINS
8635  025216  052711  002000                  BIS     #2000,(R1)      ;A DH-11.
8636  025222  032711  003000                  BIT     #3000,(R1)      ;CHECK TO INSURE THESE BITS SET.
8637  025226  001410                          BEQ     3$              ;IF NOT, BRANCH.
8638                                                                  ;SET THE MAINTENANCE BIT, THE NON-
8639  025230  052711  000400                  BIS     #400,(R1)       ;EXISTENT MEMORY BIT AND THE CLEAR
8640                                                                  ;NON-EXISTENT MEMORY INTERRUPT BIT.
8641  025234  032711  002400                  BIT     #2400,(R1)      ;IS THIS A DH-11? (BITS 8 AND 10 SHOULD
8642                                                                  ;CLEAR IF THIS IS A DH11.)
8643
8644  025240  001003                          BNE     3$              ;IF NOT, CHECK TO SEE IF THIS IS A MODEM CONTROL.
8645  025242  042711  001000                  BIC     #1000,(R1)      ;CLEAR MAINTENANCE BIT.
8646  025246  010122                          MOV     R1,(R2)+        ;SAVE THE ADDRESS IN THE DH ADR TABLE.
8647
8648
8649  025250  020127  163760          3$:     CMP     R1,#163760      ;HAVE WE REACHED THE TOP OF THE FLOATING ADDRESSES.
8650  025254  001406                          BEQ     5$              ;IF YES, GET OUT.
8651  025256  062701  000020                  ADD     #20,R1          ;IF NOT, UPDATE ADDRESS AND
8652  025262  000746                          BR      1$              ;GO CHECK IT.
8653
8654  025264  012716  025250          4$:     MOV     #3$,(SP)        ;IF DH ADDRESS DOES NOT RESPOND, GO TO 3$.
8655  025270  000002                          RTI
8656
8657                                  ;TEST FOR MODEM CONTROL ADDRESS
8658
8659  025272  012737  025324  000004  5$:     MOV     #6$,@#4         ;SETUP FOR NON-EXISTENT MEMORY TRAP.
8660  025300  012701  170500                  MOV     #170500,R1      ;R1=FIRST ADDRESS TO BE TESTED.
8661  025304  005711                  21$:    TST     (R1)            ;SEE IF ADDRESS RESPONDS.
8662  025306  010123                          MOV     R1,(R3)+        ;IF IT DOES, THIS IS A MODEM CONTROL,
8663                                                                  ;SO SAVE THE ADDRESS.
8664  025310  020127  170670          23$:    CMP     R1,#170670      ;HAVE WE REACHED THE TOP OF THE MODEM ADDRESSES?
8665  025314  001406                          BEQ     22$             ;IF YES, GET OUT.
```

```
8666  025316  062701  000010            ADD     #10,R1          ;IF NOT, UPDATE ADDRESS AND
8667  025322  000770                    BR      21$             ;GO CHECK IT.
8668
8669  025324  012716  025310    6$:     MOV     #23$,(SP)       ;IF DM ADDRESS DOES NOT RESPOND, GO TO 23$.
8670  025330  000002                    RTI
8671
8672  025332  012637  000004    22$:    MOV     (SP)+,@#4       ;RESTORE TRAP VECTOR.
8673  025336  162702  027764            SUB     #DHADRS,R2      ;HAVE WE FOUND ANY DH11'S AT ALL?
8674  025342  001003                    BNE     7$              ;IF YES, BRANCH
8675  025344  104401  035537            TYPE    ,MSG1           ;NO DH11'S WERE FOUND,
8676  025350  000000                    HALT
8677
8678  025352  006202            7$:     ASR     R2              ;R2 NOW CONTAINS THE NUMBER
8679  025354  005000                    CLR     R0              ;OF DH'S FOUND.
8680  025356  006100            8$:     ROL     R0              ;FILL R0 WITH 1'S
8681  025360  005200                    INC     R0              ;CORRESPONDING TO
8682  025362  005302                    DEC     R2              ;THE NUMBER OF DH'S
8683  025364  005702                    TST     R2              ;FOUND.
8684  025366  001373                    BNE     8$
8685  025370  010067  002642            MOV     R0,$DHSEL       ;$DHSEL CONTAINS THE DH SELECTION PARAMETER.
8686                                                            ;IE. ALL DH'S FOUND WILL BE TESTED.
8687
8688                    ;FIND DH VECTOR:
8689  025374  012702  027764            MOV     #DHADRS,R2      ;SETUP POINTER TO BEGINNING OF DH
8690  025400  012705  030026            MOV     #DHVEC,R5       ;ADDRESS TABLE AND VECTOR TABLE.
8691  025404  012737  000340  000022    MOV     #340,@#IOTVEC+2 ;SET IOT TRAP PRIORITY TO 7.
8692  025412  012737  025522  000020    MOV     #12$,@#IOTVEC   ;SETUP IOT TRAP VECTOR.
8693  025420  012703  000300            MOV     #300,R3         ;START OF FLOATING VECTORS
8694  025424  012704  000302            MOV     #302,R4         ;PC OF IOT INSTR.
8695
8696  025430  010423            9$:     MOV     R4,(R3)+        ;FILL VECTOR AREA WITH ADDRESS
8697                                                            ;OF NEXT INSTR (.+2)
8698  025432  012724  000004            MOV     #4,(R4)+        ;NEXT INSTRUCTION IS AN IOT TRAP.
8699  025436  022324                    CMP     (R3)+,(R4)+     ;UPDATE R3+R4.
8700  025440  020427  001000            CMP     R4,#1000        ;HAVE WE REACHED TO TOP OF THE
8701                                                            ;VECTOR SPACE?
8702  025444  101771                    BLOS    9$              ;IF NOT, REPEAT PROCESS.
8703
8704  025446  005712            10$:    TST     (R2)            ;HAVE WE CHECK ALL DH'S?
8705  025450  001441                    BEQ     13$             ;IF YES, GET OUT + CHECK FOR MODEM CONTROL'S VECTORS.
8706
8707  025452  005067  152320            CLR     PS              ;ZERO CPU PRIORITY.
8708  025456  052772  001000  000000    BIS     #1000,@(R2)     ;SET MAINTENANCE BIT
8709  025464  052772  000300  000000    BIS     #300,@(R2)      ;ATTEMPT TO CAUSE RECEIVER
8710                                                            ;INTERRUPT.
8711  025472  005000                    CLR     R0
8712
8713  025474  005200            11$:    INC     R0              ;WAIT...
8714  025476  001376                    BNE     11$
8715  025500  104401  035567            TYPE    ,MSG2           ;ERROR MSG-NO DH RECEIVER INTERRUPT OCCURRED.
8716  025504  052772  004000  000000    BIS     #4000,@(R2)     ;DO A MASTER CLEAR
8717  025512  042772  001000  000000    BIC     #1000,@(R2)     ;CLEAR MAINTENANCE BIT
8718  025520  000752                    BR      10$
8719
8720  025522  011601            12$:    MOV     (SP),R1
8721  025524  042701  000007            BIC     #7,R1           ;CLEAR GARBAGE.
```

```
8722  025530  010125                        MOV     R1,(R5)+        ;SAVE VECTOR ADDRESS.
8723  025532  022626                        CMP     (SP)+,(SP)+     ;POP STACK
8724  025534  012716  025446                MOV     #10$,(SP)       ;SETUP FOR RETURN.
8725  025540  052772  004000  000000        BIS     #4000,@(R2)     ;DO A MASTER CLEAR
8726  025546  042732  001000                BIC     #1000,@(R2)+    ;CLEAR MAINTENANCE BIT.
8727  025552  000002                        RTI
8728
8729                                  ;FIND MODEM CONTROL VECTORS:
8730
8731  025554  012702  030066        13$:    MOV     #DMADRS,R2      ;SET POINTERS TO BEGINNING OF
8732  025560  012705  030130                MOV     #DMVEC,R5       ;ADR TABLE & VECTOR TABLE.
8733  025564  012737  025646  000020        MOV     #16$,@#IOTVEC   ;SET IOT TRAP VECTOR.
8734
8735  025572  005712                14$:    TST     (R2)            ;HAVE WE CHECKED ALL DM'S?
8736  025574  001441                        BEQ     17$             ;IF YES, GET OUT.
8737  025576  005067  152174                CLR     PS              ;ZERO CPU PRIORITY
8738  025602  052772  001000  000000        BIS     #1000,@(R2)     ;SET MAINTENANCE BIT.
8739  025610  052772  000300  000000        BIS     #300,@(R2)      ;ATTEMPT TO CAUSE INTERRUPT.
8740  025616  005000                        CLR     R0
8741
8742  025620  005200                15$:    INC     R0              ;WAIT....
8743  025622  001376                        BNE     15$
8744  025624  104401  035634                TYPE    ,MSG3           ;ERROR MSG - NO MODEM CONTROL INTERRUPT OCCURRED.
8745  025630  052772  004000  000000        BIS     #4000,@(R2)     ;CLEAR BITS PREVIOUSLY SET.
8746  025636  042772  001000  000000        BIC     #1000,@(R2)     ;CLEAR MAINTENANCE BIT.
8747  025644  000752                        BR      14$
8748
8749  025646  011601                16$:    MOV     (SP),R1
8750  025650  162701  000004                SUB     #4,R1           ;CALCULATE VECTOR ADDRESS.
8751  025654  010125                        MOV     R1,(R5)+        ;SAVE VECTOR ADDRESS.
8752  025656  022626                        CMP     (SP)+,(SP)+     ;POP STACK.
8753  025660  012716  025572                MOV     #14$,(SP)       ;SETUP FOR RETURN.
8754  025664  052772  004000  000000        BIS     #4000,@(R2)     ;CLEAR BITS PREVIOUSLY SET.
8755  025672  042732  001000                BIC     #1000,@(R2)+    ;CLEAR MAINTENANCE BIT AND
8756                                                                ;POINT TO NEXT MODEM CONTROL ADDRESS.
8757  025676  000002                        RTI
8758
8759  025700  012737  021000  000020  17$:  MOV     #$SCOPE,@#IOTVEC  ;RESTORE IOT VECTOR FOR SCOPE ROUTINE.
8760  025706  012600                        MOV     (SP)+,R0        ;RESTORE R0.
8761  025710  012703  000300                MOV     #300,R3         ;START OF FLOATING VECTORS.
8762  025714  012704  000302                MOV     #302,R4
8763
8764  025720  010423                18$:    MOV     R4,(R3)+        ;FILL VECTOR AREA WITH ADDRESS OF NEXT
8765                                                                ;INSTRUCTION (.+2).
8766  025722  012724  000000                MOV     #0,(R4)+        ;NEXT INSTRUCTION IS A HALT.
8767  025726  022324                        CMP     (R3)+,(R4)+     ;UPDATE R3 & R4.
8768  025730  020427  001000                CMP     R4,#1000        ;ARE WE DONE?
8769  025734  101771                        BLOS    18$             ;IF NOT, REPEAT UNTIL ADDRESSES
8770                                                                ;377 TO 777 ARE DONE.
8771  025736  013701  030026                MOV     @#DHVEC,R1      ;LET R1 POINT TO 1ST DH VECTOR ADDRESS.
8772  025742  005737  030030                TST     @#DHVEC+2       ;IS THERE MORE THAN ONE VECTOR?
8773  025746  001403                        BEQ     26$             ;BRANCH IF NOT.
8774  025750  163701  030030                SUB     @#DHVEC+2,R1    ;DETERMINE NUMBER OF ADDRESSES
8775                                                                ;BETWEEN DH VECTORS (10(8) OR 20(8)).
8776  025754  005401                        NEG     R1              ;MAKE R1 POSITIVE.
8777  025756  010167  002246        26$:    MOV     R1,ADRVEC       ;SAVE THAT NUMBER.
```

```
 8778  025762  032777  000002  153150          BIT     #BIT1,@SWR      ;SHOULD DEVICE MAP BE TYPED OUT?
 8779  025770  001442                          BEQ     20$             ;IF NOT, RETURN.
 8780  025772  104401                          TYPE                    ;TYPEOUT MAP OF DH & MODEM CONTROL'S
 8781  025774  035707                          DEVMAP                  ;FOUND.
 8782  025776  012701  027764                  MOV     #DHADRS,R1      ;R1=BEGINNING OF DH ADDRESS TABLE.
 8783  026002  012702  030026                  MOV     #DHVEC,R2       ;R2=BEGINNING OF DH VECTOR TABLE.
 8784  026006  012703  030066                  MOV     #DMADRS,R3      ;R3=BEGINNING OF MODEM CONTROL ADDRESS TABLE.
 8785  026012  012704  030130                  MOV     #DMVEC,R4       ;R4=BEGINNING OF MODEM CONTROL VECTOR TABLE.
 8786
 8787  026016  012146                  19$:    MOV     (R1)+,-(SP)     ;MOVE DATA TO BE TYPED
 8788  026020  104403                          TYPOS                   ;TYPE DATA
 8789  026022     006                          .BYTE   6
 8790  026023     001                          .BYTE   1
 8791  026024  012246                          MOV     (R2)+,-(SP)     ;MOVE DATA TO BE TYPED
 8792  026026  104403                          TYPOS                   ;TYPE DATA
 8793  026030     005                          .BYTE   5
 8794  026031     000                          .BYTE   0
 8795  026032  104401  035703                  TYPE    SPACE
 8796  026036  012346                          MOV     (R3)+,-(SP)     ;MOVE DATA TO BE TYPED.
 8797  026040  104403                          TYPOS                   ;TYPE DATA.
 8798  026042     006                          .BYTE   6
 8799  026043     001                          .BYTE   1
 8800  026044  104401  035703                  TYPE    SPACE
 8801  026050  012446                          MOV     (R4)+,-(SP)     ;MOVE DATA TO BE TYPED.
 8802  026052  104403                          TYPOS                   ;TYPE DATA.
 8803  026054     005                          .BYTE   5
 8804  026055     000                          .BYTE   0
 8805  026056  104401                          TYPE                    ;TYPE A CARRIAGE RETURN & LINE FEED.
 8806  026060  001227                          $CRLF
 8807  026062  005711                          TST     (R1)            ;HAVE WE TYPED ALL DH ENTRIES?
 8808  026064  001354                          BNE     19$             ;IF NOT, DO IT AGAIN.
 8809  026066  005713                          TST     (R3)            ;HAVE WE TYPED ALL DM ENTRIES?
 8810  026070  001352                          BNE     19$             ;IF NOT - ONE MORE TIME.
 8811  026072  104401  001227                  TYPE    ,$CRLF
 8812  026076  000207                  20$:    RTS     PC              ;IF YES, GO BACK TO MAIN PROGRAM.
 8813
 8814                                  ;THIS ROUTINE IS USED TO ACCEPT INPUT PARAMETERS FROM THE CONSOLE
 8815                                  ;TELETYPE
 8816
 8817  026100  104401                  INPARA: TYPE
 8818  026102  035442                          VCWC                    ;"ASK FOR NO. ADDRESSES BETWEEN VECTORS"
 8819  026104  104412                          RDOCT                   ;READ OCTAL NO. FM TTY
 8820  026106  012600                          MOV     (SP)+,R0        ;GET THE NO. HE TYPED
 8821  026110  001407                          BEQ     3$              ;BR IF HE TYPED <CR>
 8822  026112  022700  000010                  CMP     #10,R0          ;10(8) ADDRESSES BETWEEN VECTORS ?
 8823  026116  001406                          BEQ     4$              ;BR IF YES
 8824  026120  022700  000020                  CMP     #20,R0          ;20(8) ADDRESSES BETWEEN VECTORS ??
 8825  026124  001403                          BEQ     4$              ;BR IF YES
 8826  026126  000764                          BR      INPARA          ;ASK ALL OVER AGAIN
 8827  026130  012700  000020          3$:     MOV     #20,R0          ;SET UP CONSTANT FOR 20(8) ADDRESSES
 8828  026134  000207                  4$:     RTS     PC              ;RETURN TO CALLER
 8829
 8830
 8831
 8832  026136  012700  177777          INPARC: MOV     #-1,R0          ;SET FLAG IN R0
 8833  026142  000167  154012                  JMP     BEGINA          ;GO ASK FOR SELECT PARAMETER
```

```
8834
8835  026146  012767  177777  001546  INPARX: MOV     #-1,VCFLG       ;SET SETUP FLAG
8836  026154  000167  154000                  JMP     BEGINA          ;GO START UP
8837
8838  026160  013701  027764          INPAR:  MOV     @#DHADRS,R1     ;MOVE ADDRESS OF FIRST DH INTO R1.
8839  026164  032777  000001  152746          BIT     #BIT0,@SWR      ;ARE PARAMETERS TO BE INPUT MANUALLY?
8840  026172  001405                          BEQ     2$              ;BRANCH IF NOT.
8841  026174  104401                  1$:     TYPE                    ;ASK FOR DEVICE ADDRESS
8842  026176  035017                          INMSG1
8843  026200  104412                          RDOCT                   ;READ IN WHAT IS TYPED
8844  026202  012601                          MOV     (SP)+,R1        ;GET THE NO. HE TYPED
8845  026204  001403                          BEQ     INPAR1          ;BR IF DEFAULT
8846  026206  004767  000160          2$:     JSR     PC,CHKADR       ;GO CHECK VALIDITY OF THE ADDR
8847  026212  000770                          BR      1$              ;ERROR BRANCH
8848
8849  026214  013701  030026          INPAR1: MOV     @#DHVEC,R1      ;MOVE FIRST DH VECTOR INTO R1.
8850  026220  032777  000001  152712          BIT     #BIT0,@SWR      ;ARE PARAMETERS TO BE INPUT MANUALLY?
8851  026226  001405                          BEQ     2$              ;BRANCH IF NOT.
8852  026230  104401                  1$:     TYPE                    ;ASK FOR VECTOR ADDRESS
8853  026232  035063                          INMSG2
8854  026234  104412                          RDOCT                   ;READ IN WHAT HE TYPES
8855  026236  012601                          MOV     (SP)+,R1        ;GET THE ADDRESS
8856  026240  001403                          BEQ     INPAR3          ;BR IF DEFAULT
8857  026242  004767  000240          2$:     JSR     PC,CHKVCT       ;GO CHECK VALIDITY OF VECTOR
8858  026246  000770                          BR      1$              ;ERROR BRANCH
8859
8860  026250  013701  030236          INPAR3: MOV     @#$DHSEL,R1     ;MOVE DEVICE SELECTION PARAMETER INTO R1.
8861  026254  005700                          TST     R0              ;DID WE START AT 210?
8862  026256  100404                          BMI     2$              ;BRANCH IF YES.
8863  026260  032777  000001  152652          BIT     #BIT0,@SWR      ;IS PARAMETER TO BE INPUT MANUALLY?
8864  026266  001405                          BEQ     1$              ;BRANCH IF NOT.
8865  026270  104401                  2$:     TYPE                    ;ASK FOR DEVICE SELECTION PARAMETER
8866  026272  035132                          INMSG3
8867  026274  104412                          RDOCT                   ;READ IN WHAT HE TYPES
8868  026276  012601                          MOV     (SP)+,R1        ;GET THE SELECT PARAMETER
8869  026300  001402                          BEQ     INPAR4          ;BR IF  DEFAULT
8870  026302  010167  000724          1$:     MOV     R1,DHSEL        ;SET UP DH11 SELECTION PARAMETER
8871
8872  026306  005700                  INPAR4: TST     R0              ;DID WE START AT 210?
8873  026310  100404                          BMI     3$              ;BRANCH IF YES.
8874  026312  032777  000001  152620          BIT     #BIT0,@SWR      ;IS LINE SELECT PARAMETER TO BE INPUT MANUALLY?
8875  026320  001410                          BEQ     1$              ;BRANCH IF NO.
8876  026322  104401                  3$:     TYPE                    ;ASK FOR LINE SELECT PARAMETER
8877  026324  035330                          INMSG6
8878  026326  104412                          RDOCT                   ;GET WHAT HE TYPES
8879  026330  012601                          MOV     (SP)+,R1        ;GET PARAMETER
8880  026332  001403                          BEQ     1$              ;BR IF DEFAULT
8881  026334  010167  000674                  MOV     R1,LINSEL       ;SET UP LINE SELECT PARAMETER
8882  026340  000403                          BR      2$              ;CONTINUE
8883  026342  012767  177777  000664  1$:     MOV     #-1,LINSEL      ;SET UP DEFAULT (ALL LINES)
8884  026350  032777  000400  152562  2$:     BIT     #BIT8,@SWR      ;HALT AFTER SET UP ??
8885  026356  001403                          BEQ     EXPAR           ;BR IF  NOT
8886  026360  104401                          TYPE                    ;TYPE CONTINUE MESSAGE PRIOR TO HALTING
8887  026362  035372                          INMSG7
8888  026364  000000                          HALT                    ;DEPRESS CONTINUE TO RESUME TESTING
8889  026366  000167  154242          EXPAR:  JMP     START2          ;GO START UP THE PROGRAM
```

```
8890
8891
8892  026372  020127  160020          CHKADR: CMP     R1,#160020      ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
8893  026376  002001                          BGE     1$              ;BR IF YES
8894  026400  000437                          BR      4$              ;BR IF NOT
8895  026402  020127  160420          1$:     CMP     R1,#160420      ;IS IT BELOW THE HIGH LIMIT?
8896  026406  002401                          BLT     2$              ;BR IF YES
8897  026410  000433                          BR      4$              ;BR IF NOT
8898  026412  032701  000017          2$:     BIT     #17,R1          ;CORRECT BOUNDARY ?
8899  026416  001030                          BNE     4$              ;BR IF NOT
8900  026420  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC AROUND ERROR BRANCH
8901  026424  012702  027622                  MOV     #DHADTB,R2      ;POINT TO BEGIN OF ADDR TABLE
8902  026430  032777  000001  152502          BIT     #BIT0,@$SWR     ;ARE WE AUTOSIZING?
8903  026436  001011                          BNE     3$              ;BRANCH IF NOT.
8904  026440  012703  027764                  MOV     #DHADRS,R3      ;POINT TO BEGINNING OF AUTOSIZER
8905                                                                  ;DH ADDRESS TABLE.
8906  026444  016704  001566                  MOV     $DHSEL,R4
8907  026450  012322                  6$:     MOV     (R3)+,(R2)+     ;MOVE CONTENTS OF AUTOSIZER DH TABLE
8908                                                                  ;TO THE TABLE USED BY PROGRAM.
8909  026452  006204                          ASR     R4
8910  026454  005704                          TST     R4              ;HAVE WE MOVED ALL TABLE ENTRIES?
8911  026456  001374                          BNE     6$              ;BRANCH IF NOT--ONE MORE TIME.
8912  026460  000411                          BR      5$              ;RETURN TO INPUT ROUTINES.
8913  026462  010122                  3$:     MOV     R1,(R2)+        ;SETR UP A TABLE ENTRY
8914  026464  062701  000020                  ADD     #20,R1          ;GENERATE NEXT DH11 ADDR
8915  026470  022702  027662                  CMP     #DHADTB+40,R2   ;END OF TABLE ?
8916  026474  001372                          BNE     3$              ;BR IF NOT
8917  026476  000402                          BR      5$              ;RETURN TO INPUT ROUTINES
8918  026500  104401                  4$:     TYPE                    ;TELL HIM HE GOOFED
8919  026502  035203                          INMSG4
8920  026504  000207                  5$:     RTS     PC              ;RETURN TO INPUT ROUTINES
8921
8922  026506  020127  000300          CHKVCT: CMP     R1,#300         ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
8923  026512  002001                          BGE     1$              ;BR IF YES
8924  026514  000436                          BR      4$              ;BR IF NOT
8925  026516  020127  001000          1$:     CMP     R1,#1000        ;IS IT BELOW THE HIGH LIMIT?
8926  026522  002401                          BLT     2$              ;BR IF YES
8927  026524  000432                          BR      4$              ;BR IF NOT
8928  026526  032701  000007          2$:     BIT     #7,R1           ;CORRECT BOUNDARY ?
8929  026532  001027                          BNE     4$              ;BR IF NOT
8930  026534  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC AROUND ERROR BRANCH
8931  026540  012702  027662                  MOV     #DHVCTB,R2      ;POINT TO BEGIN OF VECTOR TABLE
8932  026544  032777  000001  152366          BIT     #BIT0,@$SWR     ;ARE WE AUTOSIZING?
8933  026552  001011                          BNE     3$              ;BRANCH IF NOT.
8934  026554  012703  030026                  MOV     #DHVEC,R3       ;POINT TO BEGINING OF AUTOSIZER
8935                                                                  ;DH VECTOR TABLE.
8936  026560  016704  001452                  MOV     $DHSEL,R4
8937  026564  012322                  6$:     MOV     (R3)+,(R2)+     ;MOVE CONTENTS OF AUTOSIZER VECTOR
8938                                                                  ;TABLE TO TABLE USED BY PROGRAM.
8939  026566  006204                          ASR     R4
8940  026570  005704                          TST     R4              ;HAVE WE MOVED ALL TABLE ENTRIES?
8941  026572  001374                          BNE     6$              ;BRANCH IF NOT--ONE MORE TIME.
8942  026574  000410                          BR      5$              ;RETURN TO INPUT ROUTINES.
8943  026576  010122                  3$:     MOV     R1,(R2)+        ;SETR UP A TABLE ENTRY
8944  026600  060001                          ADD     R0,R1           ;GENERATE NEXT DH11 ADDR
8945  026602  022702  027722                  CMP     #DHVCTB+40,R2   ;END OF TABLE ?
```

```
                                        BNE     3$              ;BR IF NOT
                                        BR      5$              ;RETURN TO INPUT ROUTINES
                                4$:     TYPE                    ;TELL HIM HE GOOFED
                                        INMSG5
                                5$:     RTS     PC              ;RETURN TO INPUT ROUTINES

                                ;THESE TWO ROUTINES SERVICE UNEXPECTED BUS ERROR AND RSVD INSTR TRAPS

                                BUSER:  MOV     #340,$TMP0      ;SAVE THE PSW
                                        MOV     SP,$REG6        ;SAVE THE SP
                                        MOV     (SP)+,R1        ;GET THE TRAP PC
                                        MOV     (SP)+,R2        ;GET THE TRAP PSW
                                        MOVB    $TSTNM,R0       ;GET TEST NO.
                                        MOV     #STACK,SP       ;RESET THE STACK POINTER
                                        JSR     PC,SUER3        ;GO SET UP ERROR INFO
                                        MOV     #1$,$LPERR      ;ALWAYS COME BACK TO 1$
                                        ERROR   27              ;UNEXPECTED BUS ERROR TRAP
                                1$:     RESET                   ;PREPARE TO RESTART
                                        JSR     PC,CHPS1        ;GO CLEAR PSW
                                        JMP     REST1           ;GO RESTART THE PROGRAM

                                RESERR: MOV     #340,$TMP0      ;SAVE THE PSW
                                        MOV     SP,$REG6        ;SAVE THE SP
                                        MOV     (SP)+,R1        ;GET THE TRAP PC
                                        MOV     (SP)+,R2        ;GET THE TRAP PSW
                                        MOVB    $TSTNM,R0       ;GET TEST NO.
                                        MOV     #STACK,SP       ;RESET THE STACK POINTER
                                        JSR     PC,SUER3        ;GO SET UP ERROR INFO
                                        MOV     #1$,$LPERR      ;ALWAYS COME BACK TO 1$
                                        ERROR   30              ;UNEXPECTED RSVD INSTR ERROR TRAP
                                1$:     RESET                   ;PREPARE TO RESTART
                                        JSR     PC,CHPS1        ;GO CLEAR PSW
                                        JMP     REST1           ;GO RESTART THE PROGRAM

                                ;THIS ROUTINE IS CALLED WHEN A TEST NEEDS TO RESTORE THE TRAP
                                ;CATCHER IN THE DH11 VECTOR

                                RESTRP: MOV     DHVCT,R3        ;GET VECTOR ADDRESS
                                        MOV     R3,(R3)         ;RESTORE THE TRAP CATCHER
                                        ADD     #2,(R3)+
                                        CLR     (R3)+
                                        MOV     R3,(R3)
                                        ADD     #2,(R3)+
                                        CLR     (R3)+
                                        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

                                TIMEIT: INC     TIMEB           ;COUNT B
                                        BNE     1$              ;BR IF NOT ZERO
                                        DEC     TIMEA           ;COUNT TIME A
                                        BNE     1$              ;BR IF NO TIMEOUT
```

```
                                        ADD     #2,(SP)         ;MOVE RETURN PC TO ALLOW ERROR REPORT
                                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

                                SETALL: MOV     #20,$TMP6       ;SET UP SIXTEEN LINES
                                        CLR     R2              ;INIT A TABLE INDEX REG
                                        MOV     #200,R3         ;SET UP TO GENERATE HI BYTE OF EXPECTED DATA
                                        MOV     #1,R4           ;SET UP INDEX REG TO ODD BYTES
                                        CLR     (R1)            ;START WITH LINE 00
                                1$:     MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDR REG
                                        MOV     #-400,BCR(R1)   ;SET UP BYTE COUNT REG
                                        MOV     #31403,LPR(R1)  ;SET UP FOR 4800 BAUD/8 BIT CHARS
                                        CLRB    RBUF(R2)        ;START WITH DATA CHAR OF 000
                                        MOVB    R3,RBUF(R4)     ;SET UP HIGH BYTE OF EXPECTED DATA
                                        INC     (R1)            ;GEN NEW LINE NO. IN SCR
                                        INC     R3              ;UPDATE THE POINTERS AND DATA
                                        ADD     #2,R2
                                        ADD     #2,R4
                                        DEC     $TMP6           ;COUNT ONE LINE DONE
                                        BNE     1$              ;BR TIL ALL 16 SET UP
                                        MOV     LINSEL,LINACT   ;SET SOFTWARE FLAG FOR ALL LINES ACTIVE
                                        CLR     (R1)            ;PUT SCR REG BACK TO LINE 00
                                        RTS     PC              ;RETURN TO AUTO ECHO TEST


                                ;THIS ROUTINE IS CALLED TO SET PSW PRIORITY TO 000 IN ORDER
                                ;TO BE LSI11 COMPATIBLE

                                CHPS1:  MOV     #0,-(SP)        ;NEW PSW
                                        MOV     #1$,-(SP)       ;NEW PC
                                        RTI                     ;CHANGE PSW
                                1$:     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

                                CHPS2:  MOV     #340,-(SP)      ;NEW PSW
                                        MOV     #1$,-(SP)       ;NEW PC
                                        RTI                     ;CHANGE THE PSW
                                1$:     RTS     PC              ;RETURN TO CALLING TEST

                                ;THIS ROUTINE IS ALSO FOR LSI11 COMPATIBILITY AND IT IS CALLED
                                ;TO SAVE THE PSW IN "$TMP0"

                                SAPS:   CLR     -(SP)           ;TEMP STORAGE TO SAVE PSW
                                        MOV     34,-(SP)        ;SAVE TRAP VECTOR POINTER
                                        MOV     #1$,34          ;GO TO 1$ ON TRAP
                                        TRAP                    ;GO TO IT
                                1$:     MOV     2(SP),6(SP)     ;GET PSW SAVED
                                        MOV     #2$,(SP)        ;GO TO 2$ ON RTI
                                        RTI
                                2$:     MOV     (SP)+,34        ;RESTORE VECTOR
                                        MOV     (SP)+,$TMP0     ;FINALLY SAVE PSW IN $TMP0
                                        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
```

```
 9115
 9116                           ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
 9117                           ;TEST (ALL 1'S DATA TABLE)
 9118
 9119  027342  100377           AETAB:  100377          ;TEST DATA FOR LINE 00
 9120  027344  100777                   100777          ;TEST DATA FOR LINE 01
 9121  027346  101377                   101377
 9122  027350  101777                   101777
 9123  027352  102377                   102377
 9124  027354  102777                   102777
 9125  027356  103377                   103377
 9126  027360  103777                   103777
 9127  027362  104377                   104377
 9128  027364  104777                   104777
 9129  027366  105377                   105377
 9130  027370  105777                   105777
 9131  027372  106377                   106377
 9132  027374  106777                   106777
 9133  027376  107377                   107377
 9134  027400  107777                   107777          ;TEST DATA FOR LINE 17
 9135
 9136                           ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
 9137                           ;TEST (ALL 0'S DATA TABLE)
 9138
 9139  027402  100000           AETABO: 100000          ;TEST DATA FOR LINE 00
 9140  027404  100400                   100400          ;TEST DATA FOR LINE 01
 9141  027406  101000                   101000
 9142  027410  101400                   101400
 9143  027412  102000                   102000
 9144  027414  102400                   102400
 9145  027416  103000                   103000
 9146  027420  103400                   103400
 9147  027422  104000                   104000
 9148  027424  104400                   104400
 9149  027426  105000                   105000
 9150  027430  105400                   105400
 9151  027432  106000                   106000
 9152  027434  106400                   106400
 9153  027436  107000                   107000
 9154  027440  107400                   107400          ;TEST DATA FOR LINE 17
 9155
 9156                           ;THIS TABLE USED BY THE AUTO ECHO TEST 2 TO RESET ACTIVE BIT WHEN A
 9157                           ;LINE IS DONE
 9158
 9159  027442  000001           LINBIT: BIT00                   ;DEACTIVATE LINE 00
 9160  027444  000002                   BIT01                   ;DEACTIVATE LINE 01
 9161  027446  000004                   BIT02
 9162  027450  000010                   BIT03
 9163  027452  000020                   BIT04
 9164  027454  000040                   BIT05
 9165  027456  000100                   BIT06
 9166  027460  000200                   BIT07
 9167  027462  000400                   BIT08
 9168  027464  001000                   BIT09
 9169  027466  002000                   BIT10
 9170  027470  004000                   BIT11
```

```
9171  027472  010000                  BIT12
9172  027474  020000                  BIT13
9173  027476  040000                  BIT14
9174  027500  100000                  BIT15                   ;DEACTIVATE LINE 17
9175
9176  027502  000000          LINACT: 0                       ;MAINTAINS STATUS OF ACTIVE LINES
9177                                                          ;DURING AUTO ECHO TEST 2
9178
9179
9180                          ;THIS TABLE CONTAINS 16. COUNTERS USED BYN THE MULTI-LINE
9181                          ;PARITY TEST TO KEEP TRACK OF TOTAL CHARS RECEIVED
9182
9183  027504  000020          MULPTB: .BLKW   16.             ;SIXTEEN WORD COUNTERS TABLE
9184
9185                          ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE BREAK BIT
9186                          ;TEST
9187
9188  027544  120000          BRKTAB: 120000          ;TEST DATA FOR LINE 00
9189  027546  120400                  120400          ;TEST DATA FOR LINE 01
9190  027550  121000                  121000
9191  027552  121400                  121400
9192  027554  122000                  122000
9193  027556  122400                  122400
9194  027560  123000                  123000
9195  027562  123400                  123400
9196  027564  124000                  124000
9197  027566  124400                  124400
9198  027570  125000                  125000
9199  027572  125400                  125400
9200  027574  126000                  126000
9201  027576  126400                  126400
9202  027600  127000                  127000
9203  027602  127400                  127400          ;TEST DATA FOR LINE 17
9204
9205  027604  131177          RGMSK1: 131177          ;MASK TO SPECIFY R/W BITS FOR NORMAL "SCR" REG TEST
9206  027606  046600          RGMSK2: 46600           ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR NORMAL MODE TEST
9207  027610  177767          RGMSK3: 177767          ;MASK TO SPECIFY R/W BITS IN "LPR"
9208  027612  177777          RGMSK4: 177777          ;MASK TO SPECIFY R/W BITS IN "BKR"
9209  027614  100077          RGMSK5: 100077          ;MASK TO SPECIFY R/W BITS IN "SSR"
9210  027616  042200          RGMSK6: 42200           ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR MAINT. MODE TEST
9211  027620  030100          INTMSK: 30100           ;MASK USED TO SELECT INTR BITS TO TEST
9212
9213                          ;DH11 ADDRESS TABLE - THIS TABLE CONTAINS THE "SCR" ADDRESS FOR UP TO
9214                          ;SIXTEEN DH11'S
9215
9216  027622  160020          DHADTB: 160020          ;ADDRESS OF FIRST DH11
9217  027624  160040                  160040          ;ADDRESS OF SECOND DH11
9218  027626  160060                  160060
9219  027630  160100                  160100
9220  027632  160120                  160120
9221  027634  160140                  160140
9222  027636  160160                  160160
9223  027640  160200                  160200
9224  027642  160220                  160220
9225  027644  160240                  160240
9226  027646  160260                  160260
```

```
 9227  027650  160300                    160300
 9228  027652  160320                    160320
 9229  027654  160340                    160340
 9230  027656  160360                    160360
 9231  027660  160400                    160400          ;ADDRESS OF THE LAST DH11
 9232
 9233                                ;DH11 VECTOR TABLE - THIS TABLE CONTAINS THE VECTOR ADDRESSES FOR UP
 9234                                ;TO SIXTEEN DH11'S
 9235
 9236  027662  000330                DHVCTB: 330             ;ADDRESS OF VECTOR FOR FIRST DH11
 9237  027664  000350                        350             ;ADDRESS OF VECTOR FOR SECOND DH11
 9238  027666  000370                        370
 9239  027670  000410                        410
 9240  027672  000430                        430
 9241  027674  000450                        450
 9242  027676  000470                        470
 9243  027700  000510                        510
 9244  027702  000530                        530
 9245  027704  000550                        550
 9246  027706  000570                        570
 9247  027710  000610                        610
 9248  027712  000630                        630
 9249  027714  000650                        650
 9250  027716  000670                        670
 9251  027720  000710                        710             ;ADDRESS OF VECTOR FOR LAST DH11
 9252
 9253  027722  000000                VCFLG:  0               ;VECTOR SET UP FLAGG
 9254
 9255
 9256                                ;BR PRIORITY LEVEL TABLE - THIS TABLE CONTAINS THE PRIORITY LEVELS
 9257                                ;FOR UP TO SIXTEEN DH11'S - THE RCVR LEVEL IS STORED IN THE LOW BYTE
 9258                                ;AND THE XMTTR LEVEL IN THE HIGH BYTE
 9259
 9260  027724  120240                BRLVL:  120240          ;BRLEVELS FOR FIRST DH11
 9261  027726  120240                        120240          ;BR LEVELS FOR SECOND DH11
 9262  027730  120240                        120240
 9263  027732  120240                        120240
 9264  027734  120240                        120240
 9265  027736  120240                        120240
 9266  027740  120240                        120240
 9267  027742  120240                        120240
 9268  027744  120240                        120240
 9269  027746  120240                        120240
 9270  027750  120240                        120240
 9271  027752  120240                        120240
 9272  027754  120240                        120240
 9273  027756  120240                        120240
 9274  027760  120240                        120240
 9275  027762  120240                        120240          ;BR LEVELS FOR LAST DH11
 9276
 9277                                ;THIS DH ADDRESS TABLE IS FILLED BY THE AUTOSIZER.
 9278
 9279  027764                        DHADRS:
 9280  027764  000000                .WORD   0
 9281  027766  000000                .WORD   0
 9282  027770  000000                .WORD   0
```

```
 9283  027772  000000                .WORD   0
 9284  027774  000000                .WORD   0
 9285  027776  000000                .WORD   0
 9286  030000  000000                .WORD   0
 9287  030002  000000                .WORD   0
 9288  030004  000000                .WORD   0
 9289  030006  000000                .WORD   0
 9290  030010  000000                .WORD   0
 9291  030012  000000                .WORD   0
 9292  030014  000000                .WORD   0
 9293  030016  000000                .WORD   0
 9294  030020  000000                .WORD   0
 9295  030022  000000                .WORD   0
 9296  030024  000000                .WORD   0
 9297
 9298                                ;THIS DH VECTOR TABLE IS FILLED BY THE AUTOSIZER.
 9299
 9300  030026                        DHVEC:
 9301  030026  000000                .WORD   0
 9302  030030  000000                .WORD   0
 9303  030032  000000                .WORD   0
 9304  030034  000000                .WORD   0
 9305  030036  000000                .WORD   0
 9306  030040  000000                .WORD   0
 9307  030042  000000                .WORD   0
 9308  030044  000000                .WORD   0
 9309  030046  000000                .WORD   0
 9310  030050  000000                .WORD   0
 9311  030052  000000                .WORD   0
 9312  030054  000000                .WORD   0
 9313  030056  000000                .WORD   0
 9314  030060  000000                .WORD   0
 9315  030062  000000                .WORD   0
 9316  030064  000000                .WORD   0
 9317
 9318                                ;THIS DM ADDRESS TABLE IS FILLED BY THE AUTOSIZER.
 9319
 9320  030066                        DMADRS:
 9321  030066  000000                .WORD   0
 9322  030070  000000                .WORD   0
 9323  030072  000000                .WORD   0
 9324  030074  000000                .WORD   0
 9325  030076  000000                .WORD   0
 9326  030100  000000                .WORD   0
 9327  030102  000000                .WORD   0
 9328  030104  000000                .WORD   0
 9329  030106  000000                .WORD   0
 9330  030110  000000                .WORD   0
 9331  030112  000000                .WORD   0
 9332  030114  000000                .WORD   0
 9333  030116  000000                .WORD   0
 9334  030120  000000                .WORD   0
 9335  030122  000000                .WORD   0
 9336  030124  000000                .WORD   0
 9337  030126  000000                .WORD   0
 9338
```

```
                                ;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.
```

```
 9395            000400                  STEP=400
 9396            000001                  LINENA=1
 9397            000002                  TRMRDY=2
 9398            000004                  RS=4
 9399            000010                  SECTX=10
 9400            000020                  SECRX=20
 9401            000040                  CS=40
 9402            000100                  CO=100
 9403            000200                  RING=200
 9404            002000                  CLRMUX=2000
 9405
 9406  030256    000000                  TDATA1: 0                       ;DATA BUFFER FOR BASIC DATA TEST
 9407  030260    000037                  TDATA2: 37                      ;TEST DATA FOR FIVE BIT CHAR
 9408  030262    000077                          77                      ;TEST DATA FOR SIX BIT CHAR
 9409  030264    000177                          177                     ;TEST DATA FOR SEVEN BIT CHAR
 9410  030266    000377                          377                     ;TEST DATA FOR EIGHT BIT CHAR
 9411  030270    000000                  TITFLG: 0                       ;FLAG TO ALLOW PRINTING TITLE ONLY ONCE
 9412  030272    000000                  TIMEA:  0                       ;GENERAL PURPOSE TIMERS
 9413  030274    000000                  TIMEB:  0
 9414  030276    000000                  TIMEC:  0                       ;TIMER FOR TIMING TESTS
 9415  030300    000000                  TNULL:  0                       ;CONTAINS TWO NULL CHARS USED BY BREAK TEST
 9416
 9417
 9418
```

```
                                        ;;*****************************************************************
                                        ;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
```

```
 9475  030662  020122  050123  041505
 9476  030670  043111  042511  020104
 9477  030676  042522  000107
 9478
 9479                                  ;INFORMATION FOR MESSAGE 4
 9480
 9481  030702  044514  042516  050040  EM4:    .ASCIZ  'LINE PARAMETER REGISTER ERROR'
 9482  030710  051101  046501  052105
 9483  030716  051105  051040  043505
 9484  030724  051511  042524  020122
 9485  030732  051105  047522  000122
 9486
 9487                                  ;INFORMATION FOR MESSAGE 5
 9488
 9489  030740  051102  040505  020113  EM5:    .ASCIZ  'BREAK CONTROL REGISTER ERROR'
 9490  030746  047503  052116  047522
 9491  030754  020114  042522  044507
 9492  030762  052123  051105  042440
 9493  030770  051122  051117     000
 9494
 9495                                  ;INFORMATION FOR MESSAGE 6
 9496
 9497  030775     123  046111  020117  EM6:    .ASCIZ  'SILO STATUS REGISTER ERROR'
 9498  031002  052123  052101  051525
 9499  031010  051040  043505  051511
 9500  031016  042524  020122  051105
 9501  031024  047522  000122
 9502
 9503                                  ;INFORMATION FOR MESSAGE 7
 9504
 9505  031030  052503  051122  047105  EM7:    .ASCIZ  'CURRENT ADDRESS REGISTER ERROR - LINE #XX'
 9506  031036  020124  042101  051104
 9507  031044  051505  020123  042522
 9508  031052  044507  052123  051105
 9509  031060  042440  051122  051117
 9510  031066  026440  046040  047111
 9511  031074  020105  054043  000130
 9512
 9513                                  ;INFORMATION FOR MESSAGE 10
 9514
 9515  031102  054502  042524  041440  EM10:   .ASCIZ  'BYTE COUNTER REGISTER ERROR - LINE #XX'
 9516  031110  052517  052116  051105
 9517  031116  051040  043505  051511
 9518  031124  042524  020122  051105
 9519  031132  047522  020122  020055
 9520  031140  044514  042516  021440
 9521  031146  054130     000
 9522
 9523                                  ;INFORMATION FOR MESSAGE 11
 9524
 9525  031151     125  042516  050130  EM11:   .ASCIZ  'UNEXPECTED DH11 RCVR INTERRUPT'
 9526  031156  041505  042524  020104
 9527  031164  044104  030461  051040
 9528  031172  053103  020122  047111
 9529  031200  042524  051122  050125
 9530  031206  000124
```

```
                                        ;INFORMATION FOR MESSAGE 12
                                        ;INFORMATION FOR MESSAGE 13
                                        ;INFORMATION FOR MESSAGE 14
                                        ;INFORMATION FOR MESSAGE 15
                                        ;INFORMATION FOR MESSAGE 16
                                        ;INFORMATION FOR MESSAGE 17
```

```
9531
9532
9533
9534  031210  047125  054105  042520  EM12:   .ASCIZ  'UNEXPECTED DH11 XMITTR INTERRUPT'
9535  031216  052103  042105  042040
9536  031224  030510  020061  046530
9537  031232  052111  051124  044440
9538  031240  052116  051105  052522
9539  031246  052120     000
9540
9541
9542
9543  031251     103  040510  020122  EM13:   .ASCIZ  'CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT'
9544  031256  053101  044501  040514
9545  031264  046102  020105  040506
9546  031272  046111  042105  052040
9547  031300  020117  042507  042516
9548  031306  040522  042524  051040
9549  031314  053103  020122  047111
9550  031322  042524  051122  050125
9551  031330  000124
9552
9553
9554
9555  031332  051124  047101  046523  EM14:   .ASCIZ  'TRANSMITTER NPR LOGIC ERROR - LINE #  '
9556  031340  052111  042524  020122
9557  031346  050116  020122  047514
9558  031354  044507  020103  051105
9559  031362  047522  020122  020055
9560  031370  044514  042516  021440
9561  031376  020040     000
9562
9563
9564
9565  031401     130  044515  052124  EM15:   .ASCIZ  'XMITTR FAILED TO INTERRUPT - LINE #  '
9566  031406  020122  040506  046111
9567  031414  042105  052040  020117
9568  031422  047111  042524  051122
9569  031430  050125  020124  020055
9570  031436  044514  042516  021440
9571  031444  020040     000
9572
9573
9574
9575  031447     122  053103  020122  EM16:   .ASCIZ  'RCVR FAILED TO INTERRUPT'
9576  031454  040506  046111  042105
9577  031462  052040  020117  047111
9578  031470  042524  051122  050125
9579  031476  000124
9580
9581
9582
9583  031500  051124  047101  046523  EM17:   .ASCIZ  'TRANSMITTER TIMING ERROR - LINE #  '
9584  031506  052111  042524  020122
9585  031514  044524  044515  043516
9586  031522  042440  051122  051117
```

```
 9587  031530  026440  046040  047111
 9588  031536  020105  020043  000040
 9589  031544  024040  041520  020051  DH6:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  SPEED   TIMEB   TIMEC'
 9590  031552  020040  024040  051520
 9591  031560  020051  020040  024040
 9592  031566  050123  020051  020040
 9593  031574  052040  051505  020124
 9594  031602  020040  042504  040526
 9595  031610  051104  020040  050123
 9596  031616  042505  020104  020040
 9597  031624  044524  042515  020102
 9598  031632  020040  044524  042515
 9599  031640  000103
 9600
 9601                          ;INFORMATION FOR MESSAGE 20
 9602
 9603  031642  042522  042503  053111  EM20:   .ASCIZ  'RECEIVER TIMING ERROR - LINE #  '
 9604  031650  051105  052040  046511
 9605  031656  047111  020107  051105
 9606  031664  047522  020122  020055
 9607  031672  044514  042516  021440
 9608  031700  020040     000
 9609
 9610                          ;INFORMATION FOR MESSAGE 21
 9611
 9612  031703     122  053103  020122  EM21:   .ASCIZ  'RCVR FAILED TO INTERRUPT - LINE #  '
 9613  031710  040506  046111  042105
 9614  031716  052040  020117  047111
 9615  031724  042524  051122  050125
 9616  031732  020124  020055  044514
 9617  031740  042516  021440  020040
 9618  031746     000
 9619
 9620                          ;INFORMATION FOR MESSAGE 22
 9621
 9622  031747     103  040510  020122  EM22:   .ASCIZ  'CHAR AVAIL FAILED TO SET ON TIME - LINE #  '
 9623  031754  053101  044501  020114
 9624  031762  040506  046111  042105
 9625  031770  052040  020117  042523
 9626  031776  020124  047117  052040
 9627  032004  046511  020105  020055
 9628  032012  044514  042516  021440
 9629  032020  020040     000
 9630
 9631                          ;INFORMATION FOR MESSAGE 23
 9632
 9633  032023     102  051501  041511  EM23:   .ASCIZ  'BASIC DATA TEST ERROR - LINE #  '
 9634  032030  042040  052101  020101
 9635  032036  042524  052123  042440
 9636  032044  051122  051117  026440
 9637  032052  046040  047111  020105
 9638  032060  020043  000040
 9639  032064  024040  041520  020051  DH7:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG   WAS     S/B'
 9640  032072  020040  024040  051520
 9641  032100  020051  020040  024040
 9642  032106  050123  020051  020040
```

```
 9643  032114  052040  051505  020124
 9644  032122  020040  042504  040526
 9645  032130  051104  020040  044103
 9646  032136  046122  043516  020040
 9647  032144  053440  051501  020040
 9648  032152  020040  051440  041057
 9649  032160     000
 9650
 9651                                  ;INFORMATION FOR MESSAGE 24
 9652
 9653  032161     101  052125  020117  EM24:   .ASCIZ  'AUTO ECHO TEST ERROR - LINE #  '
 9654  032166  041505  047510  052040
 9655  032174  051505  020124  051105
 9656  032202  047522  020122  020055
 9657  032210  044514  042516  021440
 9658  032216  020040     000
 9659
 9660                                  ;INFORMATION FOR MESSAGE 25
 9661
 9662  032221     102  042522  045501  EM25:   .ASCIZ  'BREAK BIT TEST ERROR - LINE #  '
 9663  032226  041040  052111  052040
 9664  032234  051505  020124  051105
 9665  032242  047522  020122  020055
 9666  032250  044514  042516  021440
 9667  032256  020040     000
 9668
 9669                                  ;INFORMATION FOR MESSAGE 26
 9670
 9671  032261     110  046101  026506  EM26:   .ASCIZ  'HALF-DUPLEX TEST ERROR - LINE #  '
 9672  032266  052504  046120  054105
 9673  032274  052040  051505  020124
 9674  032302  051105  047522  020122
 9675  032310  020055  044514  042516
 9676  032316  021440  020040     000
 9677
 9678                                  ;INFORMATION FOR MESSAGE 27
 9679
 9680  032323     125  042516  050130  EM27:   .ASCIZ  'UNEXPECTED BUS ERROR TRAP'
 9681  032330  041505  042524  020104
 9682  032336  052502  020123  051105
 9683  032344  047522  020122  051124
 9684  032352  050101     000
 9685  032355     040  050050  024503  DH3:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   TRPPC   TRPPS '
 9686  032362  020040  020040  050050
 9687  032370  024523  020040  020040
 9688  032376  051450  024520  020040
 9689  032404  020040  042524  052123
 9690  032412  020040  052040  050122
 9691  032420  041520  020040  052040
 9692  032426  050122  051520  000040
 9693                                  .EVEN
 9694  032434  001116  001202  001176  DT3:    .WORD   $ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,0
 9695  032442  001162  001164  001166
 9696  032450  000000
 9697
 9698                                  ;INFORMATION    FOR MESSAGE 30
```

```
9699
9700  032452  047125  054105  042520  EM30:   .ASCIZ  'UNEXPECTED RSVD INSTR TRAP'
9701  032460  052103  042105  051040
9702  032466  053123  020104  047111
9703  032474  052123  020122  051124
9704  032502  050101     000
9705
9706                                  ;INFORMATION FOR MESSAGE 31
9707
9708  032505     101  052125  020117  EM31:   .ASCIZ  'AUTO ECHO DATA COMPARE ERROR - LINE #  '
9709  032512  041505  047510  042040
9710  032520  052101  020101  047503
9711  032526  050115  051101  020105
9712  032534  051105  047522  020122
9713  032542  020055  044514  042516
9714  032550  021440  020040     000
9715  032555     040  050050  024503  DH4:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   WASADR  SBADR    WAS     S/B'
9716  032562  020040  020040  050050
9717  032570  024523  020040  020040
9718  032576  051450  024520  020040
9719  032604  020040  042524  052123
9720  032612  020040  053440  051501
9721  032620  042101  020122  051440
9722  032626  040502  051104  020040
9723  032634  020040  040527  020123
9724  032642  020040  020040  027523
9725  032650  000102
9726
9727                                  ;INFORMATION FOR MESSAGE 32
9728
9729  032652  052501  047524  042440  EM32:   .ASCIZ  'AUTO ECHO TEST TIMEOUT - LINE #  '
9730  032660  044103  020117  042524
9731  032666  052123  052040  046511
9732  032674  047505  052125  026440
9733  032702  046040  047111  020105
9734  032710  020043  000040
9735  032714  024040  041520  020051  DH5:    .ASCIZ  ' (PC)   (LPRG)   TEST'
9736  032722  020040  046050  051120
9737  032730  024507  020040  052040
9738  032736  051505  000124
9739                                  .EVEN
9740  032742  001116  001202  001206  DT4:    .WORD   $ERRPC,$TMP0,$TMP2,0
9741  032750  000000
9742
9743                                  ;INFORMATION FOR MESSAGE 33
9744
9745  032752  040520  044522  054524  EM33:   .ASCIZ  'PARITY LOGIC TEST ERROR - LINE #  '
9746  032760  046040  043517  041511
9747  032766  052040  051505  020124
9748  032774  051105  047522  020122
9749  033002  020055  044514  042516
9750  033010  021440  020040     000
9751
9752                                  ;INFORMATION FOR MESSAGE 34
9753
9754  033015     115  046125  044524  EM34:   .ASCIZ  'MULTI-LINE PARITY DATA TEST ERROR - LINE #   - SUBTEST #  '
```

```
 9755  033022  046055  047111  020105
 9756  033030  040520  044522  054524
 9757  033036  042040  052101  020101
 9758  033044  042524  052123  042440
 9759  033052  051122  051117  026440
 9760  033060  046040  047111  020105
 9761  033066  020043  020040  020055
 9762  033074  052523  052102  051505
 9763  033102  020124  020043  000040
 9764
 9765                          ;INFORMATION FOR MESSAGE 35
 9766
 9767  033110  052515  052114  026511  EM35:   .ASCIZ  'MULTI-LINE PARITY DATA TEST TIMEOUT'
 9768  033116  044514  042516  050040
 9769  033124  051101  052111  020131
 9770  033132  040504  040524  052040
 9771  033140  051505  020124  044524
 9772  033146  042515  052517  000124
 9773  033154  024040  041520  020051  DH14:   .ASCIZ  ' (PC)   (LPRG)  ACTLIN'
 9774  033162  020040  046050  051120
 9775  033170  024507  020040  041501
 9776  033176  046124  047111     000
 9777          033204                  .EVEN
 9778  033204  001116  001202  001210  DT6:    .WORD   $ERRPC,$TMP0,$TMP3,0
 9779  033212  000000
 9780
 9781                          ;INFORMATION FOR MESSAGE 36
 9782
 9783  033214  044103  051101  040440  EM36:   .ASCIZ  'CHAR AVAILABLE TIMEOUT'
 9784  033222  040526  046111  041101
 9785  033230  042514  052040  046511
 9786  033236  047505  052125     000
 9787
 9788                          ;INFORMATION FOR MESSAGE 37
 9789
 9790  033243     104  052101  020101  EM37:   .ASCIZ  'DATA COMPARE ERROR - LINE #  '
 9791  033250  047503  050115  051101
 9792  033256  020105  051105  047522
 9793  033264  020122  020055  044514
 9794  033272  042516  021440  020040
 9795  033300     000
 9796
 9797                          ;INFORMATION FOR MESSAGE 40
 9798
 9799  033301     102  043125  042506  EM40:   .ASCIZ  'BUFFER ACTIVE REG ERROR - LINE #  '
 9800  033306  020122  041501  044524
 9801  033314  042526  051040  043505
 9802  033322  042440  051122  051117
 9803  033330  026440  046040  047111
 9804  033336  020105  020043  000040
 9805
 9806                          ;INFORMATION FOR MESSAGE 41
 9807
 9808  033344  041522  051126  043040  EM41:   .ASCIZ  'RCVR FALSE INTERRUPT'
 9809  033352  046101  042523  044440
 9810  033360  052116  051105  052522
```

```
 9811  033366  052120     000
 9812
 9813                                ;INFORMATION FOR MESSAGE 42
 9814
 9815  033371     123  046111  020117  EM42:   .ASCIZ  'SILO OVERFLOW ERROR'
 9816  033376  053117  051105  046106
 9817  033404  053517  042440  051122
 9818  033412  051117     000
 9819
 9820                                ;INFORMATION FOR MESSAGE 43
 9821
 9822  033415     123  046111  020117  EM43:   .ASCIZ  'SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT'
 9823  033422  053117  051105  046106
 9824  033430  053517  043040  044501
 9825  033436  042514  020104  047524
 9826  033444  043440  047105  051105
 9827  033452  052101  020105  041522
 9828  033460  051126  044440  052116
 9829  033466  051105  052522  052120
 9830  033474     000
 9831
 9832                                ;INFORMATION FOR MESSAGE 44
 9833
 9834  033475     116  047117  042440  EM44:   .ASCIZ  'NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT'
 9835  033502  020130  042515  047515
 9836  033510  054522  043040  044501
 9837  033516  042514  020104  047524
 9838  033524  043440  047105  051105
 9839  033532  052101  020105  046530
 9840  033540  052111  051124  044440
 9841  033546  052116  051105  052522
 9842  033554  052120     000
 9843
 9844                                ;INFORMATION FOR MESSAGE 45
 9845
 9846  033557     130  044515  020124  EM45:   .ASCIZ  'XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT'
 9847  033564  047504  042516  043040
 9848  033572  044501  042514  020104
 9849  033600  047524  043440  047105
 9850  033606  051105  052101  020105
 9851  033614  046530  052111  051124
 9852  033622  044440  052116  051105
 9853  033630  052522  052120     000
 9854
 9855                                ;INFORMATION FOR MESSAGE 46
 9856
 9857  033635     103  051125  042522  EM46:   .ASCIZ  'CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE #  '
 9858  033642  052116  040440  042104
 9859  033650  042122  051523  046440
 9860  033656  046505  051117  020131
 9861  033664  040520  052124  051105
 9862  033672  051516  052040  051505
 9863  033700  020124  051105  047522
 9864  033706  020122  020055  044514
 9865  033714  042516  021440  020040
 9866  033722     000
```

```
 9867  033723     040  050050  024503  DH10:   .ASCIZ  ' (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS     S/B'
 9868  033730  020040  046040  047111
 9869  033736  053505  020122  050040
 9870  033744  052101  051124  020116
 9871  033752  020040  042524  052123
 9872  033760  020040  042040  053105
 9873  033766  042101  020122  051040
 9874  033774  043505  042101  020122
 9875  034002  020040  040527  020123
 9876  034010  020040  020040  027523
 9877  034016  000102
 9878                          .EVEN
 9879  034020  001116  001202  001204  DT5:    .WORD   $ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4,0
 9880  034026  001162  001164  001166
 9881  034034  001170  001172  000000
 9882
 9883                          ;INFORMATION FOR MESSAGE 47
 9884
 9885  034042  054502  042524  041440  EM47:   .ASCIZ  'BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE #  '
 9886  034050  052517  052116  046440
 9887  034056  046505  051117  020131
 9888  034064  040520  052124  051105
 9889  034072  051516  052040  051505
 9890  034100  020124  051105  047522
 9891  034106  020122  020055  044514
 9892  034114  042516  021440  020040
 9893  034122     000
 9894                          ;INFORMATION FOR MESSAGE 50
 9895
 9896  034123     124  051505  020124  EM50:   .ASCIZ  'TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  '
 9897  034130  044524  042515  052517
 9898  034136  020124  040527  052111
 9899  034144  047111  020107  047506
 9900  034152  020122  046530  052111
 9901  034160  042040  047117  020105
 9902  034166  020055  044514  042516
 9903  034174  021440  020040     000
 9904
 9905                          ;INFORMATION FOR MESSAGE 51
 9906
 9907  034201     116  051120  046040  EM51:   .ASCIZ  'NPR LOGIC TEST 2 ERROR'
 9908  034206  043517  041511  052040
 9909  034214  051505  020124  020062
 9910  034222  051105  047522  000122
 9911  034230  024040  041520  020051  DH11:   .ASCIZ  ' (PC)   LINACT  LINCHK   TEST   DEVADR  REGADR   WAS     S/B'
 9912  034236  020040  044514  040516
 9913  034244  052103  020040  044514
 9914  034252  041516  045510  020040
 9915  034260  052040  051505  020124
 9916  034266  020040  042504  040526
 9917  034274  051104  020040  042522
 9918  034302  040507  051104  020040
 9919  034310  053440  051501  020040
 9920  034316  020040  051440  041057
 9921  034324     000
 9922
```

```
                                        ;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
```

```
 9979                                   ;INFORMATION FOR MESSAGE 60
 9980
 9981  034666  020040  024040  041520   DH15:   .ASCIZ  /   (PC)  DEVADR    LINE   /
 9982  034674  020051  042040  053105
 9983  034702  042101  020122  020040
 9984  034710  046040  047111  020105
 9985  034716  020040     000
 9986          034722                   .EVEN
 9987                                   ;MISCELLANEOUS MESSAGES
 9988
 9989  034722  005015  040515  047111   TITLE:  .ASCIZ  <15><12>'MAINDEC-11-DZDHM-C  DH11 DIAGNOSTIC'<15><12>
 9990  034730  042504  026503  030461
 9991  034736  042055  042132  046510
 9992  034744  041455  020040  044104
 9993  034752  030461  042040  040511
 9994  034760  047107  051517  044524
 9995  034766  006503  000012
 9996  034772  005015  042524  052123   TITLE2: .ASCIZ  <15><12>'TESTING DH11 #  '<15><12>
 9997  035000  047111  020107  044104
 9998  035006  030461  021440  020040
 9999  035014  005015     000
10000  035017     015  052012  050131   INMSG1: .ASCIZ  <15><12>'TYPE SCR ADDRESS FOR FIRST DH11'<15><12>
10001  035024  020105  041523  020122
10002  035032  042101  051104  051505
10003  035040  020123  047506  020122
10004  035046  044506  051522  020124
10005  035054  044104  030461  005015
10006  035062     000
10007  035063     015  052012  050131   INMSG2: .ASCIZ  <15><12>'TYPE VECTOR ADDRESS FOR FIRST DH11'<15><12>
10008  035070  020105  042526  052103
10009  035076  051117  040440  042104
10010  035104  042522  051523  043040
10011  035112  051117  043040  051111
10012  035120  052123  042040  030510
10013  035126  006461  000012
10014  035132  005015  054524  042520   INMSG3: .ASCIZ  <15><12>'TYPE DH11 DEVICE SELECTION PARAMETER'<15><12>
10015  035140  042040  030510  020061
10016  035146  042504  044526  042503
10017  035154  051440  046105  041505
10018  035162  044524  047117  050040
10019  035170  051101  046501  052105
10020  035176  051105  005015     000
10021  035203     015  044412  053116   INMSG4: .ASCIZ  <15><12>'INVALID DH11 SCR ADDRESS - TRY AGAIN'<15><12>
10022  035210  046101  042111  042040
10023  035216  030510  020061  041523
10024  035224  020122  042101  051104
10025  035232  051505  020123  020055
10026  035240  051124  020131  043501
10027  035246  044501  006516  000012
10028  035254  005015  047111  040526   INMSG5: .ASCIZ  <15><12>'INVALID DH11 VECTOR ADDRESS - TRY AGAIN'<15><12>
10029  035262  044514  020104  044104
10030  035270  030461  053040  041505
10031  035276  047524  020122  042101
10032  035304  051104  051505  020123
10033  035312  020055  051124  020131
10034  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
```

```
10091  036002  052116  047522  020114
10092  036010  020040  047515  042504
10093  036016  020115  047503  052116
10094  036024  047522     114
10095  036027     015  040412  051104          .ASCIZ  <15><12>'ADRS   VECT    ADRS      VECT'<15><12><15><12>
10096  036034  020123  020040  042526
10097  036042  052103  020040  020040
10098  036050  042101  051522  020040
10099  036056  020040  020040  042526
10100  036064  052103  005015  005015
10101  036072     000
10102  036073     015  046412  042117  MSG4:   .ASCIZ  <15><12>'MODEM CONTROL ERROR, RUN DZDHK DIAGNOSTIC'<15><12>
10103  036100  046505  041440  047117
10104  036106  051124  046117  042440
10105  036114  051122  051117  020054
10106  036122  052522  020116  055104
10107  036130  044104  020113  044504
10108  036136  043501  047516  052123
10109  036144  041511  005015     000
10110  036151     015  047012  020117  MSG5:   .ASCIZ  <15><12>/NO MODEM CONTROL FOUND/<15><12>
10111  036156  047515  042504  020115
10112  036164  047503  052116  047522
10113  036172  020114  047506  047125
10114  036200  006504  000012
10115                               .EVEN
10116                               ;SIXTEEN CHAR COUNTERS USED BY THE AUTO ECHO TEST #3
10117
10118  036204  000020               RCNT:   .BLKW   16.
10119
10120                               ;256. WORD RECEIVER INPUT BUFFER
10121
10122  036244  000400               RBUF:   .BLKW   256.
10123
10124
10125                               ;256(10) BYTE TRANSMITTER OUTPUT DATA BUFFER
10126
10127                               .EVEN
10128  037244  000400               TBUF:   .BLKB   256.
10129
10130          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
 GIN    002156
 GINA   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
REST1   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)
```