DH11

RELIABILITY DIAGNOSTIC

MD-11-DZDHN-B

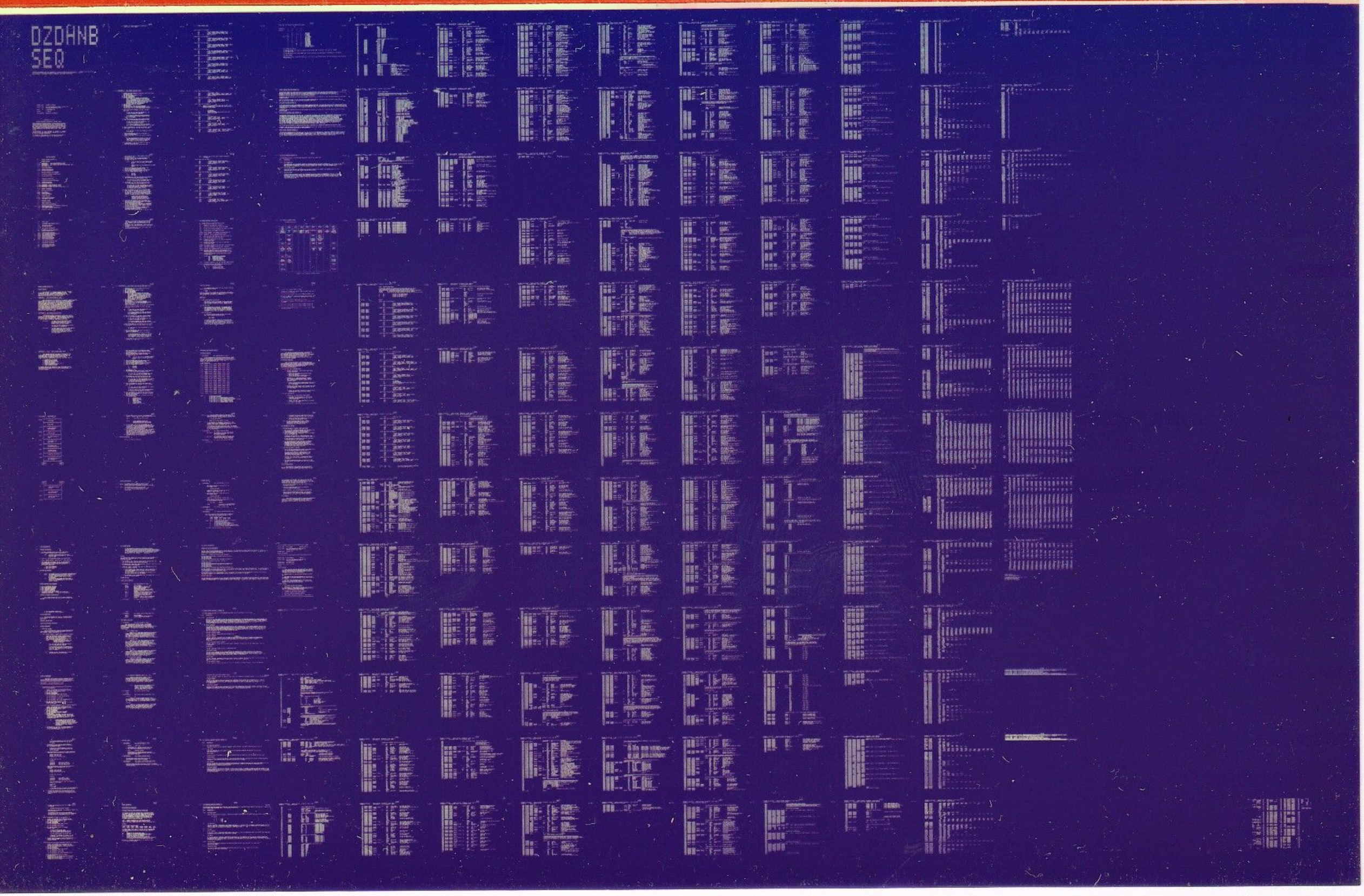
EP DZDHN B DL A

OCT 1976

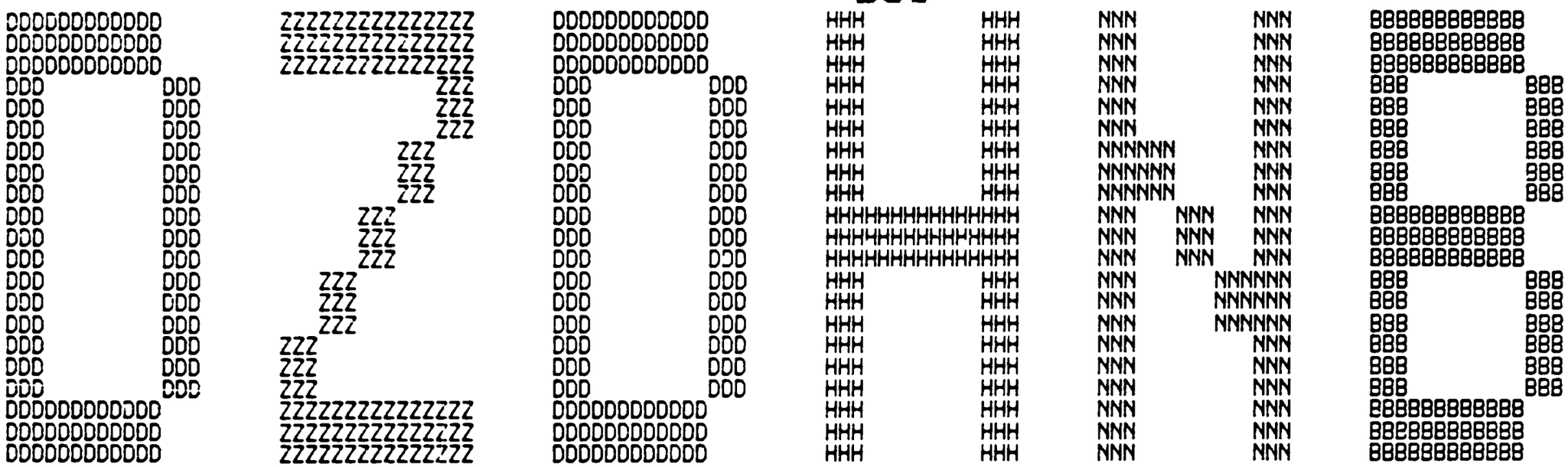
FICHE 1 OF 1

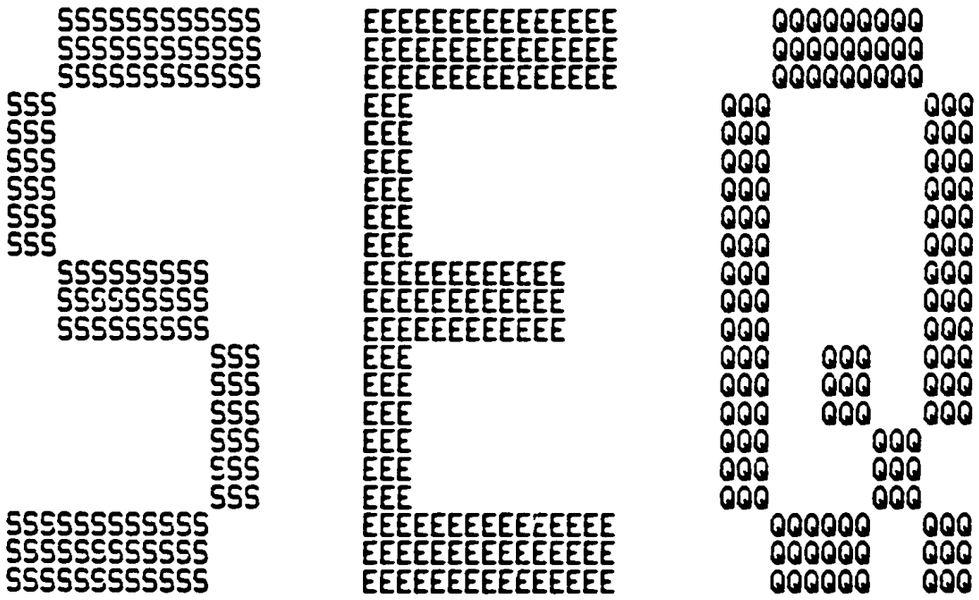
MADE IN USA

DZDHNB
SEQ



B01





START User WATSON [1,2] Job DZDHNB Seq. 7172 Date 21-Feb-77 10:15:10 Monitor RX212A SYS #514/546 *START*
File: DSKZ:DZDHNB.SEQ<055>[400,3722] Created: 15-Feb-77 13:54:11 Printed: 21-Feb-77 10:15:11
QUEUE Switches: /FILE:ASCII /COPIES:1 /SPACING:1 /LIMIT:328 /FORMS:NORMAL

.REM ↑

PRODUCT CODE: MAINDEC-11-DZDHN-B

PRODUCT NAME: DH11 DATA RELIABILITY TESTS / SINGLE LINE ECHO AND PATTERNS/CABLE TESTS

DATE: 21-SEPTEMBER-1976

AUTHOR: E. CROWLEY

MAINTAINED BY: DIAGNOSTIC ENGINEERING

TABLE OF CONTENTS

1.0 GENERAL PROGRAM DESCRIPTION

1.1 PROGRAM PURPOSE

"MD-11-DZDHN" IS A GENERAL PURPOSE TEST AND EXERCISER PROGRAM FOR THE DH11, 16. LINE ASYNCHRONOUS LINE MULTIPLEXOR. IT CONSISTS OF THREE INDEPENDENT SUB-PROGRAMS THAT MAY BE USED FOR ACCEPTANCE TESTING, INSTALLATION CHECKOUT, AND CORRECTIVE MAINTENANCE OF THE DH11 SUB-SYSTEM.

1.1.1 SUBPROGRAM 1 DH11 DATA RELIABILITY TESTS

ONCE CONFIGURED BY THE AUTOSIZER OR BY INITIAL CONSOLE DIALOGUE THIS PROGRAM CAN TEST UP TO 16. DH11'S. ALL LINES ON EACH DH11 ARE TESTED (ONE AT A TIME) WITH ALL COMBINATIONS OF LINE PARAMETERS (BAUD RATE,CHAR LENGTH,PARITY ETC.) BY TRANSMITTING AND RECEIVING A BINARY COUNT PATTERN. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE DEVICE AS THEY OCCUR AND ALSO LOGGED IN ERROR STATISTICS TABLES. AT THE COMPLETION OF TESTING FOR EACH DH11 THESE ERROR STATISTICS TABLES ARE DUMPED ON THE CONSOLE DEVICE TO PROVIDE HISTORICAL EVIDENCE OF THE DATA RELIABILITY OF EACH DH11. REFER TO SECTION 4.0 FOR A DETAILED DESCRIPTION OF THE ERROR STATISTICS PROVIDED. THIS SUB-PROGRAM IS NORMALLY SELECTED FOR OVERALL DH CHECKOUT.

1.1.2 SUB-PROGRAM 2 DH11 SINGLE LINE ECHO TEST

THIS PROGRAM PROVIDES THE MEANS OF TESTING ANY LINE ON ANY DH11 BY USING AN ASYNCHRONOUS TERMINAL DEVICE (VT50,LA36 ETC) CONNECTED TO THE LINE UNDER TEST. THIS SUB-PROGRAM WOULD NORMALLY BE SELECTED WHEN A PROBLEM IS ISOLATED TO A SPECIFIC LINE. IT HAS TWO MODES OF OPERATION, SEND MODE OR ECHO MODE:

SEND MODE: THE USER TYPES AN ASCIZ BUFFER IN ON THE CONSOLE DEVICE AND THEN TYPES A UNIQUE CONTROL CHARACTER TO SEND THIS BUFFER TO THE DH11 TEST TERMINAL.

THE USER CAN THEN COMPARE THE TWO IMAGES FOR ACCURACY OF TRANSMISSION.

ECHO MODE: THE USER TYPES IN ON THE DH11 TEST TERMINAL AND CAN OBSERVE EACH CHAR TYPED BEING ECHOED ON THE TERMINAL. BY TYPING A UNIQUE CONTROL CHARACTER THE PROGRAM WILL ECHO THE ENTIRE BUFFER TYPED IN UP TO THAT POINT.

1.1.3 SUBPROGRAM 3 DH11 DATA PATTERNS/CABLE TESTS

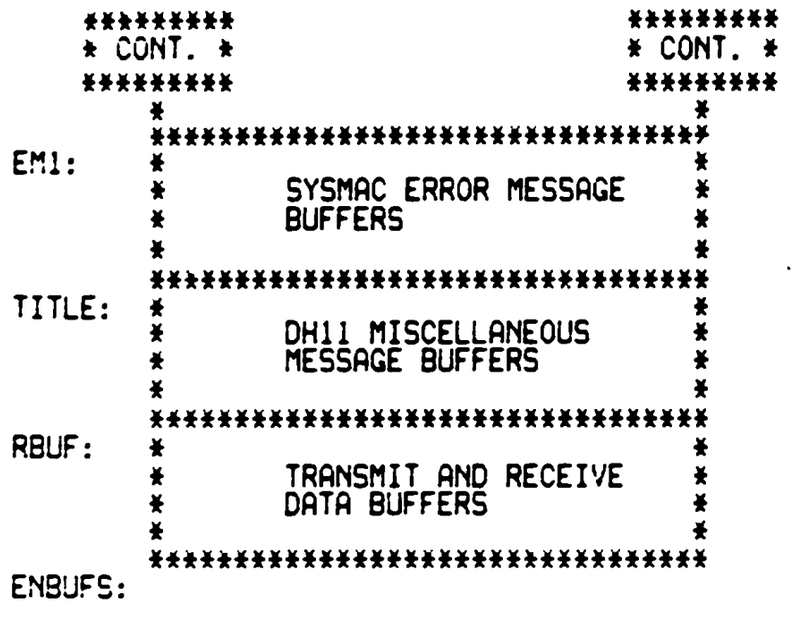
THIS PROGRAM PROVIDES THE MEANS OF TESTING ANY LINE ON ANY DH11 USING AN H315 TEST CONNECTOR TO TERMINATE THE LINE UNDER TEST. THE USER CAN SPECIFY BUFFER SIZE AND LINE PARAMETERS PRIOR TO SELECTING ONE OF THE FOLLOWING DATA PATTERNS FOR TRANSMISSION, RECEPTION, AND ERROR CHECKING:

A. ALTERNATING 1/0 PATTERN
B. BINARY UP COUNT PATTERN
C. BINARY DOWN COUNT PATTERN
D. RANDOM DATA PATTERN
E. CUMULATIVE SEQUENCE OF (A) THRU (D)
F. SINGLE CHARACTER PATTERN
G. TYPED IN BUFFER PATTERN

ALL ERRORS DETECTED ARE REPORTED AS THEY OCCUR AND A SWITCH REGISTER OPTION ALLOWS LOCKING ON A PARTICULAR PATTERN. THIS SUB-PROGRAM WOULD NORMALLY BE SELECTED FOR TROUBLESHOOTING A SPECIFIC PROBLEM.

1.1.4 MD-11-DZDHN CORE MEMORY MAP

```
        ************************************
000000: *                                  *
        *       VECTOR AREA                *
        *                                  *
        ************************************
        *                                  *
        *       STACK AREA                 *
        *                                  *
        ************************************
001100: *                                  *
        *       SYSMAC CONSTANTS           *
        *       AND VARIABLES              *
        *                                  *
        ************************************
BEGIN:  *                                  *
        *       START-UP CODE              *
        *                                  *
        ************************************
STDH1:  *                                  *
        *       DH11 DATA RELIABILITY      *
        *       TESTS                      *
        *                                  *
        ************************************
ECHO:   *                                  *
        *       DH11 SINGLE LINE           *
        *       ECHO TESTS                 *
        *                                  *
        ************************************
EXPAT:  *                                  *
        *       DH11 SINGLE LINE           *
        *       PATTERNS/CABLE TESTS       *
        *                                  *
        ************************************
$EOP:   *                                  *
        *       STANDARD SYSMAC            *
        *       UTILITY ROUTINES           *
        *                                  *
        ************************************
CKRST1: *                                  *
        *       COMMON DH11 UTILITIES      *
        *                                  *
        ************************************
DHADR:  *                                  *
        *       DH11 PROGRAM CONSTANTS     *
        *       AND VARIABLES              *
        *                                  *
        ************************************
        *                                  *
    *********                        *********
    * CONT. *                        * CONT. *
    *********                        *********
```


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

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 WILL LOAD AND RUN IN 8K OF CORE

B. A DH11 16. LINE ASYNCHRONOUS SERIAL LINE MULTIPLEXOR

C. A DH11 TERMINAL DEVICE (LA36,VT50 ETC.) [ECHO TESTS ONLY]

D. TEST CONNECTORS AND MODULE (THE NO. OF EACH REQUIRED IS DETERMINED BY THE PARTICULAR TEST APPLICATION. REFER TO SECTION 6.1 FOR A COMPLETE DISCUSSION OF THE MAINTENANCE CONNECTORS.)

1. H315 TEST CONNECTOR
2. H8611 TEST CONNECTOR
3. M974 TEST MODULE

1.2.2 SOFTWARE REQUIREMENTS

A.	ACT11/ APT11	THE PROGRAM CONTAINS THE NECESSARY "SOFTWARE HOOKS" FOR INTERFACING TO THE ACT11/APT11 MANUFACTURING SYSTEMS. THE PROGRAM CAN BE RUN AS PART OF A QUICK VERIFY "CHAIN" SINCE IT CONTAINS AN AUTOSIZER.
B.	XXDP	THE PROGRAM MAY BE LOADED FROM ANY "XXDP" MEDIA. IF AUTO-STARTED BY THE "XXDP" MONITOR CONTROL WILL BE TRANSFERRED TO THE DATA RELIABILITY PROGRAM.

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

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

A. CPU/CORE MEMORY DIAGNOSTICS
B. MD-11-DZDHM DH11 BASIC DIAGNOSTIC

1.5 FAILURE ASSUMPTIONS

MD-11-DZDHN ASSUMES THAT THE DH11 HARDWARE VERIFIED BY MD-11-DZDHM, THE BASIC DH11 DIAGNOSTIC, 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. (REFER TO 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-DZDHN.
2. BOOT THE SYSTEM TO LOAD THE MONITOR
3. ONCE LOADED THE "XXDP" MONITOR PRINTS AN INTRODUCTORY MESSAGE AND RESPONDS WITH A "."
4. TYPE: "DZDHNB" 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 AN "ALTMODE" WAS TYPED THE MONITOR WILL AUTO START THE PROGRAM AT LOCATION 000200(8) WHICH WILL BEGIN EXECUTION OF THE DATA RELIABILITY PROGRAM.

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

2.1.2 STARTING PROCEDURES

THERE ARE FIVE DIFFERENT STARTING ADDRESSES FOR THIS PROGRAM DEPENDING UPON WHICH SUB-PROGRAM IS TO BE STARTED. THERE ARE THREE FOR THE DATA RELIABILITY PROGRAM AND ONE EACH FOR THE ECHO AND DATA PATTERNS TESTS AS DESCRIBED BELOW:

2.1.2.1 SUB-PROGRAM 1 DH11 DATA RELIABILITY TESTS

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 6.1)
2. SET THE HALT/ENABLE SWITCH TO HALT
3. SET THE SR=000200(8)
4. DEPRESS LOAD ADDRESS
5. SET THE SR=000200 (WORST CASE TESTING)

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

   SET THE SR=000000 (QUICK PASS)

   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.

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

1. INSTALL THE REQUIRED TEST CONNECTORS FOR THE PARTICULAR TEST APPLICATION. (REFER TO SECTION 6.1)
2. SET THE HALT/ENABLE SWITCH TO HALT
3. SET THE SR=000200(8)
4. DEPRESS LOAD ADDRESS
   SET THE SR=000001 (FOR INPUT DIALOGUE)
5. SET THE HALT/ENABLE SWITCH TO ENABLE
6. DEPRESS START
7. THE PROGRAM WILL PRINT THE TITLE AND ASK FOR THE NUMBER OF ADDRESSES BETWEEN VECTORS. TYPE EITHER A "10" OR A "20" TO INDICATE TEN OR TWENTY ADDRESS DISPLACEMENT BETWEEN VECTORS FOLLOWED BY A <CR> (CARRIAGE RETURN).

   NOTE:
   1. SYSTEMS WHERE THE DM11-BB VECTORS ARE INTERLEAVED WITH THE DH11 VECTORS HAVE 20(8) ADDRESSES BETWEEN VECTORS. (THIS IS THE CASE FOR THE 2040 FRONT END)
   2. STANDARD SYSTEMS HAVE THE DH11 VECTORS CONTIGUOUS WITH A 10(8) ADDRESS DISPLACEMENT.

8. THE PROGRAM WILL ASK FOR THE DEVICE ADDRESS NEXT. 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 THE 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

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=0 THE PROGRAM WILL BEGIN EXECUTION TESTING
    THE FIRST SELECTED LINE OF THE FIRST SELECTED DH11.
    (REFER TO PARA 2.4, 3.0, AND 4.0 FOR ERROR AND STATUS
    REPORTS)

13. IF SR8=1 THE PROGRAM WILL HALT AND TYPE THE FOLLOW-
    ING MESSAGE:

        "DEPRESS CONTINUE TO START TESTING"

    WHEN CONTINUE IS DEPRESSED, THE PROGRAM WILL BEGIN TEST-
    ING AS IN STEP 12.

    THE PURPOSE OF THIS HALT IS TO ALLOW DUMPING
    UPDATED VERSIONS OF THE PROGRAM AFTER THE PARAMETERS
    HAVE BEEN SET UP FOR THE PARTICULAR DH11 SYSTEM.

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

1. INSTALL THE REQUIRED TEST CONNECTORS FOR THE
   PARTICULAR TEST APPLICATION. REFER TO SECION 6.1.
2. SET THE HALT/ENABLE SWITCH TO HALT
3. SET THE SR=000204(8)
4. DEPRESS LOAD ADDRESS
5. SET THE SR=000200 (WORST CASE TESTING)

   SET THE SR=000000 (QUICK PASS)

6. SET THE HALT/ENABLE SWITCH TO ENABLE
7. DEPRESS START

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

   DH11 #0 DEVADR=760020, VECTOR=330, 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. CHANGE DEVICE AND LINE SELECT PARAMETERS (START AT LOC 000210(8))

1. INSTALL THE REQUIRED TEST CONNECTORS FOR THE
   SPECIFIC TEST APPLICATION. (REFER TO SECTION 6.1)
2. SET THE HALT/ENABLE SWITCH TO THE HALT POSITION
3. SET THE SR=000210
4. DEPRESS LOAD ADDRESS
5. SET THE SR=000200 (WORST CASE TESTING)

   SET THE SR=000000 (QUICK PASS)

   SET THE SR=000400 (HALT AFTER PARAMETER SETUP)

6. SET THE HALT/ENABLE SWITCH TO ENABLE
7. DEPRESS START
8. PROGRAM WILL ASK FOR DEVICE SELECTION PARAMETER
   PROCEED AS IN (B-10) ABOVE.
9. PROGRAM WILL ASK FOR LINE SELECTION PARAMETERS.
   PROCEED AS IN (B-11) ABOVE.

10. PROGRAM WILL BEGIN EXECUTION AS DESCRIBED IN
PARA 2.1.2.1 (8,12) ABOVE

SEQ 0013

2.1.2.2 SUBPROGRAM 2 DH11 SINGLE LINE ECHO TESTS

1. CONNECT THE TEST TERMINAL TO THE DH11 LINE TO BE TESTED
   AND POWER IT UP.
2. SET THE HALT/ENABLE SWITCH TO HALT
3. SET THE SR=000214(8)
4. DEPRESS LOAD ADDRESS
5. SET THE SR=000001 (TO INHIBIT THE AUTOSIZER)
   NOTE: WHEN THE AUTOSIZER IS NOT INHIBITED
   THAT IS, SR=000000, IT WILL LOAD THE FIRST
   DH CSR ADDRESS AND VECTOR FOUND AND TEST THAT
   DEVICE ONLY. IF ANY OTHER DH'S ARE TO BE TESTED,
   THE CSR ADDRESS AND VECTOR MUST BE INPUT
   MANUALLY, THAT IS, WITH SR=000001.
6. SET THE HALT/ENABLE SW TO ENABLE
7. DEPRESS START
8. THE PROGRAM WILL TYPE THE TITLE MESSAGES
   AND ASK FOR THE NO. OF ADDRESSES BETWEEN VECTORS.
   TYPE EITHER A "10" OR "20" AS DESCRIBED IN PARA 2.1.2.1 (B7) ABOVE.

9. TYPE IN THE DEVICE ADDRESS (IN OCTAL) FOLLOWED
   BY A <CR>

   IF <CR> ONLY IS TYPED THE PROGRAM WILL USE
   A DEFAULT ADDRESS OF 760020(8)

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

10. NEXT THE PROGRAM WILL ASK FOR A VECTOR ADDRESS
11. TYPE IN THE VECTOR ADDRESS FOLLOWED BY A <CR>

    IF <CR> ONLY IS TYPED THE PROGRAM WILL USE
    A DEFAULT VECTOR ADDR OF 330(8)

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

12. NEXT THE PROGRAM WILL ASK FOR THE LINE NO. TO TEST
13. TYPE IN THE LINE NO. (IN OCTAL 00-17) FOLLOWED
    BY A <CR>

    IF A <CR> ONLY IS TYPED THE PROGRAM WILL DEFAULT
    TO LINE #00.

14. NEXT THE PROGRAM WILL ASK YOU IF YOU WANT TO
    CHANGE LINE PARAMETERS.
15. TYPE "Y" FOR YES - "N" OR <CR> FOR NO FOLLOWED BY
    A <CR>.

    IF "NO" THE PROGRAM WILL DEFAULT TO THE LAST
    LINE PARAMETERS TYPED IN OR IF THIS IS THE
    FIRST DIALOGUE IT WILL DEFAULT TO 9600 BAUD,
    8 BIT CHARS, 1 STOP BIT, AND ODD PARITY.

16. IF YOU TYPED "Y" IN (15) DO STEPS (17) THRU (21)
    OTHERWISE GO TO STEP (22)
17. WHEN THE PROGRAM ASKS FOR TRANSMITTER SPEED

TYPE IN ONE OF THE 13. LEGAL SPEEDS IN DECIMAL
FOLLOWED BY A <CR>.

18. WHEN THE PROGRAM ASKS FOR RECEIVER SPEED TYPE
IN ONE OF THE 13. LEGAL SPEEDS IN DECIMAL
FOLLOWED BY A <CR>.

NOTE: FOR (17) AND (18) IF THE SPEED DESIRED
IS 134.5, TYPE IT WITHOUT THE DECIMAL
POINT.

REFER TO PARA 5.2.3 FOR DESCRIPTION OF SPEED
TABLES.

19. WHEN THE PROGRAM ASKS FOR CHAR LENGTH, TYPE
IN THE NO. DESIRED FOLLOWED BY A <CR>
20. WHEN THE PROGRAM ASKS FOR THE NO. OF STOP BITS
TYPE IN THE NO. DESIRED FOLLOWED BY A <CR>
21. WHEN THE PROGRAM ASKS FOR PARITY, TYPE IN:

O	FOR ODD
E	FOR EVEN
<CR>	FOR NONE

22. THE PROGRAM WILL NEXT ASK FOR THE FILLER CHARACTER.
TYPE IN THE FILLER CHAR FOLLOWED BY A <CR>

IF A <CR> ONLY IS TYPED THE PROGRAM WILL USE
A "NULL" FILLER WHICH IS THE NORMAL CASE.

23. THE PROGRAM WILL NEXT ASK FOR THE FILLER COUNT.
TYPE IN THE COUNT IN OCTAL FOLLOWED BY A <CR>.

IF A <CR> ONLY IS TYPED THE PROGRAM WILL DEFAULT
TO ONE FILLER. IF A NO. GREATER THAN 4 BITS
IS TYPED THE PROGRAM WILL TRUNCATE IT TO 4 BITS.
THE MAXIMUM COUNT ALLOWED IS 15(10).

24. NEXT THE PROGRAM WILL ASK YOU IF YOU WANT SEND
MODE. TYPE A "Y" IF YES - "N" OR "<CR>" IF NO.
25. IF YOU TYPED "Y" IN RESPONSE TO (24) THE PROGRAM
WILL ASK YOU TO TYPE IN THE SEND BUFFER ON THE
CONSOLE TTY. WHEN YOU WANT TO SEND THIS BUFFER TO
THE TEST TERMINAL ON THE DH11 TYPE A "CONTROL-C".

NOTE: ALWAYS START THE BUFFER WITH A
<CR><LF> TO MAKE IT EASIER TO
INTERPRET THE DISPLAY ON THE DH11
TERMINAL WHEN THE BUFFER IS SENT.

26. AFTER THE TEST BUFFER IS SENT THE PROGRAM WILL
ASK FOR LINE # AGAIN AND YOU REPEAT THE SEQUENCE
STARTING WITH STEP (12) ABOVE.
27. IF YOU TYPED SOME CHAR OTHER THAN "Y" IN RESPONSE
TO (24) THE PROGRAM WILL ASSUME "ECHO" MODE AND
ASK YOU TO TYPE IN ON THE TEST TERMINAL CONNECTED
TO THE LINE UNDER TEST.
28. NOW GO TO THE TEST TERMINAL AND BEGIN TYPING.
(IF THIS IS A REMOTE TERMINAL, ESTABLISH
APPROPRIATE MODEM CONNECTION.)
EACH CHAR TYPED SHOULD BE ECHOED ON THE DH11 TEST

TERMINAL. IF YOU WANT TO ECHO AN ENTIRE BUFFER
TYPE "CONTROL-E" AND THE PROGRAM WILL ECHO THE
ENTIRE BUFFER TYPED IN ON THE TERMINAL TO THAT
POINT.

29. TO CHANGE LINE # AND PARAMETERS - TYPE "CONTROL-C"
ON THE DH11 TEST TERMINAL AND RETURN TO THE CONSOLE
TERMINAL
30. TO TEST A DIFFERENT DH11 UNIT, THE PROGRAM MUST
BE RESTARTED AT 000214(8).

2.1.2.3 SUBPROGRAM 3 DH11 SINGLE LINE DATA PATTERNS/CABLE TESTS

1. TERMINATE THE LINE TO BE TESTED WITH AN H315 TEST CONNECTOR.
2. SET THE HALT/ENABLE SWITCH TO HALT
3. SET THE SR=000220(8)
4. DEPRESS LOAD ADDRESS
5. SET THE SR=000001 (TO INHIBIT THE AUTOSIZER)
   NOTE: WHEN THE AUTOSIZER IS NOT INHIBITED, THAT IS SR=000000, IT WILL LOAD THE FIRST DH CSR ADDRESS AND VECTOR FOUND AND TEST THAT DEVICE ONLY. IF ANY OTHER DH'S ARE TO BE TESTED, THE CSR ADDRESS AND VECTOR MUST BE INPUT MANUALLY, THAT IS SR=000001.
6. SET THE HALT/ENABLE SW TO ENABLE
7. DEPRESS START
8. THE PROGRAM WILL TYPE THE TITLE MESSAGES AND ASK FOR THE NO. OF ADDRESSES BETWEEN VECTORS. TYPE EITHER A "10" OR "20" AS DESCRIBED IN PARA 2.1.2.1 (B 7).
9. NEXT THE PROGRAM WILL ASK FOR THE DH11 DEVICE ADDRESS

   TYPE IN THE DEVICE ADDRESS (IN OCTAL) FOLLOWED BY A <CR>

   IF <CR> ONLY IS TYPED THE PROGRAM WILL USE A DEFAULT ADDRESS OF 760020(8)

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

10. NEXT THE PROGRAM WILL ASK FOR A VECTOR ADDRESS
11. TYPE IN THE VECTOR ADDRESS FOLLOWED BY A <CR>

    IF <CR> ONLY IS TYPED THE PROGRAM WILL USE A DEFAULT VECTOR ADDR OF 330(8)

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

12. NEXT THE PROGRAM WILL ASK FOR THE LINE NO. TO TEST
13. TYPE IN THE LINE NO. (IN OCTAL 00-17) FOLLOWED BY A <CR>

    IF A <CR> ONLY IS TYPED THE PROGRAM WILL DEFAULT TO LINE #00.

14. NEXT THE PROGRAM WILL ASK YOU IF YOU WANT TO CHANGE LINE PARAMETERS.
15. TYPE "Y" IF YOU DO - "N" OR <CR> IF YOU DON'T

    IF "NO" THE PROGRAM WILL DEFAULT TO THE LAST LINE PARAMETERS TYPED IN OR IF THIS IS THE FIRST DIALOGUE IT WILL DEFAULT TO 9600 BAUD, 8 BIT CHARS, 1 STOP BIT, AND ODD PARITY.

16. IF YOU TYPED "Y" IN (15) DO STEPS (17) THRU (21) OTHERWISE GO TO STEP (22)

17. WHEN THE PROGRAM ASKS FOR TRANSMITTER SPEED TYPE IN ONE OF THE 13. LEGAL SPEEDS IN DECIMAL FOLLOWED BY A <CR>.
18. WHEN THE PROGRAM ASKS FOR RECEIVER SPEED TYPE IN ONE OF THE 13. LEGAL SPEEDS IN DECIMAL FOLLOWED BY A <CR>.

    NOTE: FOR (17) AND (18) IF THE SPEED DESIRED IS 134.5, TYPE IT WITHOUT THE DECIMAL POINT.

19. WHEN THE PROGRAM ASKS FOR CHAR LENGTH, TYPE IN THE NO. DESIRED FOLLOWED BY A <CR>
20. WHEN THE PROGRAM ASKS FOR THE NO. OF STOP BITS TYPE IN THE NO. DESIRED FOLLOWED BY A <CR>
21. WHEN THE PROGRAM ASKS FOR PARITY, TYPE IN:

O	FOR ODD
E	FOR EVEN
<CR>	FOR NONE

    FOLLOWED BY A <CR>
22. THE PROGRAM WILL NEXT ASK FOR THE FILLER CHARACTER. TYPE IN THE FILLER CHAR FOLLOWED BY A <CR>

    IF A <CR> ONLY IS TYPED THE PROGRAM WILL USE A "NULL" FILLER WHICH IS THE NORMAL CASE.

23. THE PROGRAM WILL NEXT ASK FOR THE FILLER COUNT. TYPE IN THE COUNT IN OCTAL FOLLOWED BY A <CR>.

    IF A <CR> ONLY IS TYPED THE PROGRAM WILL DEFAULT TO ONE FILLER. IF A NO. GREATER THAN 4 BITS IS TYPED THE PROGRAM WILL TRUNCATE IT TO 4 BITS. THE MAXIMUM COUNT ALLOWED IS 15.

24. NEXT THE PROGRAM WILL ASK YOU THE BUFFER SIZE. TYPE IN A DECIMAL NO. BETWEEN 1 TO 512. FOLLOWED BY A <CR>.

    IF A <CR> ONLY IS TYPED THE PROGRAM WILL DEFAULT TO A BUFFER SIZE OF 256. BYTES.

    IF THE NO. TYPED IS TOO LARGE AN ERROR MESSAGE IS TYPED AND YOU ARE ASKED TO TRY AGAIN.

25. NEXT THE PROGRAM WILL ASK FOR THE TYPE OF PATTERN AND TELL YOU TO SET SR07=1 IF YOU WANT TO LOCK ON THE SELECTED PATTERN.
26. TYPE (A,U,D,R,B,S, OR <CR>) TO SELECT THE PATTERN AS DESCRIBED BELOW:

A	ALTERNATING 1/0
U	BINARY UP COUNT
D	BINARY DOWN COUNT
R	RANDOM DATA
B	TYPED IN BUFFER
S	SINGLE CHARACTER
<CR>	SEQUENCE OF A,U,D, AND R

27. IF YOU TYPED A,U,D,R, OR <CR>, THE PROGRAM WILL TRANSMIT, RECEIVE, AND DATA CHECK THE SELECTED PATTERN.

SR07=1 IT WILL LOCK ON THIS PATTERN

SR07=0 IT WILL RETURN TO STEP (24) AFTER COMPLETING THE TEST OF THIS PATTERN AND ASK FOR A NEW PATTERN.

28. IF YOU TYPED A "B" IN (26) THE PROGRAM WILL ASK YOU TO TYPE IN A TEST PATTERN AND TERMINATE IT WITH A "CONTROL-C". WHEN THE PROGRAM SENSES THE TERMINATOR IT WILL BEGIN EXECUTION AS IN (27) USING THE THE TYPED IN BUFFER AS THE PATTERN.
29. IF YOU TYPED AN "S" IN RESPONSE TO (26) THE PROGRAM WILL ASK FOR A SINGLE CHAR. TYPE A SINGLE CHAR FOLLOWED BY A <CR>. THE PROGRAM WILL FILL THE BUFFER WITH THE TYPED IN CHAR AND BEGIN EXECUTION AS IN (27) USING THE BUFFER FULL OF THE TEST CHAR AS A PATTERN.
30. TO CHANGE DH11'S, LINE PARAMETERS ETC. YOU MUST RESTART THE TESTS AT LOC. 000220(8).

2.1.3 RESTART PROCEDURES

SAME AS THE STARTING PROCEDURES

2.2 SPECIAL ENVIRONMENTS

SEQ 0020

2.2.1 ACT11/APT11 — THE PROGRAM MAY BE LOADED BY THE ACT11/APT11 SYSTEMS, AND MAY BE RUN AS PART OF A QUICK VERIFY CHAIN SINCE THE PROGRAM CONTAINS AN AUTOSIZER.

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

A. SUB-PROGRAM 1		DH11 DATA RELIABILITY TESTS
	SR15=1	HALT ON ERROR
	SR14=1	LOOP ON CURRENTLY SELECTED DH11
	SR13=1	INHIBIT ERROR, PROGRESS, AND PERFORMANCE PRINTOUTS
	SR8=1	HALTS AFTER CONFIGURATION TO PERMIT DUMPING PRECONFIGURED COPIES OF THE PROGRAM.
	SR7=1	PERFORMS A STANDARD PASS(NOT QUICK VERIFY.)
	SR7=0	QUICK VERIFY - DO COMPLETE TESTING ON EACH LINE AT 9600. BAUD ONLY
	SR1=1	TYPES DEVICE MAP GENERATED BY THE AUTOSIZER.
	SR0=1	ALLOWS THE USER TO INPUT DH PARAMETERS MANUALLY. (INHIBITS THE AUTOSIZER.)
B. SUB-PROGRAM 2		DH11 SINGLE LINE ECHO TESTS

(NONE)

C. SUB-PROGRAM 3 DH11 SINGLE LINE PATTERNS/CABLE TESTS

SR15=1 HALT ON ERROR
SR13=1 INHIBIT ERROR AND STATUS PRINTOUTS
SR07=1 LOOP ON CURRENT TEST PATTERN

2.3.2 CORE MEMORY LOCATIONS

A. SUB-PROGRAM 1 DH11 DATA RELIABILITY TESTS

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) DEVICE ADDRESS TABLE

THERE IS A 16. WORD TABLE STARTING AT THE ADDRESS TAGGED "DHADTB:" THAT IS PROGRAM LOADED TO SPECIFY 16. CONTIGUOUS DH11'S STARTING AT THE BUS ADDRESS 160020(8). THIS TABLE IS MODIFIED AT CONFIGURATION TIME IF THE USER TYPES IN A DIFFERENT STARTING ADDRESS, OR IT MAY BE PATCHED TO REFLECT ANY UNIQUE DH11 SYSTEM CONFIGURATION.

2) VECTOR ADDRESS TABLE

THERE IS A 16. WORD TABLE STARTING AT THE ADDRESS TAGGED "DHVCTB:" THAT IS PROGRAM LOADED TO SPECIFY 16. CONTIGUOUS VECTORS STARTING WITH 330(8) AND EACH ENTRY DISPLACED BY 8. WORDS (330,350,370, ETC.) THIS TABLE IS MODIFIED AT CONFIGURATION TIME IF THE USER TYPES A DIFFERENT STARTING VECTOR ADDRESS, OR IT MAY BE PATCHED TO REFLECT ANY UNIQUE DH11 SYSTEM CONFIGURATION.

3) BR LEVEL TABLE

THERE IS A 16. WORD TABLE STARTING AT THE ADDRESS TAGGED "BRLVL:" THAT IS PROGRAM LOADED TO CONTAIN A 120240(8) IN EACH ENTRY WHICH SPECIFIES BR LEVEL 5 FOR BOTH XMIT AND RECEIVE FOR ALL 16. DH11'S. IT IS NOT CHANGED AT CONFIGURATION TIME BUT MAY BE PATCHED TO REFLECT ANY UNIQUE DH11 SYSTEM CONFIGURATION.

4) DEVICE SELECTION PARAMETER

THERE IS A WORD TAGGED "DHSEL:" THAT IS PROGRAM LOADED TO CONTAIN A 000001(8) WHICH SELECTS ONLY ONE DH11 (DH11 #00). THIS LOCATION CAN BE MODIFIED AT CONFIGURATION TIME TO SPECIFY ANY COMBINATION OF DH11'S UP TO A MAXIMUM OF 16. UNITS. REFER TO PARA 2.1.3.1 (B 10) FOR A DESCRIPTION OF ITS ENCODING.

5) LINE SELECTION PARAMETER (PARA 2.1.2.1 (B11))

THERE IS A WORD TAGGED "LINSEL:" THAT IS PROGRAM LOADED AS 177777(8) TO SPECIFY ALL 16. LINES ARE TO BE TESTED IN EACH SELECTED DH11. IT MAY BE MODIFIED AT CONFIGURATION TIME TO SPECIFY ANY COMBINATION OF LINES TO TEST. REFER TO SECTION 2.1.2 (B 11) FOR A DESCRIPTION OF ITS ENCODING.

NOTE: THE DATA RELIABILITY PROGRAM IS TABLE DRIVEN IN THAT IT USES "DHSEL:" "LINSEL:" AND THE CONTENTS OF THE THREE 16. WORD TABLES TO DEFINE THE DH11 CONFIGURATION TO BE TESTED.

B. SUB-PROGRAM 2 DH11 SINGLE LINE ECHO TESTS

(NONE)

C. SUB-PROGRAM 3 DH11 SINGLE LINE PATTERNS/CABLE TESTS

1)PATLIM: 10.

THERE IS A LOCATION TAGGED "PATLIM:" THAT SPECIFIES THE NO. OF TEST PATTERN ITERATIONS TO EXECUTE IN THE PATTERNS TESTS. IT IS PROGRAM LOADED TO SPECIFY TEN ITERATIONS BEFORE THE "TEST DONE" REPORT IS TYPED.

2) DATCNT:

THERE IS A LOCATION TAGGED "DATCNT:" THAT KEEPS A COUNT OF THE NO. OF ITERATIONS COMPLETED DURING THE PATTERNS TESTS. THIS INFORMATION GETS TYPED OUT AS PART OF THE ERROR MESSAGE IF A DATA ERROR OCCURS IN THE PATTERNS TEST UNDER THE HEADING "ICOUNT".

2.4 EXECUTION TIMES

A. SUB-PROGRAM 1 DH11 DATA RELIABILITY TESTS

1. SR07=0 QUICK TEST

APPROXIMATELY 15. SECONDS FOR EACH LINE WITH 1824 CHARS BEING TRANSMITTED AND RECEIVED.

2. SR7=1 COMPLETE TESTING

APPROXIMATELY 15. MINUTES FOR EACH LINE WITH 18,720 CHARS BEING TRANSMITTED AND RECEIVED ON EACH LINE SELECTED FOR TEST.

B. SUB-PROGRAM 2 DH11 SINGLE LINE ECHO TESTS

NOT APPLICABLE SINCE THESE TESTS INVOLVE THE USER MANUALLY TYPING IN ON THE TERMINAL.

C. SUB-PROGRAM 3 DH11 SINGLE LINE PATTERNS/CABLE TESTS

EXECUTION TIMES VARY FROM LESS THAN 5 SECONDS TO GREATER THAN 15. MINUTES DEPENDING UPON BUFFER SIZE, LINE PARAMETERS, AND PATTERN SELECTED.

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 NO.S)

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 ERROP MESSAGES WITHIN MD-11-DZDHN 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 WHETHER 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.

3.1.2 ERROR MESSAGE TABLE


```
;ERROR TABLE ITEM FOR ERROR 1

        EM1             ;"NON EX MEMORY ERROR - DROPPED LINE #  "
        DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
        DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 2

        EM2             ;"TRANSMITTER FALSE INTERRUPT - DROPPED LINE#  "
        DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
        DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 3

        EM3             ;"BUFFER ACTIVE REGISTER ERROR - DROPPED LINE #  "
        DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
        DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 4

        EM4             ;"BYTE COUNT REGISTER ERROR - DROPPED LINE #  "
        DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
        DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 5

        EM5             ;"CURRENT ADDRESS REGISTER ERROR - DROPPED LINE #  "
        DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
        DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 6

        EM6             ;"SILO OVERFLOW ERROR - DROPPED LINE #  "
        DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
        DT1             ;"$ERRPC,CURLPR,$PEG1,$REG2,$REG3,$REG4
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 7

        EM7             ;"RECEIVER FALSE INTERRUPT - LINE #  "
        DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
        DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 10

        EM10            ;"INVALID DATA IN SILO - DROPPED LINE #  "
        DH2             ; (PC)   CURLPR  CHAR #  WASADR  SHBADR   WAS     S/B"
        DT2             ;$ERRPC,CURLPR,$REG0,$REG1,$REG2,$REG3,$REG4
        DF2             ;PRINT ALL OCTAL
```

```
;ERROR TABLE ITEM FOR ERROR 11

        EM11            ;"DATA ERROR - LINE #  "
        DH2             ; (PC)   CURLPR  CHAR #  WASADR  SHBADR   WAS     S/B"
        DT2             ;$ERRPC,CURLPR,$REG0,$REG1,$REG2,$REG3,$REG4
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 12

        EM12            ;"TEST TIMEOUT - DROPPED LINE #  "
        DH3             ;" (PC)   CURLPR  RTOTAL  XTOTAL  RDONE"
        DT3             ;"$ERRPC,CURLPR,$TMP0,$TMP1RDONE"
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 13

        NOTE:   ERROR 13 IS CALLED TO PRINT EACH LINE OF DATA IN THE
                ERROR STATISTICS TABLE. IT PRINTS ONLY DATA WITHOUT ANY
                MESSAGE OR DATA HEADERS.


        0               ;NO MESSAGE
        0               ;NO DATA HEADER
        DT4             ;$TMP0,$TMP1,$TMP2,$TMP3,$TMP4,$TMP5,$TMP6
        DF1             ;PRINT ALL DECIMAL

;ERROR TABLE ITEM FOR ERROR 14

        EM14            ;"BUS ERROR TRAP TO 04"
        DH4             ;" (PC)    (PS)    (SP)   TRAPPC  TRAPPS"
        DT5             ;$ERRPC,$TMP0,$REG6,$REG1,$REG2"
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 15

        EM15            ;"RSVD INSTR TRAP TO 10"
        DH4             ;" (PC)    (PS)    (SP)   TRAPPC  TRAPPS"
        DT5             ;$ERRPC,$TMP0,$REG6,$REG1,$REG2"
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 16

        EM16            ;"SINGLE LINE ECHO TEST - INTR WAIT TIMEOUT"
        DH5             ;" (PC)   DEVADR   LINE    (SCR)  CURLPR  EXFLAG"
        DT6             ;$ERRPC,$REG1,LINE,$TMP0,CURLPR,EXFLAG"
        DF2             ;PRINT ALL OCTAL
```

```
        NOTE:   ERRORS 17 THRU 24 ARE USED TO REPORT PERFORMANCE
                NOT ERRORS.

;ERROR TABLE ITEM FOR ERROR 17

        EM17            ;"ALTERNATING 1/0 PATTERN TEST DONE"
        DH6             ;"  (PC)    DEVADR    LINE    CURLPR   ICOUNT"
        DT7             ;$ERRPC,DHADR,LINE,CURLPR,$REGO
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 20

        EM20            ;"BINARY UP COUNT PATTERN TEST DONE"
        DH6             ;"  (PC)    DEVADR    LINE    CURLPR   ICOUNT"
        DT7             ;$ERRPC,DHADR,LINE,CURLPR,$REGO
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 21

        EM21            ;"BINARY DOWN COUNT PATTERN TEST DONE"
        DH6             ;"  (PC)    DEVADR    LINE    CURLPR   ICOUNT"
        DT7             ;$ERRPC,DHADR,LINE,CURLPR,$REGO
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 22

        EM22            ;"RANDOM DATA PATTERN TEST DONE"
        DH6             ;"  (PC)    DEVADR    LINE    CURLPR   ICOUNT"
        DT7             ;$ERRPC,DHADR,LINE,CURLPR,$REGO
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 23

        EM23            ;SINGLE CHAR PATTERN TEST DONE"
        DH6             ;"  (PC)    DEVADR    LINE    CURLPR   ICOUNT"
        DT7             ;$ERRPC,DHADR,LINE,CURLPR,$REGO
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 24

        EM24            ;"TYPED BUFFER PATTERN TEST DONE"
        DH6             ;"  (PC)    DEVADR    LINE    CURLPR   ICOUNT"
        DT7             ;$ERRORPC,DHADR,LINE,CURLPR,$REGO
        DF2             ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 25

        EM25            ;"DATA PATTERNS TEST TIMEOUT"
        DH7             ;"  (PC) DEVADR    LINE    CURLPR   ICOUNT   PATCDE"
        DT10            ;$ERRPC,DHADR,LINE,CURLPR,$REGO,$REG1
        DF2             ;PRINT ALL OCTAL
```

3.1.3 DATA HEADER MNEUMONIC DEFINITIONS

ALL NUMBERS PRINTED AS ERROR DATA ARE IN OCTAL

(PC) ADDRESS OF THE ERROR CALL (ERROR PC)

(PS) CONTENTS OF THE PSW AT THE TIME OF THE ERROR

(SP) CONTENTS OF THE STACK POINTER AT THE TIME OF THE ERROR

LINE INDICATES THE LINE NUMBER THAT FAILED

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

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.

WASADR CORE MEMORY ADDRESS OF THE "WAS" DATA (ACTUAL DATA READ)

SBADR CORE MEMORY ADDRESS OF THE S/B DATA (GOOD DATA)

CHAR # INDICATES THE CHARACTER POSITION IN THE DATA BUFFER

ICOUNT INDICATES ITERATION COUNT OF DATA PATTERNS TESTS - PROGRAM DEFAULTS TO ITERATING EACH PATTERN 10. TIMES.

PATCDE INDICATES PATTERN BEING TESTED WHEN ERROR OCCURRED AS SHOWN BELOW:

PATCDE=	101	ALTERNATING 1/0 PATTERN
	125	BINARY UP COUNT PATTERN
	104	BINARY DOWN COUNT
	122	RANDOM DATA PATTERN
	123	SINGLE CHAR PATTERN
	102	TYPED IN BUFFER PATTERN

(SCR) INDICATES CONTENTS OF THE "SCR" REG WHEN ERROR OCCURRED

EXFLAG INDICATES STATE OF THE XMITTER INTR SERVICE ROUTINE IN THE ECHO TESTS AS SHOWN BELOW:

EXFLAG=	1	CONTROL-C WAS TYPED
	2	CONTROL-E WAS TYPED
	3	BUFFER WAS BEING DUMPED

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

A. SUB-PROGRAM 1 DH11 DATA RELIABILITY TESTS

AT THE COMPLETION OF EACH PASS THROUGH EACH DH11 UNDER TEST, OR WHEN INVOKED BY THE USER TYPING AN "S" ON THE CONSOLE TERMINAL, THE STATISTICS TABLE SHOWN BELOW IS TYPED ON THE CONSOLE DEVICE. THE TABLE CONSISTS OF "NN" ENTRIES WHERE "NN" IS THE NUMBER OF LINES SELECTED FOR TEST.

DH #NN STATISTICS:

LINE #	RTOTAL	XTOTAL	DATERR	PARERR	FRMERR	OVRERR
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO
LLLLLL	RRRRRR	XXXXXX	DDDDDD	PPPPPP	FFFFFF	OOOOOO

```
WHERE:  N       IS THE DH11 # IN OCTAL
        L       IS THE LINE # IN OCTAL
        R       IS THE TOTAL NO. OF CHARS RECEIVED IN DECIMAL
        X       IS THE TOTAL NO. OF CHARS TRANSMITTED IN DECIMAL
        D       IS THE TOTAL NO. OF DATA COMPARE ERRORS DETECTED IN DECIMAL
        P       IS THE TOTAL NO. OF PARITY ERRORS IN DECIMAL
        F       IS THE TOTAL NO. OF FRAMING ERRORS IN DECIMAL
        O       IS THE TOTAL NO. OF OVERRUN ERRORS IN DECIMAL
```

NOTES: 1.) IF A LINE WAS DROPPED DURING THE TEST DUE TO AN UNRECOVERABLE READ OR WRITE ERROR THE MESSAGE SHOWN BELOW WILL REPLACE THE NORMAL ERROR STATISTICS ENTRY:

"LINE #NN WAS DROPPED"

WHERE "NN" IS THE LINE NO. IN OCTAL.

2.) IF THE PRINTOUT IS INVOKED BY TYPING AN "S" , THE "RTOTAL" AND "XTOTAL" ENTRIES MAY OR MAY NOT BE EQUAL DEPENDING UPON WHEN THE PROGRAM "SAW" THE "S".

3.) AFTER PRINTING THE ERROR STATISTICS TABLE, THE PROGRAM WILL RESTART AND BEGIN TESTING THE NEXT DH11 IN SEQUENCE. IF ONLY ONE DH11 IS SELECTED FOR TEST OR SR14=1, THE SAME DH11 WILL BE TESTED AGAIN.

B. SUB-PROGRAM 2 DH11 SINGLE LINE ECHO TESTS

1) SEND MODE: THE DISPLAY ON THE DH11 TEST TERMINAL SHOULD MATCH THE BUFFER TYPED IN ON THE CONSOLE TERMINAL.

2) ECHO MODE: THE CHARACTERS ECHOED ON THE DH11 TEST TERMINAL SHOULD MATCH THE CHARACTERS TYPED ON THE TEST TERMINAL KEYBOARD.

C. SUB-PROGRAM 3 DH11 SINGLE LINE PATTERNS/CABLE TESTS

(NONE PROVIDED)

4.2 PROGRESS REPORTS

A. SUB-PROGRAM 1 DH11 DATA RELIABILITY TESTS

1. EACH TIME A NEW DH11 IS SELECTED FOR TEST THE PROGRAM TYPES:

   "TESTING DH11 #NN"

   WHERE "NN" IS THE NO. IN OCTAL OF THE DH11 CURRENTLY BEING TESTED. (00 - 17)

2. EACH TIME A NEW LINE IS SELECTED FOR TEST THE PROGRAM TYPES:

   "TESTING LINE #NN"

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

3. AFTER COMPLETE TESTING OF ALL SELECTED DH11'S THE FOLLOWING MESSAGE IS PRINTED:

   "END PASS #NNNNN"

   WHERE: N IS THE NO. OF COMPLETE PROGRAM PASSES DURING THE CURRENT "RUN"

B. SUB-PROGRAM 2 DH11 SINGLE LINE ECHO TESTS

(NONE SUPPLIED)

C. SUB-PROGRAM 3 DH11 SINGLE LINE PATTERNS/CABLE TESTS

EACH TIME A SPECIFIC TEST PATTERN TEST IS COMPLETED (10. ITERATIONS) THE FOLLOWING MESSAGE IS TYPED:

```
"NAME" PATTERN TEST DONE
 (PC)   DEVADR   LINE   CURLPR  ICOUNT
PPPPPP  DDDDDD  LLLLLL  CCCCCC  IIIIII
```

WHERE:	NAME	IS THE NAME OF THE PATTERN - IE "RANDOM", "BINARY UP COUNT", ETC
	P	IS THE PC OF THE MESSAGE CALL
	D	IS THE ADDRESS OF THE DH11 UNDER TEST
	L	IS THE LINE NO. BEING TESTED
	C	IS THE CONTENTS OF THE "LPR" DURING THE TEST
	I	IS THE NO. OF TEST PATTERN ITERATIONS COMPLETED

THIS TYPEOUT MAY BE INHIBITED BY SETTING SR13=1.

5.0 DH11 DEVICE INFORMATION

SEQ 0034

5.1 ADDRESS AND VECTOR ASSIGNMENTS

THE DH11 USES FLOATING ADDRESSES AND IS LOCATED AFTER DJ11'S IN THE FLOATING ADDRESS SPACE THAT BEGINS AT LOCATION 760 010. BECAUSE THE DH11 HAS EIGHT REGISTERS, IT MUST BE ASSIGNED AN ADDRESS THAT IS A MULTIPLE OF 20 (OCTAL). ALL DH11'S IN A 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 STARTS AT ADDRESS 300. THE VECTORS STARTING AT 300 ARE USED IN THE FOLLOWING ORDER: DC11; KL11/DL11-A, B; DP11; DM11-A; DN11; DM11-BB; DR11-A; DR11-C; PA611 READERS; PA611 PUNCHES; DT11; DX11; DL11-C, D, E; DH11.

THE RECEIVER VECTOR IS THE LOWER NUMBERED VECTOR. THE PRIORITY OF THE RECEIVER AND TRANSMITTER INTERRUPTS ARE INDIVIDUALLY SELECTABLE BY MEANS OF TWO STANDARD PDP11 PRIORITY JUMPER PLUGS. BR LEVEL 5 IS STANDARD.

5.2 REGISTER DEFINITION

THE FOLLOWING SECTION DESCRIBES THE BIT ASSIGNMENTS WITHIN EACH REGISTER: BITS MARKED UNUSED AND WRITE ONLY ARE ALWAYS READ AS ZERO. ATTEMPTING TO WRITE INTO UNUSED OR READ ONLY BITS HAS NO EFFECT ON THOSE BITS. INIT REFERS TO THE INITIALIZE SIGNAL GENERATED BY THE PROCESSOR (E.G. UPON EXECUTION OF A RESET INSTRUCTION). TRANSMIT AND RECEIVE ARE WITH RESPECT TO THE DH11.

5.2.1 THE SYSTEM CONTROL REGISTER - ADDRESS X00

THE SYSTEM CONTROL REGISTER IS A BYTE-ADDRESSABLE REGISTER. THE BIT ASSIGNMENT IS AS FOLLOWS:

BITS	DESCRIPTION

00-03 LINE SELECTION

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

04, 05 MEMORY EXTENSION

THE INFORMATION STORED IN THESE BITS BECOMES BITS 16 AND 17 RESPECTIVELY OF ANY CURRENT ADDRESS LOADED BY THE PROGRAM INTO THE CURRENT ADDRESS REGISTER. THESE BITS ARE READ/WRITE BUT, WHEN READ, REPRESENT ONLY THE STATUS OF BITS 4 AND 5 OF THE SYSTEM CONTROL REGISTER, NOT THE STATUS OF ADDRESS BITS 16 AND 17 OF THE SELECTED LINE. SEE THE SILO STATUS REGISTER FOR FURTHER INFORMATION. THIS ARRANGEMENT PERMITS INTERRUPT SERVICE ROUTINES TO 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 THE SILO EXCEEDS THE "ALARM LEVEL" SPECIFIED BY THE LOW BYTE OF THE SILO STATUS REGISTER. THIS BIT IS READ ONLY, EXCEPT IN MAINTENANCE MODE, WHERE IT IS READ/WRITE. SETTING OF THIS BIT WILL GENERATE AN INTERRUPT REQUEST IF BIT 6 (ABOVE) IS ALSO SET.

08 CLEAR NON-EXISTENT MEMORY INTERRUPT

THIS BIT, WHEN SET, CLEARS THE NON-EXISTENT MEMORY INTERRUPT FLIP-FLOP (BIT 10) AND CLEARS ITSELF. THIS BIT 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 MEMORY LOCATION ON THE UNIBUS AND NO SLAVE SYNC IS RECEIVED IN 20 US. THIS INDICATES THAT THE ADDRESSED LOCATION OR DEVICE DOES NOT EXIST. THIS BIT CAUSES AN INTERRUPT REQUEST IF SET WHILE TRANSMITTER AND NON-EXISTENT MEMORY INTERRUPT ENABLE IS SET. THIS BIT IS READ ONLY, EXCEPT IN MAINTENANCE MODE, WHERE IT IS READ/WRITE.

11 MASTER CLEAR

THIS BIT, WHEN SET, GENERATES "INITIALIZE" WITHIN THE DH11, CLEARING THE SILO, THE UARTS, AND THE REGISTERS. THE EXACT BITS CLEARED ARE DISCUSSED IN THE SECTION ON INITIALIZATION. READ/WRITE.

12 STORAGE INTERRUPT ENABLE

THIS BIT, WHEN SET, PERMITS THE SETTING OF BIT 14 TO GENERATE AN INTERRUPT REQUEST. THIS BIT IS READ/WRITE.

13 TRANSMITTER AND NON-EX-MEM INTERRUPT ENABLE

SEQ 0036

THIS BIT, WHEN SET, PERMITS THE SETTING OF BIT 10 OR 15 TO GENERATE AN INTERRUPT REQUEST. THIS BIT IS READ/WRITE.

14 STORAGE INTERRUPT

THIS BIT IS SET WHEN THE RECEIVER SCANNER FINDS A RECEIVER HOLDING BUFFER WITH A CHARACTER IN IT, TRIES TO STORE THAT CHARACTER IN THE SILO, AND CANNOT DO SO BECAUSE OF A LACK OF SPACE. WHEN SET THIS BIT WILL CAUSE AN INTERRUPT REQUEST IF BIT 12 IS SET. THIS BIT IS READ ONLY, EXCEPT IN MAINTENANCE MODE, WHERE IT IS READ/WRITE.

15 TRANSMITTER INTERRUPT

THIS BIT IS SET WHEN THE DH11 CONCLUDES AN NPR CYCLE THAT INCREMENTED A BYTE COUNT TO ZERO, INDICATING THE LAST CHARACTER IN A MESSAGE BUFFER WAS LOADED INTO A UART TRANSMITTER HOLDING REGISTER. THIS BIT WILL CAUSE AN INTERRUPT REQUEST IF BIT 13 IS SET. THIS BIT IS READ/WRITE. (IT IS SET DURING AN NPR CYCLE.)

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 LEAST SIGNIFICANT BITS IS BIT 00.
08-11	LINE NUMBER THESE BITS INDICATE THE LINE NUMBER ON WHICH THE NEXT RECEIVED CHARACTER WAS RECEIVED. BIT 8 IS THE LEAST SIGNIFICANT BIT.
12	PARITY ERROR THIS BIT IS SET IF THE PARITY OF THE RECEIVED CHARACTER DOES NOT AGREE WITH THAT DESIGNATED FOR THAT LINE.
13	FRAMING ERROR THIS BIT IS SET IF THE RECEIVER SAMPLES A LINE FOR THE FIRST STOP BIT, AND FINDS THE LINE IN A SPACING CONDITION (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 THAT WAS LOST DUE TO THE INABILITY OF THE RECEIVER SCANNER TO SERVICE THE UART RECEIVER HOLDING BUFFER. REFER TO THE SECTION ON PROGRAMMING FOR FURTHER DETAILS ON DOUBLE-BUFFERED RECEPTION.
15	VALID DATA PRESENT THIS BIT INDICATES THAT THE DATA PRESENTED IN BITS 14-00 IS VALID. IT PERMITS A CHARACTER HANDLING PROGRAM TO TAKE CHARACTERS FROM THE SILO UNTIL IT IS EMPTY. THIS IS DONE BY READING THIS REGISTER AND CHECKING BIT 15 UNTIL A WORD IS OBTAINED FOR WHICH BIT 15 IS A ZERO. THE ENTIRE NEXT RECEIVED CHARACTER REGISTER IS READ-ONLY AND IS ADDRESSABLE 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 CONTROL REGISTER HAVE BEEN SET TO SELECT THE 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 THE LENGTH (EXCLUDING PARITY) SHOWN:

BIT 01	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 CODE TO TRANSMIT CHARACTERS HAVING TWO STOP MARKS. IF THE LINE IS TRANSMITTING 5-BIT CODE, ASSERTION OF THIS BIT CAUSES THE CHARACTERS TO BE TRANSMITTED WITH 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 APPROPRIATE PARITY BIT AFFIXED, AND CHARACTERS 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 GENERATED ON THIS LINE AND INCOMING CHARACTERS WILL BE EXPECTED TO HAVE ODD PARITY. IF THIS BIT IS NOT SET, BUT BIT 4 IS SET, CHARACTERS OF EVEN PARITY WILL BE GENERATED ON THIS LINE AND INCOMING CHARACTERS WILL BE EXPECTED TO HAVE EVEN PARITY. IF BIT 4 IS NOT SET, THE SETTIN OF THIS BIT IS IMMATERIAL.

06-09 RECEIVER SPEED

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

10-13 TRANSMITTER SPEED

THE STATE OF THESE BITS DETERMINES THE OPERATING SPEED FOR THIS LINE'S TRANSMITTER. THE SPEED TABLE ON THE NEXT PAGE IS APPLICABLE.

SPEED TABLE FOR RECEIVER AND TRANSMITTER SPEEDS:

SEQ 0039

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

14 HALF DUPLEX/FULL DUPLEX

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

IN THIS APPLICATION HALF-DUPLEX MEANS THAT THE DH11 RECEIVER IS BLINDED DURING TRANSMISSION OF A CHARACTER.

15 AUTO-ECHO ENABLE

WHEN THIS BIT IS SET, CHARACTERS RECEIVED ON THIS LINE WILL BE HARDWARE ECHOED. SEE THE DISCUSSION OF AUTO-ECHO FOR FURTHER DETAILS.

5.2.4 CURRENT ADDRESS REGISTER ADDRESS X06

THIS REGISTER SHOULD BE LOADED ONLY AFTER THE SYSTEM CONTROL REGISTER (SCR) HAS HAD THE APPROPRIATE BITS SET TO SELECT THE DESIRED LINE NUMBER. WHEN THIS REGISTER IS LOADED, ADDRESS BITS 00-15 ARE TRANSFERRED INTO SEMICONDUCTOR MEMORIES IN THE DH11 FROM BITS 00-15 OF THIS REGISTER. ADDRESS BITS 16-17 ARE TRANSFERRED INTO SEMICONDUCTOR 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 BITS 0-3 AND THE READ OR WRITE OF THE CURRENT ADDRESS REGISTER.

WHEN THIS REGISTER IS READ, IT WILL INDICATE THE CURRENT ADDRESS OF THE LINE SELECTED BY THE SYSTEM CONTROL REGISTER. 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 REGISTER SHOULD NOT BE LOADED OR READ WITHOUT FIRST SELECTING A LINE NUMBER BY MEANS OF THE LOWER-ORDER FOUR BITS OF THE SYSTEM CONTROL REGISTER. THIS REGISTER SHOULD BE LOADED WITH THE TWO'S COMPLEMENT OF THE NUMBER OF CHARACTERS (BYTES) TO BE TRANSMITTED ON THAT LINE. THE BYTE COUNT REGISTER IS READ/WRITE.

INTERRUPTS MUST BE INHIBITED OR THE SCR SAVED BETWEEN THE SETTING OF THE SCR BITS 0-3 AND THE READ OR WRITE OF THE BYTE COUNT REGISTER

5.2.6 BUFFER ACTIVE REGISTER (BAR) ADDRESS X12

THIS REGISTER CONTAINS ONE BIT FOR EACH LINE. THE BITS ARE INDIVIDUALLY SET USING BIS INSTRUCTIONS. SETTING A BIT INITIATES TRANSMISSION ON THE ASSOCIATED LINE. THE BIT IS CLEARED BY THE HARDWARE WHEN THE LAST CHARACTER TO BE TRANSMITTED IS LOADED INTO THE TRANSMITTER DATA HOLDING REGISTER OF THE UART FOR THAT LINE. IT SHOULD BE NOTED THAT WHILE THE CLEARING OF A BAR DOES INDICATE THAT A MESSAGE MAY BE SENT, IT DOES NOT INDICATE THAT THE LAST CHARACTERS FROM THE PRECEDING MESSAGE HAVE BEEN COMPLETELY SENT. SPECIFICALLY, TWO MORE CHARACTERS WILL BE SENT AFTER THE BAR BIT CLEARS. THESE ARE THE LAST TWO CHARACTERS OF THE MESSAGE; ONE OF THEM WAS JUST STARTING WHEN THE BAR WAS CLEARED AND ONE WAS THAT FINAL CHARACTER THAT WAS LOADED INTO THE HOLDING REGISTER, THUS CLEARING THE BAR BIT. THIS EFFECT IS A NORMAL CONSEQUENCE OF DOUBLE-BUFFERED TRANSMISSION AND IS MENTIONED HERE FOR THE BENEFIT OF PROGRAMMERS WHO WANT TO WRITE PROGRAMS THAT CONTROL SUCH MODEM LEADS ARE REQUEST TO SEND. REQUEST TO SEND (RTS) SHOULD NOT BE DROPPED UNTIL AT LEAST TWO CHARACTER TIMES AFTER THE BAR BIT FOR A GIVEN LINE CLEARS.

THIS TIMING MAY BE EFFECTED BY SENDING TWO EXTRA (NULL) CHARACTERS IN A MESSAGE AND DROPPING RTS WHEN BAR CLEARS.

CLEARING A BAR BIT SHOULD NOT BE USED TO ABORT TRANSMISSION ON A LINE. RATHER, THE BYTE COUNT FOR THAT LINE SHOULD BE SET 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 WILL IMMEDIATELY GENERATE A BREAK CONDITION ON THE LINE CORRESPONDING TO THAT BIT NUMBER. CLEARING THE BIT WILL TERMINATE THE BREAK CONDITION. THE BREAK CONDITION MAY BE TIMED BY SENDING CHARACTERS DURING THE BREAK INTERVAL, SINCE THESE CHARACTERS WILL NEVER ACTUALLY REACH THE LINE. FURTHER COMMENTS CONCERNING THE TRANSMISSION OF BREAK SIGNALS MAY BE FOUND IN THE BREAK SIGNALS SECTION.

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 LOCATION (E.G., 0, 1, 2, 4, 8, 16, OR 32). WHEN THE NUMBER OF CHARACTERS STORED IN THE SILO EXCEEDS THAT NUMBER, AN INTERRUPT REQUEST (SYSTEM CONTROL REGISTER BIT 7) IS GENERATED, IF SYSTEM CONTROL REGISTER BIT 6 IS SET. THESE BITS ARE READ/WRITE.

06-07 READ EXTENDED MEMORY

THESE BITS ARE READ ONLY AND CONTAIN THE A16 AND A17 BITS OF THE CURRENT LINE ADDRESS WHICH THE LINE 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 CHARACTERS IN THE SILO. IT SHOULD BE NOTED THAT THERE ARE SIX BITS, HENCE NUMBERS BETWEEN 0 AND 63 CAN BE REPRESENTED. A FULL SILO HAS 64 ENTRIES AND THE FILL LEVEL APPEARS AS 00000, BUT ONE MAY EASILY TELL THE DIFFERENCE BETWEEN AN EMPTY SILO (00000) AND A FULL SILO (00000) BY CHECKING THE STORAGE OVERFLOW BIT (BIT 14 OF SYSTEM CONTROL). THESE BITS ARE READ ONLY.

5.3 DH11 MODULE ALLOCATION CHART

VIEW FROM WIRING SIDE

	SLOT 1	2	3	4	5	6	7	8	9
	M920	M7821	M7277	M7287	M7289	M7821	M7360	M7288	M920
	CABLE								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)	LINE PARAMETER CNTL	UNIBUS CONNECTOR (NOTES #1) & #2)
		M796				M405	M971		
							CABLE		
B		UNIBUS MASTER CNTL				EXTERNAL B CLOCK (NOTE #5)	DATA CABLE (NOTES #6 & #9)		
	M7247	M7247				M7280	M7280		M7279
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		FIFO BUFFER
D									
	M105	M7246							M405
E	* ADDRESS SELECTOR (NOTE #7)	* CONTROL SCAN (NOTES #4) & #8							EXTERNAL A CLOCK (NOTE #5)
	M7821								M4540
F	* INTR CNTL (NOTE #7)								DH11 DC11 CLOCK

FIGURE 2-4 DH11 MODULE UTILIZATION DIAGRAM

SEQ 0043

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 DM11-BB 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.

5.0 MAINTENANCE PROCEDURES

THIS SECTION OUTLINES SOME GENERAL TECHNIQUES FOR USING MD-11-DZDHN FOR MAINTENANCE AND CHECKOUT OF THE DH11 SUBSYSTEM. SINCE THIS PROGRAM DOES NOT TEST ALL POSSIBLE DH11 FEATURES (BREAK,AUTO-ECHO, HALF DUPLEX ETC.) THE USER MUST ALSO RUN THE DIAGNOSTIC, MD-11-DZDHM, PRIOR TO USING THIS PROGRAM TO INSURE COMPLETE CHECKOUT AND VERIFICATION OF THE DH11 HARDWARE.

5.1 MAINTENANCE CONNECTORS

BOTH THE DATA RELIABILITY AND PATTERNS/CABLE SUBPROGRAMS REQUIRE THAT THE USER INSTALL THE APPROPRIATE MAINTENANCE JUMPERS 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.

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.

2. TESTING ONE OR MORE SINGLE LINES INCLUDING DATA CABLES

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

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

NOTE: TO TEST THE DEVICE CABLE AS WELL, INSTALL THE H315 TEST CONNECTOR AT THE END OF THE DEVICE CABLE AND LEAVE THE DEVICE CABLE CONNECTED TO THE DISTRIBUTION PANEL.

6.2 DATA RELIABILITY TESTING

A. COMPLETE RELIABILITY TESTING (OVER NIGHT RUNS)

1) SET UP THE TEST JUMPERS AS REQUIRED FOR THE PARTICULAR CONFIGURATION TO BE TESTED. (REFER TO PARA 6.1)
2) LOAD MD-DZDHN AND START IT AT LOC 000200(8).
3) TYPE IN THE DESIRED DH11 PARAMETERS - SET THE SR=000200 AND LET THE PROGRAM RUN.

   A COMPLETE TEST RUN FOR 16. LINES ON EACH DH11 WILL TAKE APPROX 4 HOURS (TWO DH11'S WOULD TAKE 8. HOURS) ETC.

   AT THE COMPLETION OF TESTING FOR EACH DH11 THE ERROR STATISTICS TABLE WILL BE TYPED OUT.

4) LET THE PROGRAM RUN AT LEAST ONE PASS (4 HRS/DH11) PREFERABLY OVERNIGHT, AND THEN ANALYZE ANY ERROR PRINTOUTS AND THE ERROR STATISTIC TABLE DATA.
5) IF ERRORS OCCUR IT SHOULD BE SIMPLE FOR THE USER TO DETERMINE WHICH LINE, WHICH DH11, AND THE FAILING MODES. OF OPERATION TO AID IN FAULT ISOLATION.

B. QUICK DATA RELIABILITY TESTING

1) FOLLOW THE SAME PROCEDURE AS IN PARA 6.2(A) ABOVE EXCEPT SET THE SR=000000(8) BEFORE STARTING THE RUN.

   THE QUICK TESTS VERIFY ALL COMBINATIONS OF LINE PARAMETERS ON ALL LINES AT 9600. BAUD ONLY. ALL OTHER BAUD RATES ARE TESTED WITH 5 BIT CHARS, ONE STOP BIT, AND ODD PARITY ONLY.

2) THE QUICK TEST TAKES APPROX. 15 SECONDS PER LINE SO 2 DH11'S (ALL 16. LINES) COULD BE TESTED IN APPROX 8. MINUTES.

3) THE ERROR INFORMATION PROVIDED IS IDENTICAL TO THAT FOR THE COMPLETE TEST EXCEPT LESS TOTAL DATA TRANSFERS OCCUR.

6.3 DATA PATTERNS TESTING

THE DIAGNOSTIC, MD-11-DZDHM, AND THE DATA RELIABILITY TESTS USE ONLY A BINARY UP COUNT PATTERN FOR DATA TESTING WITH A MAXIMUM BUFFER SIZE OF 256. BYTES. TO PRO-

VIDE DIFFERENT DATA PATTERNS, THE USER CAN RUN THE DATA PATTERNS/CABLE TESTS. THESE TESTS ALLOW HIM TO SIT AT THE CONSOLE TERMINAL AND TEST EACH LINE INDIVIDUALLY WITH VARIOUS PARAMETERS, DATA PATTERNS, BUFFER SIZES, ETC.

1) SET UP THE TEST JUMPERS FOR THE LINES TO BE TESTED AS DESCRIBED IN PARA 6.1.

2) LOAD MD-11-DZDHN AND START IT AT LOC 000220(8) TO RUN THE DATA PATTERNS TESTS.

3) REFER TO PARA 2.1.2.3 FOR THE OPERATING INSTRUCTIONS.

4) ONCE A FAILING PATTERN TEST IS FOUND, THE USER CAN RECONFIGURE THE TEST JUMPERS TO ISOLATE THE FAULT TO EITHER THE DH11 OR A FAULTY CABLE AND/OR CONNECTOR.

6.4 ECHO TESTING

THESE TESTS ALLOW THE USER TO CONNECT AN ASYNCHRONOUS TERMINAL TO THE DH11 DISTRIBUTION PANEL AND VERFIFY THE PARTICULAR LINE AS IT MIGHT BE USED ON-LINE. REFER TO PARA 2.1.2.2 FOR THE OPERATING INSTRUCTIONS FOR THE DH11 ECHO TEST.

THIS IS A HISTORY FILE OF THE DEVELOPMENT OF MD-11-DZDHN

PRODUCT CODE: MD-11-DZDHN VERSION 000.000

PRODUCT NAME: DH11 DATA RELIABILITY EXERCISER AND
SINGLE LINE ECHO/CABLE TESTS

RELEASE DATE: JANUARY 1976

AUTHOR: ED CROWLEY

****

PROCUCT CODE: MD-11-DZDHN-B (REV B)

PRODUCT NAME: DH11 DATA RELIABILITY EXERCISER AND
SINGLE LINE ECHO/CABLE TESTS

RELEASE DATE: SEPTEMBER 1976

SUBMITTED BY: PAUL F. CANTIN

****

PROGRAM CHANGES:

1. A DH11/DM11-BB AUTOSIZER HAS BEEN BUILT AND INSERTED INTO THE PROGRAM TO MAKE IT CHAINABLE UNDER ACT AND APT. CODE WAS ALSO INSERTED TO LOAD AND RUN NON-CONTIGUOUS DH11 ADDRESSES WITHOUT OPREATOR INTERVENTION.

2. PROGRAM UPDATED TO USE SWITCHLESS FEATURE IN SYSMAC REV. C1.

3. THE SENSE OF SR07 HAS BEEN REVERSED SO THAT WITH ALL SWITCHES DOWN, THE PROGRAM PERFORMS A QUICK VERIFY PASS ON ALL PASSES.

4. THE "HALT AFTER PARAMETER SETUP" OPTION SHOULD BE CONTROLLED BY A SWITCH OTHER THAN SR15. CODING HAS BEEN CHANGED TO SENSE SR08 FOR THIS OPTION.

DOCUMENT CHANGES:

1. FLOW CHARTS DELETED FROM DOCUMENT.

2. REFERENCES TO "WORDS BETWEEN VECTORS (4(8) OR 10(8))" HAVE BEEN CHANGED TO "ADDRESSES BETWEEN VECTORS (10(8) OR 20(8))", WITH APPROPRIATE CODING CAHNGES.

3. PARAGRAPH ON SWITCHLESS FEATURE INSERTED.

4. AUTOSIZER STARTING PROCEDURE INSERTED.

5. MADE SEVERAL ADDITIONS TO SECTION 6.1 FOR CLARITY.

****

(5+4302)+

```
                                .NLIST  CND,MD,MC
                                .LIST   TOC,ME,SEQ,BIN
        165000                  $SWR=165000
                                .MCALL  .HEADER,.$CMTAG,.SETUP,.$ERROR
                                .MCALL  .SWRHI,.SWRLO,.$CATCH,.EQUATE,SETUP,.$EOP,GETSWR
                                .MCALL  .$SCOPE,.$ERRTYP,.$TYPOCT,.$TYPDEC,.$TYPE,.$POWER,.$TRAP
                                .MCALL  .$READ,.$RDOCT,.$RDDEC
                                .MCALL  .$ACT11,.$APTHDR,.$APTBLS,.$APTYPE

                                .ENABLE ABS
                                .TITLE  MAINDEC-11-DZDHN-B
                                ;*COPYRIGHT (C) 1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY ED CROWLEY
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE ('MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                                ;*
        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
                                .SBTTL  ACT11 HOOKS

                                ;;******************************************************************
                                ;HOOKS REQUIRED BY ACT11
                                        $SVPC=.                 ;SAVE PC
                                        .=46
        000046  120000                  120000                  ;;1)SET LOC.46 TO ADDRESS OF 120000
                                        .=52
        000052  000000                  .WORD   0               ;;2)SET LOC.52 TO ZERO
                                        .=$SVPC                 ;; RESTORE PC
                                .SBTTL  APT PARAMETER BLOCK

                                ;;******************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;******************************************************************
                                        .$X=.   ;;SAVE CURRENT LOCATION
                                        .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
        000024  000200                  200     ;;FOR APT START UP
                                        .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
        000044  000000                  $APTHDR ;;POINT TO APT HEADER BLOCK
                                        .=.$X   ;;RESET LOCATION COUNTER
                                ;;******************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

        000000                  $APTHD:
```

```
                                ;
000000  000000          $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
000002  001232          $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
000004  001604          $TSTM:  .WORD   1604    ;;RUN TIM OF LONGEST TEST
000006  001604          $PASTM: .WORD   1604    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
000010  001604          $UNITM: .WORD   1604    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
000012  000052                  .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
                        .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  016166          JMP     @#INPARX        ;;JUMP TO STARTING ADDRESS OF PROGRAM

000204  000137  001626          JMP     @#BEGIN         ;BEGIN EXECUTION WITH DEFAULT PARAMETERS
000210  000137  016200          JMP     @#INPARC        ;INPUT PARAMETERS - DEVICE SELECTION ONLY
000214  000137  004706          JMP     @#ECHO          ;GO START LINE ECHO TESTS
000220  000137  006106          JMP     @#EXPAT         ;GO START DATA PATTERNS TESTS
```

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

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

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

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

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
```

```
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

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

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

```
                                .SBTTL  COMMON TAGS

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

                        001100                  .=1100
001100                  001100  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                         .WORD   0
001102     000          001102  $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103     000          001103  $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          001104  $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          001106  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          001110  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          001112  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114     000          001114  $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115     001          001115  $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          001116  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          001120  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          001122  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          001124  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          001126  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0                       ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134     000          001134  $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135     000          001135  $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0

001140  177570          001140  SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          001142  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          001144  $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          001146  $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          001150  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          001152  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154     000          001154  $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155     002          001155  $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156     012          001156  $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157     000          001157  $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          001160  $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                                ;;WHICH  ($REG0) WAS OBTAINED
001162  000000          001162  $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001164  000000          001164  $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001166  000000          001166  $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001170  000000          001170  $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001172  000000          001172  $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001174  000000          001174  $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001176  000000          001176  $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
001200  000000          001200  $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
001202  000000          001202  $TMP0:  .WORD   0               ;;USER DEFINED
001204  000000          001204  $TMP1:  .WORD   0               ;;USER DEFINED
001206  000000          001206  $TMP2:  .WORD   0               ;;USER DEFINED
001210  000000          001210  $TMP3:  .WORD   0               ;;USER DEFINED
001212  000000          001212  $TMP4:  .WORD   0               ;;USER DEFINED
001214  000000          001214  $TMP5:  .WORD   0               ;;USER DEFINED
001216  000000          001216  $TMP6:  .WORD   0               ;;USER DEFINED
001220  000000          001220  $TMP7:  .WORD   0               ;;USER DEFINED
001222  000000          001222  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
```

```
                                ;ESCAPE ON ERROR ADDRESS
                                ;QUESTION MARK
                                ;CARRIAGE RETURN
                                ;LINE FEED
                                ;*******************************************************************
                                .SBTTL  APT MAILBOX-ETABLE


                                ;*******************************************************************
                                .EVEN
                                ;APT MAILBOX
                                ;MESSAGE TYPE CODE
                                ;FATAL ERROR NUMBER
                                ;TEST NUMBER
                                ;PASS COUNT
                                ;DEVICE COUNT
                                ;I/O UNIT NUMBER
                                ;MESSAGE ADDRESS
                                ;MESSAGE LENGTH
                                ;APT ENVIRONMENT TABLE
                                ;ENVIRONMENT BYTE
                                ;ENVIRONMENT MODE BITS
                                ;APT SWITCH REGISTER
                                ;USER SWITCHES
                                ;CPU TYPE,OPTIONS
                                ;*                       BITS 15-11=CPU TYPE
                                ;*                               11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                               11/70=06,PDQ=07,Q=10
                                ;*                       BIT 10=REAL TIME CLOCK
                                ;*                       BIT  9=FLOATING POINT PROCESSOR
                                ;*                       BIT  8=MEMORY MANAGEMENT
                                ;HIGH ADDRESS,M.S. BYTE
                                ;MEM. TYPE,BLK#1
                                ;*                       MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                               900 NSEC CORE=001
                                ;*                               300 NSEC BIPOLAR=002
                                ;*                               500 NSEC MOS=003
                                ;HIGH ADDRESS,BLK#1
                                ;*                         MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
                                ;HIGH ADDRESS,M.S. BYTE
                                ;MEM.TYPE,BLK#2
                                ;MEM.LAST ADDRESS,BLK#2
                                ;HIGH ADDRESS,M.S.BYTE
                                ;MEM.TYPE,BLK#3
                                ;MEM.LAST ADDRESS,BLK#3
                                ;HIGH ADDRESS,M.S.BYTE
                                ;MEM.TYPE,BLK#4
                                ;MEM.LAST ADDRESS,BLK#4
                                ;INTERRUPT VECTOR#1,BUS PRIORITY#1
                                ;INTERRUPT VECTOR#2BUS PRIORITY#2
                                ;BASE ADDRESS OF EQUIPMENT UNDER TEST
                                ;DEVICE MAP
                                ;CONTROLLER DESCRIPTION WORD#1
                                ;CONTROLLER DESCRIPTION WORD#2
                                ;DEVICE DESCRIPTOR WORD#0
                                ;DEVICE DESCRIPTOR WORD#1
                                ;DEVICE DESCRIPTOR WORD#2
```

```
  306  001324  000000                $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
  307  001326  000000                $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
  308  001330  000000                $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
  309  001332  000000                $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
  310  001334  000000                $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
  311  001336  000000                $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
  312  001340  000000                $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
  313  001342  000000                $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
  314  001344  000000                $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
  315  001346  000000                $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
  316  001350  000000                $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
  317  001352  000000                $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
  318  001354  000000                $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
  319
  320
  321  001356                        $ETEND:
  322
```

```
                                .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 1

001356  022442                          EM1             ;"NON EX MEMORY ERROR - DROPPED LINE #  "
001360  022511                          DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
001362  022566                          DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
001364  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 2

001366  022614                          EM2             ;"TRANSMITTER FALSE INTERRUPT - DROPPED LINE#  "
001370  022511                          DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
001372  022566                          DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
001374  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 3

001376  022673                          EM3             ;"BUFFER ACTIVE REGISTER ERROR - DROPPED LINE #  "
001400  022511                          DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
001402  022566                          DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
001404  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 4

001406  022753                          EM4             ;"BYTE COUNT REGISTER ERROR - DROPPED LINE #  "
001410  022511                          DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
001412  022566                          DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
001414  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 5

001416  023030                          EM5             ;"CURRENT ADDRESS REGISTER ERROR - DROPPED LINE #  "
001420  022511                          DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
001422  022566                          DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
001424  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 6

001426  023112                          EM6             ;"SILO OVERFLOW ERROR - DROPPED LINE #  "
001430  022511                          DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
001432  022566                          DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
```

```
001434  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 7

001436  023161                          EM7             ;"RECEIVER FALSE INTERRUPT - LINE #  "
001440  022511                          DH1             ;" (PC)   CURLPR  DEVADR  REGADR   WAS     S/B"
001442  022566                          DT1             ;"$ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4
001444  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 10

001446  023235                          EM10            ;"INVALID DATA IN SILO - DROPPED LINE #  "
001450  023305                          DH2             ; (PC)   CURLPR  CHAR #  WASADR  SHBADR   WAS     S/B"
001452  023372                          DT2             ;$ERRPC,CURLPR,$REG0,$REG1,$REG2,$REG3,$REG4
001454  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 11

001456  023412                          EM11            ;"DATA ERROR - LINE #  "
001460  023305                          DH2             ; (PC)   CURLPR  CHAR #  WASADR  SHBADR   WAS     S/B"
001462  023372                          DT2             ;$ERRPC,CURLPR,$REG0,$REG1,$REG2,$REG3,$REG4
001464  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 12

001466  023440                          EM12            ;"TEST TIMEOUT - DROPPED LINE #  "
001470  023500                          DH3             ;" (PC)   CURLPR  RTOTAL  XTOTAL  RDONE"
001472  023546                          DT3             ;"$ERRPC,CURLPR,$TMP0,$TMP1RDONE"
001474  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 13

001476  000000                          0               ;NO MESSAGE
001500  000000                          0               ;NO DATA HEADER
001502  023562                          DT4             ;$TMP0,$TMP1,$TMP2,$TMP3,$TMP4,$TMP5,$TMP6
001504  023602                          DF1             ;PRINT ALL DECIMAL

                                ;ERROR TABLE ITEM FOR ERROR 14

001506  023612                          EM14            ;"BUS ERROR TRAP TO 04"
001510  023637                          DH4             ;" (PC)    (PS)    (SP)   TRAPPC  TRAPPS"
001512  023706                          DT5             ;$ERRPC,$TMP0,$REG6,$REG1,$REG2"
001514  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 15

001516  023722                          EM15            ;"RSVD INSTR TRAP TO 10"
001520  023637                          DH4             ;" (PC)    (PS)    (SP)   TRAPPC  TRAPPS"
001522  023706                          DT5             ;$ERRPC,$TMP0,$REG6,$REG1,$REG2"
001524  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 16

001526  023750                          EM16            ;"SINGLE LINE ECHO TEST - INTR WAIT TIMEOUT"
001530  024022                          DH5             ;" (PC)   DEVADR   LINE    (SCR)  CURLPR  EXFLAG"
001532  024102                          DT6             ;$ERRPC,$REG1,LINE,$TMP0,CURLPR,EXFLAG"
```

```
001534  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 17

001536  024120                          EM17            ;"ALTERNATING 1/0 PATTERN TEST DONE"
001540  024162                          DH6             ;" (PC)   DEVADR   LINE   CURLPR  ICOUNT"
001542  024232                          DT7             ;$ERRPC,DHADR,LINE,CURLPR,$REGO
001544  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 20

001546  024246                          EM20            ;"BINARY UP COUNT PATTERN TEST DONE"
001550  024162                          DH6             ;" (PC)   DEVADR   LINE   CURLPR  ICOUNT"
001552  024232                          DT7             ;$ERRPC,DHADR,LINE,CURLPR,$REGO
001554  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 21

001556  024310                          EM21            ;"BINARY DOWN COUNT PATTERN TEST DONE"
001560  024162                          DH6             ;" (PC)   DEVADR   LINE   CURLPR  ICOUNT"
001562  024232                          DT7             ;$ERRPC,DHADR,LINE,CURLPR,$REGO
001564  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 22

001566  024354                          EM22            ;"RANDOM DATA PATTERN TEST DONE"
001570  024162                          DH6             ;" (PC)   DEVADR   LINE   CURLPR  ICOUNT"
001572  024232                          DT7             ;$ERRPC,DHADR,LINE,CURLPR,$REGO
001574  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 23

001576  024412                          EM23            ;SINGLE CHAR PATTERN TEST DONE"
001600  024162                          DH6             ;" (PC)   DEVADR   LINE   CURLPR  ICOUNT"
001602  024232                          DT7             ;$ERRPC,DHADR,LINE,CURLPR,$REGO
001604  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 24

001606  024450                          EM24            ;"TYPED BUFFER PATTERN TEST DONE"
001610  024162                          DH6             ;" (PC)   DEVADR   LINE   CURLPR  ICOUNT"
001612  024232                          DT7             ;$ERRPC,DHADR,LINE,CURLPR,$REGO
001614  022604                          DF2             ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR 25

001616  024507                          EM25            ;"DATA PATTERNS TEST TIMEOUT"
001620  024542                          DH7             ;" (PC) DEVADR   LINE   CURLPR  ICOUNT  PATCDE"
001622  024622                          DT10            ;$ERRPC,DHADR,LINE,CURLPR,$REGO,$REG1
001624  022604                          DF2             ;PRINT ALL OCTAL


001626  005000          BEGIN:  CLR     R0              ;INIT R0 TO INDICATE DEFAULT PARAMETERS
001630  005067  020060          CLR     VCFLG           ;INIT VECTOR ADDR SET UP FLAG
```

```
001634  005067  020506          CLR     DPFLG           ;CLEAR DATA PATTERNS TEST FLAG
001640  005067  020514          CLR     RETFLG          ;CLEAR ECHO TEST RETURN FLAG
001644  005067  020466  BEGINA: CLR     TITFLG          ;INIT TITLE MESSAGE FLAG
                        .SBTTL  INITIALIZE THE COMMON TAGS
                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001650  012706  001100          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001654  005026                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001656  022706  001140          CMP     #$WR,R6 ;;DONE?
001662  001374                  BNE     .-6             ;;LOOP BACK IF NO
001664  012706  001100          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                        ;;INITIALIZE A FEW VECTORS
001670  012737  011112  000020  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001676  012737  000340  000022  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001704  012737  011356  000030  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001712  012737  000340  000032  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
001720  012737  014326  000034  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001726  012737  000340  000036  MOV     #340,@#TRAPVEC+2;LEVEL 7
001734  012737  014412  000024  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
001742  012737  000340  000026  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
001750  005067  177246          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
001754  005067  177244          CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
001760  112767  000001  177127  MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
001766  012767  001766  177112  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
001774  012767  001774  177106  MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
002002  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002006  012737  002042  000004  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002014  012767  177570  177116  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002022  012767  177570  177112  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002030  022777  177777  177102  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002036  001012                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
002040  000403                  BR      65$             ;;BRANCH IF NO TIMEOUT
002042  012716  002050  64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
002046  000002                  RTI
002050  012767  000176  177062  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
002056  012767  000174  177056          MOV     #DISPREG,DISPLAY
002064  012637  000004  66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

002070  005067  177144          CLR     $PASS           ;;CLEAR PASS COUNT
002074  132767  000200  177151  BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
002102  001403                  BEQ     67$             ;;YES,USE NON-APT SWITCH
002104  012767  001254  177026  MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
002112                  67$:
                        .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
002112  005737  000042          TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
002116  001012                  BNE     68$             ;;BRANCH IF YES
002120  126727  177126  000001  CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
002126  001406                  BEQ     68$             ;;BRANCH IF YES
002130  026727  177004  000176  CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
002136  001005                  BNE     69$             ;;BRANCH IF NO
002140  104406                  GTSWR                   ;;GET SOFT-SWR SETTINGS
002142  000403                  BR      69$
002144  112767  000001  176762  68$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
002152                  69$:
```

```
                                ;SET UP THE BUS ERROR VECTOR
                                ;SET UP THE RSVD INSTR VECTOR
                                ;HAVE WE TYPED TITLE ONCE ?
                                ;BR IF YES
                                ;GO TYPE PROGRAM TITLE
                                ;SET FLAG - TYPE TITLE ONLY ONCE PER LOAD
                                ;DO WE WANT TO AUTOSIZE?
                                ;BRANCH IF NOT.
                                ;GO AUTOSIZE.
                                ;START AT 200 ??
                                ;BR IF NOT
                                ;ARE PARAMETERS TO BE INPUT MANUALLY?
                                ;BRANCH IF YES
                                ;OTHERWISE, GET ADDRESSES BETWEEN VECOTRS FROM AUTOSIZER
                                ;GO ASK FOR PARAMETERS
                                ;RE INIT START FLAG
                                ;RETURN TO ECHO TESTS ?
                                ;BR IF NOT
                                ;RETURN TO ECHO TEST START-UP
                                ;RETURN TO DATA PATTERNS TEST ?
                                ;BR IF NOT
                                ;GO BACK TO DATA PATTERNS TESTS
                                ;USE DEFAULT PARAMETERS ?
                                ;BR IF YES
                                ;CHANGE DH SELECT PARAM ONLY ?
                                ;BR IF NOT
                                ;GO ASK FOR SELECT PARAM.
                                ;GO ASK FOR ALL PARAMETERS

                                ;GET POINTER TO ADDRESS TABLE
                                ;GET POINTER TO VECTOR TABLE
                                ;GET POINTER TO BR LEVEL TABLE
                                ;START WITH DH #00
                                ;SET UP DH11 BIT TEST MARKER

                                ;GENERATE DH11 DEV NUMBER
                                ;UPDATE TABLE POINTERS

                                ;TEST FOR SELECTED DH11
                                ;BR IF SELECTED FOR TEST
                                ;SHIFT MARKER TO TEST NEXT DH11
                                ;BR IF 16 TESTED - START OVER
                                ;GO TEST IF THIS ONE SELECTED
                                ;SET UP DH11 ADDRESS
                                ;SET UP THE DH11 VECTOR ENTRY
                                ;GET BR LEVEL VALUES
                                ;GO SET DH NUMBER IN THE MESSAGE BUFFER

                                ;GO PRINT "TESTING DH11 #XX"
```

```
                                        MOV     #.,$LPADR       ;INIT SCOPE RETURN
                                ;;**************************************************************
                                ;*TEST 1         SUB-PROGRAM 1 - DATA RELIABILITY TESTS
                                ;;**************************************************************
002506  000004                  TST1:   SCOPE
002510  012767  000001  176504          MOV     #1,$TIMES       ;;DO 1 ITERATION
002516  004767  014414          STDH1:  JSR     PC,CLSTAT       ;GO CLEAR THE STATISTICS TABLES
002522  004767  014774                  JSR     PC,KYBD1        ;GO SET UP FOR KEYBOARD INTRS.
002526  005067  017006                  CLR     QUICK           ;INIT THE QUICK TEST FLAG
002532  005067  017000                  CLR     DRPLIN          ;INIT DROPPED LINE FLAGS
002536  005067  017410                  CLR     LINE            ;INIT LINE NO. TO 00
002542  016702  016756                  MOV     DHVCT,R2        ;SET UP THE VECTORS
002546  012722  003550                  MOV     #RINT1,(R2)+    ;GO TO RINT1 ON RCVR INTR
002552  116722  017370                  MOVB    DHRLVL,(R2)+    ;
002556  105722                          TSTB    (R2)+
002560  012722  003044                  MOV     #TINT1,(R2)+    ;GO TO TINT1 ON XMITTR INTR
002564  116712  017357                  MOVB    DHTLVL,(R2)
002570  016701  016726                  MOV     DHADR,R1        ;SET UP DEVICE ADDRESS

002574  004767  012152          NEWLIN: JSR     PC,SELINE       ;GO SELECT NEW LINE FOR TEST
002600  000401                  1$:     BR      1$              ;BR IF TESTED ALL SELECTED LINES
002602  000402                          BR      2$              ;BR IF NOT DONE
002604  000167  001616          2$:     JMP     PRSTAT          ;GO CHECK AND REPORT STATISTICS
002610  004567  012230                  JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
002614  022152                          LINE
002616  025602                          STMSG2+20
002620  104401                          TYPE                    ;TYPE LINE NO. BEING TESTED
002622  025562                          STMSG2
002624  012767  021544  016746          MOV     #LPRTAB,LPRPTR  ;SET UP LPR TABLE POINTER

002632  004567  014316          NEWLPR: JSR     R5,SETLPR       ;GO SET UP LPR CONSTANT
002636  000756                          BR      NEWLIN          ;BR IF DONE ALL BAUD RATES AT THIS LINE
002640  012767  177760  016734          MOV     #177760,CHRCNT  ;INIT CHAR COUNT
002646  012767  177777  016730          MOV     #-1,CLSEL       ;INIT CHR LNGTH SELECT CODE

002654  005067  016662                  CLR     QUICKX          ;INIT QUICK TEST EXIT FLAG
002660  004567  014354          NEWCL:  JSR     R5,SETCL        ;GO SET UP THE CHAR LENGTH
002664  000762                          BR      NEWLPR          ;BR IF DONE ALL FOUR LENGTHS
002666  005067  016714                  CLR     PARBIT          ;INIT PARITY SELECT BIT CODE

002672  004567  014502          NEWPAR: JSR     R5,SETPAR       ;GO SET PARITY SELECT
002676  000770                          BR      NEWCL           ;BR IF DONE ALL COMBOS

002700  004767  011750          DHST1:  JSR     PC,DHSET1       ;GO SET UP FOR TESTING THIS LINE
002704  056761  016624  000012          BIS     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
002712  004767  016510                  JSR     PC,CHPS1        ;GO CLEAR PSW


002716  012767  000100  017414          MOV     #100,TIMEA      ;INIT TIMER A
002724  005067  017412                  CLR     TIMEB           ;INIT TIMER B
002730  022767  000003  016652  1$:     CMP     #3,RDONE        ;BOTH RCVR AND XMITTR DONE ?
002736  001435                          BEQ     3$              ;BR IF YES
002740  004767  014150                  JSR     PC,TIMEIT       ;CALL THE TIMER
002744  000771                          BR      1$              ;TIMER STEPS AROUND THIS BRANCH IF
                                                                ;TIMEOUT OCCURS
```

```
                                                BIS     #BIT11,@DHADR   ;CLEAR OUT THE DH11
                                                MOVB    LINE,R0         ;GET LINE NO.
                                                ASL     R0              ;FORM TABLE INDEX
                                                MOV     RTOTAL(R0),$TMP0;SAVE XMITTED COUNT
                                                MOV     XTOTAL(R0),$TMP1;SAVE THE RCVD COUNT
                                                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
                                                LINE
                                                EM12+35
                                                MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
                                                ERROR   12              ;LINE FAILED TO FINISH ON TIME - HUNG
                                        2$:     BIS     LINMSK,DRPLIN   ;SET DROP FLAG
                                                MOV     #STACK,SP       ;RESET STACK POINTER
                                                BR      NEWLIN          ;GO TRY ANOTHER LINE


                                        3$:     MOV     #BIT11,(R1)     ;CLEAR THE WORLD OUT IN THE DH11
                                                JSR     PC,CKER         ;GO UPDATE THE DATA ERROR TABLES
                                                BR      NEWPAR          ;GO TRY NEXT PARITY COMBINATION
```

```
                                        ;TRANSMITTER INTERRUPT SERVICE ROUTINE ONE
003044  032711  002000          TINT1:  BIT     #BIT10,(R1)     ;NON EX MEM ERROR ??
003050  001432                          BEQ     2$              ;BR IF NOT

003052  011103                          MOV     (R1),R3         ;SAVE THE SCR
003054  004767  016362                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
003060  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
003064  116704  017062                  MOVB    LINE,R4         ;SET UP THE S/B DATA
003070  042703  175760                  BIC     #175760,R3      ;CLEAR OUT SUPERFLUOUS BITS
003074  010102                          MOV     R1,R2           ;SET UP REGADR
003076  004767  012026                  JSR     PC,SUER1        ;GO SET UP ERROR INFO
003102  004567  011736                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG
003106  022152                          LINE
003110  022506                          EM1+44
003112  012767  003122  175770          MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
003120  104001                          ERROR   1               ;NON EX MEM ERROR
003122  022626                  1$:     CMP     (SP)+,(SP)+     ;POP THE STACK
003124  056767  016404  016404          BIS     LINMSK,DRPLIN   ;SET THE DROPPED FLAG FOR THIS LINE
003132  000167  177436                  JMP     NEWLIN          ;GO TRY NEXT LINE

003136  011103                  2$:     MOV     (R1),R3         ;GET THE SCR REG CONTENTS
003140  100433                          BMI     4$              ;BR IF XMIT DONE SET

003142  004767  016274                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
003146  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE DH11 - FATAL ERROR
003152  012704  100000                  MOV     #BIT15,R4       ;SET UP S/B DATA
003156  156704  016770                  BISB    LINE,R4
003162  042703  077760                  BIC     #77760,R3       ;CLEAR OUT SUPERFLUOUS BITS
003166  010102                          MOV     R1,R2           ;SET UP REGADR
003170  004767  011734                  JSR     PC,SUER1        ;GO SET UP ERROR INFO
003174  004567  011644                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG
003200  022152                          LINE
003202  022670                          EM2+54
003204  012767  003214  175676          MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN
003212  104002                          ERROR   2               ;XMITTR FALSE INTERRUPT
003214  022626                  3$:     CMP     (SP)+,(SP)+     ;POP THE STACK
003216  056767  016312  016312          BIS     LINMSK,DRPLIN   ;SET THE DROPPED FLAG FOR THIS LINE
003224  000167  177344                  JMP     NEWLIN          ;GO TRY NEXT LINE

003230  005761  000012          4$:     TST     BAR(R1)         ;DID BAR BIT CLEAR ??
003234  001432                          BEQ     6$              ;BR IF YES

003236  004767  016200                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
003242  016103  000012                  MOV     BAR(R1),R3      ;GET THE WAS DATA
003246  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE DH11
003252  005004                          CLR     R4              ;SET UP S/B DATA
003254  010102                          MOV     R1,R2           ;SET UP REGADR
003256  062702  000012                  ADD     #BAR,R2
003262  004767  011642                  JSR     PC,SUER1        ;GO SET UP ERROR INFO
003266  004567  011552                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG
003272  022152                          LINE
003274  022750                          EM3+55
003276  012767  003306  175604          MOV     #5$,$LPERR      ;SAVE THE ERROR LOOP RETURN
003304  104003                          ERROR   3               ;BUFFER ACTIVE REG FAILED TO CLEAR
```

```
                                5$:     CMP     (SP)+,(SP)+     ;POP GOES THE STACK
003310  056767  016220  016220          BIS     LINMSK,DRPLIN   ;SET THE DROPPED FLAG FOR THIS LINE
003316  000167  177252                  JMP     NEWLIN          ;GO TRY NEXT LINE

003322  005761  000010          6$:     TST     BCR(R1)         ;DID BYTE COUNT GO TO ZERO ??
003326  001432                  BEQ     8$              ;BR IF YES

003330  004767  016106                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
003334  016103  000010                  MOV     BCR(R1),R3      ;GET THE WAS DATA
003340  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE DH11
003344  005004                          CLR     R4              ;SET UP S/B DATA
003346  010102                          MOV     R1,R2           ;SET UP REGADR
003350  062702  000010                  ADD     #BCR,R2
003354  004767  011550                  JSR     PC,SUER1        ;GO SET UP THE ERROR INFO
003360  004567  011460                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG
003364  022152                          LINE
003366  023025                          EM4+52
003370  012767  003400  175512          MOV     #7$,$LPERR      ;SET UP ERROR LOOP RETURN
003376  104004                          ERROR   4               ;BYTE COUNT REG FAILED TO GO TO 000000
003400  022626                  7$:     CMP     (SP)+,(SP)+     ;POP GOES THE STACK
003402  056767  016126  016126          BIS     LINMSK,DRPLIN   ;SET THE DROPPED FLAG FOR THIS LINE
003410  000167  177160                  JMP     NEWLIN          ;GO TRY NEXT LINE

003414  016103  000006          8$:     MOV     CAR(R1),R3      ;GET THE WAS DATA
003420  016704  016156                  MOV     CHRCNT,R4       ;SET UP S/B DATA
003424  005404                          NEG     R4
003426  062704  032776                  ADD     #TBUF,R4
003432  020304                          CMP     R3,R4           ;WAS CAR CORRECT ??
003434  001425                  BEQ     10$             ;BR IF YES

003436  004767  016000                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
003442  010102                          MOV     R1,R2           ;SET UP REGADR
003444  062702  000006                  ADD     #CAR,R2
003450  004767  011454                  JSR     PC,SUER1        ;GO SET UP ERROR INFO
003454  004567  011364                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG
003460  022152                          LINE
003462  023107                          EM5+57
003464  012767  003474  175416          MOV     #9$,$LPERR      ;SET UP THE ERROR RETURN
003472  104005                          ERROR   5               ;CURRENT ADDRESS REG NOT CORRECT
003474  022626                  9$:     CMP     (SP)+,(SP)+     ;POP THE STACK
003476  056767  016032  016032          BIS     LINMSK,DRPLIN   ;SET THE DROPPED FLAG FOR THIS LINE
003504  000167  177064                  JMP     NEWLIN          ;GO TRY NEXT LINE

003510                          10$:
003510  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
003512  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
003514  016703  016062                  MOV     CHRCNT,R3
003520  005403                          NEG     R3              ;CHAR COUNT IN R3
003522  116704  016424                  MOVB    LINE,R4         ;GET LINE NO.
003526  006304                          ASL     R4              ;DOUBLE IT
003530  060364  030242                  ADD     R3,XTOTAL(R4)   ;UPDATE TOTAL XMIT COUNT
003534  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
003536  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
003540  052767  000001  016042          BIS     #BIT0,RDONE     ;SET XMIT DONE FLAG
003546  000002                          RTI                     ;RETURN TO WAIT LOOP
```

```
                                        ;RECEIVER INTERRUPT SERVICE ROUTINE ONE
                                        ;
003550  032711  040000          RINT1:  BIT     #BIT14,(R1)     ;SILO OVERFLOW ERROR ??
003554  001431                          BEQ     2$              ;BR IF NOT
                                        ;
003556  004767  015660                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
003562  011103                          MOV     (R1),R3         ;GET THE WAS DATA
003564  012711  004000                  MOV     #BIT11,(R1)     ;NOW CLEAR THE DH11
003570  042703  177760                  BIC     #177760,R3      ;CLEAR JUNK
003574  116704  016352                  MOVB    LINE,R4
003600  004767  011324                  JSR     PC,SUER1        ;GO SET UP ERROR INFO
003604  004567  011234                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG
003610  022152                          LINE
003612  023156                          EM6+44
003614  012767  003624  175266          MOV     #1$,$LPERR      ;SET UP ERROR LOOP RETURN
003622  104006                          ERROR   6               ;SILO OVERFLOW - BAD,BAD,BAD !!!
003624  022626                  1$:     CMP     (SP)+,(SP)+     ;POP GOES THE STACK
003626  056767  015702  015702          BIS     LINMSK,DRPLIN   ;SET THE DROPPED FLAG FOR THIS LINE
003634  000167  176734                  JMP     NEWLIN          ;GO TRY NEXT LINE
                                        ;
003640  105711                  2$:     TSTB    (R1)            ;CHAR AVAIL SET ??
003642  100434                          BMI     4$              ;BR IF YES
                                        ;
003644  004767  015572                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
003650  011103                          MOV     (R1),R3         ;GET WAS DATA
003652  042703  177560                  BIC     #177560,R3      ;CLEAN IT UP
003656  012711  004000                  MOV     #BIT11,(R1)     ;NOW CLEAR DH11
003662  012704  000200                  MOV     #BIT07,R4       ;SET UP S/B DATA
003666  156704  016260                  BISB    LINE,R4
003672  010102                          MOV     R1,R2           ;SET UP REGADR
003674  004767  011230                  JSR     PC,SUER1        ;GO SET UP ERROR INFO
003700  004567  011140                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG
003704  022152                          LINE
003706  023232                          EM7+51
003710  012767  003720  175172          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
003716  104007                          ERROR   7               ;RECEIVER FALSE INTERRUPT
003720  022626                  3$:     CMP     (SP)+,(SP)+     ;POP GOES THE SP
003722  056767  015606  015606          BIS     LINMSK,DRPLIN   ;SET THE DROPPED FLAG FOR THIS LINE
003730  000167  176640                  JMP     NEWLINE         ;GO TRY NEXT LINE
                                        ;
003734  016167  000002  175240  4$:     MOV     NRC(R1),$TMPO   ;SAVE THE DATA RECEIVED
003742  100431                          BMI     6$              ;BR IF IT WAS VALID DATA
                                        ;
003744  004767  015472                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
003750  012711  004000                  MOV     #BIT11,(R1)     ;NOW CLEAR THE DH11
003754  162767  030516  024530          SUB     #RBUF,RBFPTR    ;WHICH CHAR WAS IT ??
003762  016702  024524                  MOV     RBFPTR,R2       ;SAVE CHAR NUMBER
003766  004767  011132                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
003772  004567  011046                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG
003776  022152                          LINE
004000  023302                          EM10+45
004002  012767  004012  175100          MOV     #5$,$LPERR      ;SET UP ERROR RETURN
004010  104010                          ERROR   10              ;RECEIVED INVALID DATA
004012  022626                  5$:     CMP     (SP)+,(SP)+     ;POP GOES THE STACK
004014  056767  015514  015514          BIS     LINMSK,DRPLIN   ;SET THE DROPPED FLAG FOR THIS LINE
004022  000167  176546                  JMP     NEWLIN          ;GO TRY ANOTHER LINE
```

```
844
845  004026                          6$:
846  004026  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
847  004030  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
848  004032  016777  175144  024452          MOV     $TMPO,@RBFPTR   ;STORE CHAR IN THE BUFFER
849  004040  062767  000002  024444          ADD     #2,RBFPTR       ;UPDATE THE POINTER
850  004046  026767  015540  024436          CMP     RBFEND,RBFPTR   ;END OF BUFFER ??
851  004054  001013                          BNE     7$              ;BR IF NOT
852  004056  016703  015520                  MOV     CHRCNT,R3       ;GET CHAR COUNT
853  004062  005403                          NEG     R3
854  004064  116704  016062                  MOVB    LINE,R4         ;GET THE LINE NO.
855  004070  006304                          ASL     R4              ;DOUBLE IT
856  004072  060364  030202                  ADD     R3,RTOTAL(R4)   ;UPDATE TOTAL RECEIVED COUNT
857  004076  052767  000002  015504          BIS     #BIT1,RDONE     ;SET THE RCVR DONE BLAG
858  004104                          7$:
859  004104  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
860  004106  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
861  004110  000002                          RTI                     ;RETURN TO WAIT LOOP
```

```
                                        ;THIS ROUTINE IS CALLED TO CHECK THE RECEIVED DATA, REPORT ALL ERRORS,
                                        ;AND UPDATE THE STATISTICS TABLE ENTRIES FOR ALL LINES ACTIVE
004112  012767  030516  024372  CKER:   MOV     #RBUF,RBFPTR    ;SET UP POINTERS
004120  012767  032776  024366          MOV     #TBUF,TBFPTR
004126  012767  030302  024346          MOV     #DATERR,DEPTR   ;SET UP POINTERS TO STATISTICS TABLES
004134  012767  030342  024342          MOV     #PARERR,PEPTR
004142  012767  030402  024336          MOV     #OVRERR,ORPTR
004150  012767  030442  024332          MOV     #FRMERR,FRPTR
004156  116705  015770                  MOVB    LINE,R5         ;GET LINE NO. AND DOUBLE IT
004162  006305                          ASL     R5
004164  060567  024312                  ADD     R5,DEPTR        ;POINT TO CORRECT LINE ENTRY IN TABLE
004170  060567  024310                  ADD     R5,PEPTR
004174  060567  024306                  ADD     R5,ORPTR
004200  060567  024304                  ADD     R5,FRPTR

004204  117704  024304          1$:     MOVB    @TBFPTR,R4      ;GET THE S/B DATA
004210  000304                          SWAB    R4              ;PUT LINE NO. IN HIGH BYTE
004212  105004                          CLRB    R4
004214  156704  015732                  BISB    LINE,R4
004220  000304                          SWAB    R4
004222  052704  100000                  BIS     #BIT15,R4       ;AND FINALLY THE VALID DATA BIT
004226  017703  024260                  MOV     @RBFPTR,R3      ;GET THE WAS DATA
004232  020304                          CMP     R3,R4           ;WAS = S/B ?????
004234  001435                          BEQ     3$              ;BR IF YES

004236  010367  174756                  MOV     R3,$TMP7        ;SAVE THE WAS DATA
004242  010146                          MOV     R1,-(SP)        ;SAVE THE DEVADR
004244  016701  024242                  MOV     RBFPTR,R1       ;GET THE SBADR
004250  016702  024240                  MOV     TBFPTR,R2       ;GET THE WASADR
004254  010200                          MOV     R2,R0           ;GET XMIT BUFFER ADDR
004256  162700  032776                  SUB     #TBUF,R0        ;GENERATE CHAR #
004262  004767  000066                  JSR     PC,UPDER        ;GO CHECK AND UPDATE THE DATA ERROR TABLE
004266  004767  010632                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
004272  004567  010546                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
004276  022152                          LINE
004300  023435                          EM11+23
004302  012767  004312  174600          MOV     #2$,$LPERR      ;SET UP ERROR RETURN
004310  104011                          ERROR   11              ;DATA COMPARE ERROR OR PARITY,FRAMING
                                        ;OR OVERRUN
004312  012601                  2$:     MOV     (SP)+,R1        ;RESTORE THE DEVADR
004314  032767  070000  174676          BIT     #70000,$TMP7    ;ANY PARITY,OVERRUN, OR FRAMING ERROR
004322  001402                          BEQ     3$              ;BR IF NOT

004324  004767  000036                  JSR     PC,SOFT         ;GO TAKE CARE OF SOFT ERROR REPORT

004330  005267  024160          3$:     INC     TBFPTR          ;UPDATE POINTERS
004334  062767  000002  024150          ADD     #2,RBFPTR
004342  026767  015244  024142          CMP     RBFEND,RBFPTR   ;COMPARED ALL CHARS ??
004350  001315                          BNE     1$              ;BR IF NOT

004352  000207                          RTS     PC              ;RETURN TO WAIT LOOP

004354  120304                  UPDER:  CMPB    R3,R4           ;DATA BYTES CORRECT ??
004356  001402                          BEQ     1$              ;BR IF YES
004360  005277  024116                  INC     @DEPTR          ;COUNT THE DATA ERROR
```

```
                                1$:     RTS     PC              ;RETURN

                        174626  SOFT:   ASL     @TMP7           ;TEST FOR OVERRUN ERRORS
                                        BPL     1$              ;BR IF NONE
                        024106          INC     @ORPTR          ;COUNT IT
                        174614  1$:     ASL     @TMP7           ;TEST FOR FRAMING ERRORS
                                        BPL     2$              ;BR IF NONE
                        024076          INC     @FRPTR          ;COUNT IT
                        174602  2$:     ASL     @TMP7           ;TEST FOR PARITY ERRORS
                                        BPL     3$              ;BR IF NONE
                        024060          INC     @PEPTR          ;COUNT IT
                                3$:     RTS     PC              ;RETURN
```

```
                                ;THIS ROUTINE IS CALLED TO PRINT OUT THE TEST STATISTICS
                                
004426  012767  000001  015100  PRSTAT: MOV     #1,LINMSK       ;SET UP BIT TEST MARKER
004434  005001                          CLR     R1              ;R1 CONTAINS THE LINE NO.
004436  004567  010402                  JSR     R5,SUNUM        ;GO SET UP DH11 # IN STAT MESSAGE
                                
004442  022150                          DHNUM
004444  025543                          STMSG1+6
004446  104401                          TYPE                    ;GO TYPE THE STATISTICS HEADER
004450  025535                          STMSG1
004452  104401                          TYPE                    ;TYPE HEADER
004454  025640                          STMSG4
                                
004456  036767  015052  015052  1$:     BIT     LINMSK,DRPLIN   ;DID THIS LINE GET DROPPED ?
004464  001411                          BEQ     2$              ;BR IF NOT
                                
004466  010167  174526                  MOV     R1,$TMP7        ;SAVE THE LINE NO.
004472  004567  010346                  JSR     R5,SJNUM        ;GO PUT LINE NO. IN MESSAGE
004476  001220                          $TMP7
004500  025617                          STMSG3+10
004502  104401                          TYPE
004504  025607                          STMSG3
004506  000436                          BR      3$              ;GO TEST NEXT LINE
                                
004510  010102                  2$:     MOV     R1,R2           ;SET UP R2 WITH TABLE INDEX
004512  006302                          ASL     R2
004514  036767  015014  015010          BIT     LINMSK,LINSEL   ;WAS THIS LINE SELECTED ??
004522  001430                          BEQ     3$              ;BR IF NOT
004524  010167  174452                  MOV     R1,$TMPO        ;SET UP THE ERROR INFORMATION FROM
                                                                ;THE TABLES INTO THE MESSAGE POINTERS
004530  016267  030202  174446          MOV     RTOTAL(R2),$TMP1
004536  016267  030242  174442          MOV     XTOTAL(R2),$TMP2
004544  016267  030302  174436          MOV     DATERR(R2),$TMP3
004552  016267  030342  174432          MOV     PARERR(R2),$TMP4
004560  016267  030402  174426          MOV     OVRERR(R2),$TMP5
004566  016267  030442  174422          MOV     FRMERR(R2),$TMP6
004574  012767  004604  174306          MOV     #3$,$LPERR      ;RETURN TO 3$ AFTER PRINTING LINE
004602  104013                          ERROR   13
004604  005201                  3$:     INC     R1              ;STEP TO NEXT LINE
004606  006367  014722                  ASL     LINMSK          ;SHIFT THE MARKER
004612  001401                          BEQ     ENDA            ;BR IF ALL LINES REPORTED
004614  000720                          BR      1$              ;GO BACK AND DO THIS LINE
```

```
 975  004616  000004                      ENDA:   SCOPE
 976  004620  105067  174256                      CLRB    $ISTNM          ;RE-INIT TEST NUMBER FOR NEXT PASS
 977  004624  012767  000240  004136              MOV     #240,$EOP       ;NOP THE SCOPE IN ENDPASS RCUTINE
 978  004632  005267  015312                      INC     DHNUM           ;GENERATE NEW DH11 NUMBER
 979  004636  062767  000002  015464              ADD     #2,ADPTR        ;UPDATE THE TABLE POINTERS
 980  004644  062767  000002  015460              ADD     #2,VCPTR
 981  004652  062767  000002  015454              ADD     #2,BRPTR
 982  004660  006367  014642                      ASL     SELMSK          ;SHIFT MARKER TO TEST NEXT DH11
 983  004664  001002                              BNE     1$              ;BR IF NOT TESTED ALL DH11'S
 984  004666  000167  004076                      JMP     $EOP            ;JUMP TO EOP IF WE HAVE
 985  004672  036767  014630  014630      1$:     BIT     SELMSK,DHSEL    ;IS THIS DH11 SELECTED ?
 986  004700  001746                              BEQ     ENDA            ;BR IF NOT
 987  004702  000167  175534                      JMP     RSTRTA          ;GO TEST THIS DH11
```

```
                                .SBTTL SUB-PROGRAM 2 - SINGLE LINE ECHO/CABLE TESTS

                                ;               ***********************************
                                ;               *  SINGLE LINE ECHO TESTS         *
                                ;               ***********************************

004706  012767  177777  015444  ECHO:   MOV     #-1,RETFLG      ;SET RETURN FLAG - COME BACK
004714  005067  015426          CLR     DPFLG           ;CLEAR PATTERNS TEST FLAG
004720  005067  014770          CLR     VCFLG           ;INIT VECTOR SETUP FLAG
004724  000167  174714          JMP     BEGINA          ;TO "ECHO1" AFTER SETUP

004730  012767  160020  014564  ECHO1:  MOV     #160020,DHADR   ;SET UP DH11 DEFAULT ADDRESS
004736  012767  000330  014560          MOV     #330,DHVCT      ;SET UP DH11 DEFAULT VECTOR
004744  104401                  TYPE                    ;PRINT I.D. MESSAGE
004746  026252                  ECMSG1                  ;"SINGLE LINE ECHO TEST - CONNECT
                                                        ;TERMINAL TO TEST LINE"
004750  000167  011250    .     JMP     INPAR           ;GO SET UP DEVICE AND VECTOR
                                                        ;ADDRESSES - COME BACK TO "ECHO2"

004754  104401          ECHO2:  TYPE                    ;GO ASK FOR TTY INPUT
004756  026351                  ECMSG2                  ;"LINE # (00 - 17 OCTAL)"
004760  104412                  RDOCT                   ;INPUT LINE NO. FM TTY
004762  012667  015164          MOV     (SP)+,LINE      ;GET NO. TYPED
004766  042767  177760  015156  BIC     #177760,LINE    ;CLEAR JUNK
004774  016702  015152          MOV     LINE,R2         ;GET LINE NO.
005000  005202                  INC     R2              ;CORRECT FOR SHIFT ROUTINE
005002  012767  000001  014524  MOV     #1,LINMSK       ;INIT LINE SELECT BIT MASK
005010  005302          1$:     DEC     R2              ;COUNT ONE LINE CHECKED
005012  001403                  BEQ     2$              ;BR IF DONE
005014  006367  014514          ASL     LINMSK          ;SHIFT SELECT BIT
005020  000773                  BR      1$              ;GO COUNT IT
005022  004767  013204          2$:     JSR     PC,LPRIN        ;GO ASK FOR AND SET UP LINE PARAMETERS

005026  005767  015314          TST     DPFLG           ;DATA PATTERNS TEST ?
005032  001401                  BEQ     3$              ;BR IF NOT
005034  000207                  RTS     PC              ;RETURN TO PATTERNS TEST

005036  105067  021201          3$:     CLRB    EC2             ;CLEAR ECHO BUFFER
005042  104401                  TYPE
005044  027373                  SNMSG1                  ;"SEND MODE - Y OR N ?"
005046  104410          4$:     RDCHR                   ;GET CHAR TYPED
005050  012600                  MOV     (SP)+,R0        ;GET CHAR TYPED
005052  122700  000015          CMPB    #15,R0          ;WAS IT A <CR> ?
005056  001405                  BEQ     5$              ;BR IF YES
005060  110067  021157          MOVB    R0,EC2          ;ECHO WHAT WAS TYPED
005064  104401                  TYPE
005066  026243                  EC2
005070  000766                  BR      4$              ;GO WAIT FOR TERMINATOR
005072  105767  021145          5$:     TSTB    EC2             ;<CR> ONLY ??
005076  001412                  BEQ     ECHO3           ;BR IF YES
005100  122767  000116  021135  CMPB    #116,EC2        ;WAS IT AN "N" ??
005106  001406                  BEQ     ECHO3           ;BR IF YES
005110  122767  000131  021125  CMPB    #131,EC2        ;WAS IT A "Y" ??
005116  001347                  BNE     3$              ;BR IF NOT ASK AGAIN
005120  000167  000574          JMP     SENDP1          ;GO TO SEND ROUTINE
```

```
1044  005124  004767  014312          ECHO3:  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
1045  005130  005067  015210                  CLR     CEXIT           ;INIT CONTROL-C EXIT FLAG
1046  005134  005067  015232                  CLR     EXFLAG          ;CLEAR TEST EXIT FLAGS
1047  005140  012767  120240  015000          MOV     #120240,DHRLVL  ;INIT FOR BR LEVEL 5
1048  005146  016701  014350                  MOV     DHADR,R1        ;SET UP DEVICE ADDRESS
1049  005152  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE SELECTED DH11
1050  005156  016700  014342                  MOV     DHVCT,R0        ;GET THE FIRST VECTOR ADDRESS
1051  005162  012720  005476                  MOV     #RINT2,(R0)+    ;SET UP THE VECTORS
1052  005166  116710  014754                  MOVB    DHRLVL,(R0)
1053  005172  005720                          TST     (R0)+
1054  005174  012720  005330                  MOV     #TINT2,(R0)+
1055  005200  116710  014743                  MOVB    DHTLVL,(R0)
1056  005204  016711  014742                  MOV     LINE,(R1)       ;SET THE LINE SELECT BITS
1057  005210  012702  032776                  MOV     #TBUF,R2        ;INIT BUFFER POINTER
1058  005214  052711  000100                  BIS     #BIT06,(R1)     ;ENABLE  RCVR INTRS
1059  005220  016761  014352  000004          MOV     CURLPR,LPR(R1)  ;SET UP LINE PARAMETERS
1060  005226  004567  007612                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
1061  005232  022152                          LINE
1062  005234  026427                          ECMSG3+20
1063  005236  104401                          TYPE                    ;GIVE DIRECTIONS
1064  005240  026407                          ECMSG3                  ;"GO TYPE CHARS ON TEST TERMINAL"
1065  005242  104401                          TYPE                    ;PRINT DIRECTIONS
1066  005244  026466                          ECMSG4                  ;"[CONTROL-C TO EXIT] [CONTROL-E TO ECHO BUFFER]"
1067  005246  004767  014154                  JSR     PC,CHPS1        ;GO CLEAR PSW
1068
1069  005252  012767  000100  015060  DHWAIT: MOV     #100,TIMEA      ;INIT TIMER "A"
1070  005260  005067  015056                  CLR     TIMEB           ;INIT TIMER "B"
1071  005264  005767  015054          1$:     TST     CEXIT           ;CONTROL-C EXIT ??
1072  005270  001015                          BNE     2$              ;BR IF YES
1073  005272  004767  011616                  JSR     PC,TIMEIT       ;CALL TIMER
1074  005276  000772                          BR      1$              ;BR IF NO TIMEOUT
1075
1076  005300  010167  173660                  MOV     R1,$REG1        ;SAVE DEVADR
1077  005304  011167  173672                  MOV     (R1),$TMP0      ;SAVE CONTENT OF SCR
1078  005310  052711  004000                  BIS     #BIT11,(R1)     ;CLEAR OUT THE DH11
1079  005314  012767  005324  173566          MOV     #2$,$LPERR      ;SET ERROR LOOP RETURN
1080  005322  104016                          ERROR   16              ;REPORT RCVR WAIT TIMEOUT
1081  005324  000167  177424          2$:     JMP     ECHO2           ;GO RESTART
1082
1083
```

```
1084                                        ;TRANSMITTER INTERRUPT SERVICE ROUTINE TWO
1085
1086  005330  042711  120000          TINT2:  BIC     #BIT15+BIT13,(R1)       ;DISABLE XMIT INTRS
1087  005334  022767  000001  015030          CMP     #1,EXFLAG               ;CONTROL-C FLAG ?
1088  005342  001437                          BEQ     2$                      ;BR IF YES
1089  005344  022767  000003  015020          CMP     #3,EXFLAG               ;WAS BUFFER JUST DUMPED ?
1090  005352  001437                          BEQ     3$                      ;BR IF YES
1091  005354  022767  000002  015010          CMP     #2,EXFLAG               ;CONTROL-E FLAG ?
1092  005362  001403                          BEQ     1$                      ;BR IF YES
1093  005364  020227  034126                  CMP     R2,#TBUF+600.           ;BUFFER FULL ?
1094  005370  002434                          BLT     31$                     ;BR IF NOT
1095  005372  012767  000003  014772  1$:     MOV     #3,EXFLAG               ;SET DUMP FLAG
1096  005400  162702  032776                  SUB     #TBUF,R2                ;SET UP BYTE COUNT REG
1097  005404  005402                          NEG     R2
1098  005406  010261  000010                  MOV     R2,BCR(R1)
1099  005412  012761  032776  000006          MOV     #TBUF,CAR(R1)           ;SET UP CURRENT ADDR REG
1100  005420  012767  000100  014712          MOV     #100,TIMEA              ;INIT TIMER
1101  005426  052711  020000                  BIS     #BIT13,(R1)             ;ENABLE XMITTR INTR
1102  005432  016761  014076  000012          MOV     LINMSK,BAR(R1)          ;ACTIVATE LINE
1103  005440  000415                          BR      4$                      ;GO EXIT
1104
1105  005442  012767  177777  014674  2$:     MOV     #-1,CEXIT               ;SET CONTROL-C EXIT
1106  005450  000411                          BR      4$                      ;GO EXIT
1107
1108  005452  012702  032776          3$:     MOV     #TBUF,R2                ;RESET ECHO BUFFER PIOPINTER
1109  005456  005067  014710                  CLR     EXFLAG                  ;INIT EXIT FLAG
1110  005462  012767  000100  014650  31$:    MOV     #100,TIMEA              ;INIT TIMER AGAIN
1111  005470  052711  000100                  BIS     #BIT06,(R1)             ;ENABLE RCVR INTR
1112
1113  005474  000002          4$:     RTI                                     ;RETURN TO MAINLINE
1114
1115
```

```
                                        ;RECEIVER INTERRUPT SERVICE ROUTINE TWO
                                        ;
005476  042711  000100          RINT2:  BIC     #BIT06,(R1)             ;DISABLE RCVR INTR
005502  016167  000002  173472          MOV     NRC(R1),$TMP0           ;SAVE THE DATA TYPED
005510  042767  177600  173464          BIC     #177600,$TMP0           ;CLEAR HIGH BYTE
005516  022767  000003  173456          CMP     #3,$TMP0                ;CONTROL-C TYPED ??
005524  001015                          BNE     1$                      ;BR IF NOT
                                        ;
005526  112767  000136  014640          MOVB    #136,ECBUF              ;SET UP TO ECHO CONTROL-C
005534  112767  000103  014633          MOVB    #103,ECBUF+1
005542  012761  177776  000010          MOV     #-2,BCR(R1)             ;SET UP BCR REG
005550  012767  000001  014614          MOV     #1,EXFLAG               ;SET CONTROL-C FLAG
005556  000444                          BR      4$                      ;GO OUT PUT CHAR TYPED
                                        ;
005560  022767  000005  173414  1$:     CMP     #5,$TMP0                ;WAS IT A CONTROL-E ??
005566  001021                          BNE     2$                      ;BR IF NOT
                                        ;
005570  112767  000136  014576          MOVB    #136,ECBUF              ;SET UP TO ECHO CONTROL-E
005576  112767  000105  014571          MOVB    #105,ECBUF+1
005604  112722  000136                  MOVB    #136,(R2)+              ;PUT IN THE ECHO BUFFER
005610  112722  000105                  MOVB    #105,(R2)+
005614  012761  177776  000010          MOV     #-2,BCR(R1)             ;SET UP BYTE COUNT REG
005622  012767  000002  014542          MOV     #2,EXFLAG               ;SET CONTROL-E FLAG
005630  000417                          BR      4$                      ;GO EXIT
                                        ;
005632  022767  000012  173342  2$:     CMP     #12,$TMP0               ;WAS IT A LINE FEED ??
005640  001003                          BNE     3$                      ;BR IF NOT
005642  004767  012060                  JSR     PC,LDFILL               ;GO LOAD FILLERS
005646  000410                          BR      4$                      ;GO EXIT
                                        ;
005650  116767  173326  014516  3$:     MOVB    $TMP0,ECBUF             ;SET UP CHAR TO ECHO
005656  116722  173320                  MOVB    $TMP0,(R2)+
005662  012761  177777  000010          MOV     #-1,BCR(R1)             ;OUTPUT ONE CHAR ONLY
005670  012761  022374  000006  4$:     MOV     #ECBUF,CAR(R1)          ;SET UP CURRENT ADDR REG
005676  012767  000100  014434          MOV     #100,TIMEA              ;INIT TIMER AGAIN
005704  052711  020000                  BIS     #BIT13,(R1)             ;ENABLE XMITTR INTR
005710  016761  013620  000012          MOV     LINMSK,BAR(R1)          ;ACTIVATE THE LINE
005716  000002                          RTI                             ;RETURN TO MAINLINE
```

```
1157  005720  104401                  SENDP1: TYPE                    ;ASK FOR DIRECTIONS
1158  005722  027423                          SNMSG2                  ;"TYPE SEND BUFFER - TERMINATE WITH CONTROL-C"
1159  005724  012705  032776                  MOV     #TBUF,R5        ;SET UP BUFFER POINTER
1160  005730  104410                  1$:     RDCHR                   ;GET CHAR
1161  005732  012600                          MOV     (SP)+,R0
1162  005734  110067  020303                  MOVB    R0,EC2          ;ECHO CHAR
1163  005740  104401                          TYPE
1164  005742  026243                          EC2
1165  005744  022700  000003                  CMP     #3,R0           ;WAS IT A CONTROL-C ??
1166  005750  001421                          BEQ     4$              ;BR IF YES
1167
1168  005752  026727  013624  000400          CMP     CHRCNT,#256.    ;BUFFER FULL ??
1169  005760  003015                          BGT     4$              ;BR IF YES
1170  005762  022700  000012                  CMP     #12,R0          ;WAS IT A LINE FEED ?
1171  005766  001010                          BNE     3$              ;BR IF NOT
1172
1173  005770  110025                          MOVB    R0,(R5)+        ;LOAD CHAR TYPED
1174  005772  116704  014442                  MOVB    FILLB,R4        ;GET FILLER COUNT
1175  005776  116725  014434          2$:     MOVB    FILLA,(R5)+     ;LOAD A FILLER
1176  006002  005304                          DEC     R4              ;COUNT IT
1177  006004  001374                          BNE     2$              ;BR IF NOT DONE
1178  006006  000750                          BR      1$              ;GET SOME MORE INPUT
1179
1180  006010  110025                  3$:     MOVB    R0,(R5)+        ;LOAD BUFFER
1181  006012  000746                          BR      1$              ;GO GET SOME MORE
1182
1183  006014  004767  011630          4$:     JSR     PC,SENDP2       ;GO XMIT THE BUFFER
1184  006020  105067  020217          5$:     CLRB    EC2             ;CLEAR ECHO BUFFER
1185  006024  104401                          TYPE
1186  006026  027503                          SNMSG3                  ;"CHANGE PARAMETERS- Y OR N"
1187  006030  104410                  6$:     RDCHR
1188  006032  012600                          MOV     (SP)+,R0        ;GET CHAR
1189  006034  122700  000015                  CMPB    #15,R0          ;WAS IT A <CR> HE TYPED ??
1190  006040  001405                          BEQ     7$              ;BR IF IT WAS
1191  006042  110067  020175                  MOVB    R0,EC2          ;ECHO IT
1192  006046  104401                          TYPE
1193  006050  026243                          EC2
1194  006052  000766                          BR      6$              ;GO WAIT FOR TERMINATOR
1195
1196  006054  105767  020163          7$:     TSTB    EC2             ;<CR> ONLY ??
1197  006060  001717                          BEQ     SENDP1          ;BR IF YES
1198  006062  122767  000116  020153          CMPB    #116,EC2        ;DID HE SAY NO ??
1199  006070  001713                          BEQ     SENDP1          ;BR IF HE DID
1200  006072  122767  000131  020143          CMPB    #131,EC2        ;DID HE SAY YES ??
1201  006100  001347                          BNE     5$              ;GO ASK ALL OVER AGAIN
1202  006102  000167  176646                  JMP     ECHO2           ;GO ASK FOR NEW PARAMETERS
```

```
                                .SBTTL SUB-PROGRAM THREE - DATA PATTERNS TESTS

                                ;                       ****************************
                                ;                       * DATA PATTERNS TESTS     *
                                ;                       ****************************

006106  012767  177777  014232  EXPAT:  MOV     #-1,DPFLG       ;SET PATTERNS TEST FLAG
006114  005067  014240          CLR     RETFLG          ;CLR ECHO TESTS FLAG
006120  005067  013570          CLR     VCFLG           ;CLEAR VECTOR SETUP FLAG
006124  000167  173514          JMP     BEGINA          ;GO SET UP RETURN TO "EXPAT1"

006130  012767  160020  013364  EXPAT1: MOV     #160020,DHADR   ;SET UP DEFAULT DH11 ADDR
006136  012767  000330  013360          MOV     #330,DHVCT      ;AND VECTOR TOO
006144  104401                        TYPE
006146  027542                        DPMSG1                  ;"DATA PATTERNS TESTS - CONNECT TEST JUMPAR"
006150  000167  010050          JMP     INPAR           ;GO GET SOME PARAMETERS RETURN TO EXPAT2

006154  004767  176574  EXPAT2: JSR     PC,ECHO2        ;GO GET REST OF THE PARAMETERS
006160  004767  013170          JSR     PC,SUCLMK       ;GO SET UP CHAR LENGTH MASK
006164  104401          1$:     TYPE
006166  027620                  DPMSG2                  ;"BUFFER SIZE ? (1-512)
006170  104413                  RDDEC                   ;GET THE SIZE TYPED
006172  012600                  MOV     (SP)+,R0
006174  001406                  BEQ     2$              ;BR IF DEFAULT TO 256. <CR>

006176  020027  001001          CMP     R0,#513.        ;TOO BIG ?
006202  002405                  BLT     3$              ;BR IF NOT
006204  104401                  TYPE
006206  027652                  DPMSG3                  ;"INVALID SIZE - TRY AGAIN"
006210  000765                  BR      1$              ;GO ASK AGAIN

006212  012700  000400  2$:     MOV     #256.,R0        ;DEFAULT TO 256. BYTE BUFFER
006216  005400          3$:     NEG     R0              ;MAKE IT NEG BYTE COUNT
006220  010067  013356          MOV     R0,CHRCNT       ;SAVE IT FOR TEST

006224  012767  120240  013714  MOV     #120240,DHRLVL  ;SET BR LEVELS TO BR5
006232  016700  013266          MOV     DHVCT,R0        ;SET UP VECTORS
006236  012720  010266          MOV     #RINT3,(R0)+
006242  116710  013700          MOVB    DHRLVL,(R0)
006246  005720                  TST     (R0)+
006250  012720  007644          MOV     #TINT3,(R0)+
006254  116710  013667          MOVB    DHTLVL,(R0)
```

```
1245  006260  104407                  EXPAT3: CKSWR                   ;TEST FOR CHANGE IN SOFT-SWR
1246  006262  005067  014066                  CLR     PATFLG          ;CLEAR <CR> SEQUENCE FLAG
1247  006266  105067  017751                  CLRB    EC2             ;CLEAR ECHO BUFFER
1248  006272  016701  013224                  MOV     DHADR,R1        ;INIT R1 TO POINT TO SCR REG
1249  006276  104401                          TYPE
1250  006300  027707                          DPMSG4                  ;"PATTERN TYPE ? (A,U,D,R,S, OR B)"
1251  006302  104410                  7$:     RDCHR
1252  006304  012600                          MOV     (SP)+,R0        ;GET WHAT HE TYPED
1253  006306  120027  000015                  CMPB    R0,#15          ;WAS IT A <CR> ??
1254  006312  001407                          BEQ     9$              ;BR IF YES
1255  006314  010067  014032                  MOV     R0,DATPAT
1256  006320  110067  017717                  MOVB    R0,EC2          ;ECHO IT
1257  006324  104401                          TYPE
1258  006326  026243                          EC2
1259  006330  000764                          BR      7$              ;GO WAIT FOR TERMINATOR
1260
1261  006332  104401                  9$:     TYPE
1262  006334  027761                          DPMSG5                  ;"SET SR07=1 TO LOCK ON TEST PATTERN"
1263  006336  105767  017701                  TSTB    EC2             ;<CR> ONLY ??
1264  006342  001005                          BNE     8$              ;BR IF NOT
1265  006344  012767  000015  014002          MOV     #15,PATFLG
1266  006352  000167  000506                  JMP     DPATCR          ;GO SEQUENCE A,U,D,R PATTERNS
1267
1268  006356  022767  000101  013766  8$:     CMP     #101,DATPAT     ;ALTERNATING 1/0 ?
1269  006364  001002                          BNE     1$              ;BR IF NOT
1270  006366  000167  000102                  JMP     DPATA           ;GO DO IT
1271
1272  006372  022767  000125  013752  1$:     CMP     #125,DATPAT     ;UP COUNT PATTERN ?
1273  006400  001002                          BNE     2$              ;BR IF NOT
1274  006402  000167  000164                  JMP     DPATU           ;GO DO IT
1275
1276  006406  022767  000104  013736  2$:     CMP     #104,DATPAT     ;DOWN COUNT PATTERN ?
1277  006414  001002                          BNE     3$              ;BR IF NOT
1278  006416  000167  000246                  JMP     DPATD           ;GO DO IT
1279
1280  006422  022767  000122  013722  3$:     CMP     #122,DATPAT     ;RANDOM PATTERN ?
1281  006430  001002                          BNE     4$              ;BR IF NOT
1282  006432  000167  000330                  JMP     DPATR           ;GO DO IT
1283
1284  006436  022767  000123  013706  4$:     CMP     #123,DATPAT     ;SINGLE CHAR PATTERN ?
1285  006444  001002                          BNE     5$              ;BR IF NOT
1286  006446  000167  000474                  JMP     DPATS           ;GO DO IT
1287
1288  006452  022767  000102  013672  5$:     CMP     #102,DATPAT     ;TYPE IN BUFFER ?
1289  006460  001002                          BNE     6$              ;BR IF NOT
1290  006462  000167  000620                  JMP     DPATB           ;GO DO IT
1291
1292  006466  104401                  6$:     TYPE
1293  006470  030023                          DPMSG6                  ;"INVALID PATTERN - TRY AGAIN"
1294  006472  000672                          BR      EXPAT3          ;GO ASK AGAIN
1295
```

```
1296  006474  005067  013650          DPATA:  CLR     DATCNT          ;INIT ITERATION COUNTER
1297  006500  004767  012432          1$:     JSR     PC,SUPATA       ;GO SET UP THE PATTERN
1298  006504  004767  001016                  JSR     PC,DHST2        ;GO EXECUTE IT ON SELECTED DH11
1299  006510  005267  013634                  INC     DATCNT          ;COUNT IT
1300  006514  026767  013642  013626          CMP     PATLIM,DATCNT   ;DONE IT ENOUGH TIMES
1301  006522  001366                          BNE     1$              ;BR IF NOT DO IT AGAIN
1302
1303  006524  016767  013620  172430          MOV     DATCNT,$REGO    ;SAVE ITERATION COUNT
1304  006532  005067  013612                  CLR     DATCNT          ;INIT COUNTER
1305  006536  012767  006546  172344          MOV     #2$,$LPERR      ;COME BACK TO 2$
1306  006544  104017                          ERROR   17              ;REPORT DONE SPECIFIED NO. OF ITERATIONS
1307  006546  022767  000015  013600  2$:     CMP     #15,PATFLG      ;CYCLING FOUR PATTERNS ?
1308  006554  001001                          BNE     3$              ;BR IF NOT
1309  006556  000207                          RTS     PC              ;RETURN TO EXECUTE NEXT PATTERN
1310  006560  105777  172354          3$:     TSTB    @SWR            ;LOCK ON THIS PATTERN ??
1311  006564  100745                          BMI     1$              ;BR IF YES
1312  006566  000167  177466                  JMP     EXPAT3          ;GO ASK FOR NEW PATTERNS
1313
1314  006572  005067  013552          DPATU:  CLR     DATCNT          ;INIT ITERATION COUNTER
1315  006576  004767  012362          1$:     JSR     PC,SUPATU       ;GO SET UP THE PATTERN
1316  006602  004767  000720                  JSR     PC,DHST2        ;GO EXECUTE IT ON SELECTED DH11
1317  006606  005267  013536                  INC     DATCNT          ;COUNT IT
1318  006612  026767  013544  013530          CMP     PATLIM,DATCNT   ;DONE IT ENOUGH TIMES ?
1319  006620  001366                          BNE     1$              ;BR IF NOT DO IT AGAIN
1320
1321  006622  016767  013522  172332          MOV     DATCNT,$REGO    ;SAVE ITERATION COUNT
1322  006630  005067  013514                  CLR     DATCNT          ;INIT COUNTER
1323  006634  012767  006644  172246          MOV     #2$,$LPERR      ;COME BACK TO 2$
1324  006642  104020                          ERROR   20              ;REPORT DONE SPECIFIED NO. OF ITERATIONS
1325  006644  022767  000015  013502  2$:     CMP     #15,PATFLG      ;CYCLING FOUR PATTERNS ?
1326  006652  001001                          BNE     3$              ;BR IF NOT
1327  006654  000207                          RTS     PC              ;RETURN TO EXECUTE NEXT PATTERN
1328  006656  105777  172256          3$:     TSTB    @SWR            ;LOCK ON THIS PATTERN ??
1329  006662  100745                          BMI     1$              ;BR IF YES
1330  006664  000167  177370                  JMP     EXPAT3          ;GO ASK FOR NEW PATTERNS
1331
1332  006670  005067  013454          DPATD:  CLR     DATCNT          ;INIT ITERATION COUNTER
1333  006674  004767  012314          1$:     JSR     PC,SUPATD       ;GO SET UP THE PATTERN
1334  006700  004767  000622                  JSR     PC,DHST2        ;GO EXECUTE IT ON SELECTED DH11
1335  006704  005267  013440                  INC     DATCNT          ;COUNT IT
1336  006710  026767  013446  013432          CMP     PATLIM,DATCNT   ;DONE IT ENOUGH TIMES
1337  006716  001366                          BNE     1$              ;BR IF NOT DO IT AGAIN
1338
1339  006720  016767  013424  172234          MOV     DATCNT,$REGO    ;SAVE ITERATION COUNT
1340  006726  005067  013416                  CLR     DATCNT          ;INIT COUNTER
1341  006732  012767  006742  172150          MOV     #2$,$LPERR      ;COME BACK TO 2$
1342  006740  104021                          ERROR   21              ;REPORT DONE SPECIFIED NO. OF ITERATIONS
1343  006742  022767  000015  013404  2$:     CMP     #15,PATFLG      ;CYCLING FOUR PATTERNS ?
1344  006750  001001                          BNE     3$              ;BR IF NOT
1345  006752  000207                          RTS     PC              ;RETURN TO EXECUTE NEXT PATTERN
1346  006754  105777  172160          3$:     TSTB    @SWR            ;LOCK ON THIS PATTERN ??
1347  006760  100745                          BMI     1$              ;BR IF YES
1348  006762  000167  177272                  JMP     EXPAT3          ;GO ASK FOR NEW PATTERNS
1349
1350  006766  005067  013356          DPATR:  CLR     DATCNT          ;INIT ITERATION COUNTER
1351  006772  004767  012250          1$:     JSR     PC,SUPATR       ;GO SET UP THE PATTERN
```

```
1352  006776  004767  000524                  JSR     PC,DHST2        ;GO EXECUTE IT ON SELECTED DH11
1353  007002  005267  013342                  INC     DATCNT          ;COUNT IT
1354  007006  026767  013350  013334          CMP     PATLIM,DATCNT   ;DONE IT ENOUGH TIMES
1355  007014  001366                          BNE     1$              ;BR IF NOT DO IT AGAIN
1356
1357  007016  016767  013326  172136          MOV     DATCNT,$REGO    ;SAVE ITERATION COUNT
1358  007024  005067  013320                  CLR     DATCNT          ;INIT COUNTER
1359  007030  012767  007040  172052          MOV     #2$,$LPERR      ;COME BACK TO 2$
1360  007036  104022                          ERROR   22              ;REPORT DONE SPECIFIED NO. OF ITERATIONS
1361  007040  022767  000015  013306  2$:     CMP     #15,PATFLG      ;CYCLING FOUR PATTERNS ?
1362  007046  001001                          BNE     3$              ;BR IF NOT
1363  007050  000207                          RTS     PC              ;RETURN TO EXECUTE NEXT PATTERN
1364  007052  105777  172062          3$:     TSTB    @SWR            ;LOCK ON THIS PATTERN ??
1365  007056  100745                          BMI     1$              ;BR IF YES
1366  007060  000167  177174                  JMP     EXPAT3          ;GO ASK FOR NEW PATTERNS
1367
1368  007064  012767  000101  013260  DPATCR: MOV     #101,DATPAT     ;FLAG 1/0 PATTERN
1369  007072  004767  177376                  JSR     PC,DPATA        ;CALL FOR 1/0 PATTERN
1370  007076  012767  000125  013246          MOV     #125,DATPAT     ;FLAG UP COUNT PATTERN
1371  007104  004767  177462                  JSR     PC,DPATU        ;CALL FOR UP COUNT PATTERN
1372  007110  012767  000104  013234          MOV     #104,DATPAT     ;FLAG DOWN COUNT PATTERN
1373  007116  004767  177546                  JSR     PC,DPATD        ;CALL FOR DOWN COUNT PATTERN
1374  007122  012767  000122  013222          MOV     #122,DATPAT     ;FLAG RANDOM DATA PATTERN
1375  007130  004767  177632                  JSR     PC,DPATR        ;CALL FO RANDOM PATTERN
1376  007134  105777  172000                  TSTB    @SWR            ;LOCK ON ALL FOUR PATTERNS
1377  007140  100751                          BMI     DPATCR          ;BR IF YES
1378  007142  000167  177112                  JMP     EXPAT3          ;GO ASK FOR NEW PATTERN
1379
1380  007146  105067  017071  DPATS:  CLRB    EC2             ;CLEAR THE ECHO BUFFER
1381  0071_   005067  013172                  CLR     DATCNT          ;INIT ITERATION COUNTER
1382  007156  104401                          TYPE
1383  007160  030063                          DPMSG7                  ;"TYPE SINGLE TEST CHAR"
1384  007162  104410                  3$:     RDCHR                   ;GET CHAR
1385  007164  012600                          MOV     (SP)+,R0        ;GET WHAT HE TYPED
1386  007166  122700  000015                  CMPB    #15,R0          ;WAS IT A <CR> ??
1387  007172  001407                          BEQ     4$              ;BR IF YES
1388  007174  010067  013156                  MOV     R0,SINGLE       ;SAVE IT FOR LOADING BUFFER
1389  007200  110067  017037                  MOVB    R0,EC2          ;ECHO IT ON TTY
1390  007204  104401                          TYPE
1391  007206  026243                          EC2
1392  007210  000764                          BR      3$              ;GO WAIT FOR TERMINATOR
1393
1394  007212  105767  017025          4$:     TSTB    EC2             ;WAS SINGLE CHAR A <CR> ??
1395  007216  001003                          BNE     1$              ;BR IF NOT A <CR> ONLY
1396  007220  012767  000015  013130          MOV     #15,SINGLE      ;SET UP TO LOAD ALL <CR>'S
1397  007226  004767  012074          1$:     JSR     PC,$UPATS       ;GO SET IT UP IN BUFFER
1398  007232  004767  000270                  JSR     PC,DHST2        ;GO EXECUTE IT ON DH11
1399  007236  005267  013106                  INC     DATCNT          ;COUNT ONE TIME
1400  007242  026767  013114  013100          CMP     PATLIM,DATCNT   ;DONE REQUIRED ITERATIONS ?
1401  007250  001366                          BNE     1$              ;BR IF NOT
1402
1403  007252  016767  013072  171702          MOV     DATCNT,$REGO    ;SAVE ITERATION COUNT
1404  007260  005067  013064                  CLR     DATCNT          ;INIT ITERATION COUNTER
1405  007264  012767  007274  171616          MOV     #2$,$LPERR      ;COME BACK TO 2$ ALWAYS
1406  007272  104023                          ERROR   23              ;REPORT DONE SINGLE CHAR PATTERN
1407  007274  105777  171640          2$:     TSTB    @SWR            ;LOCK ON THIS PATTERN ??
```

```
1408  007300  100752          BMI     1$          ;BR IF YES
1409  007302  000167  176752  JMP     EXPAT3      ;GO ASK FOR NEW PATTERN
1410
```

```
1411  007306  005067  013036          DPATB:  CLR     DATCNT          ;INIT ITERATION COUNTER
1412  007312  104401                          TYPE
1413  007314  030116                          DPMSGA                  ;"TYPE IN BUFFER - TERMINATE WITH CONTROL-C"
1414  007316  012705  032776                  MOV     #TBUF,R5        ;POINT TO XMIT BUFFER
1415  007322  104410                  1$:     RDCHR                   ;GET A CHAR
1416  007324  012667  013026                  MOV     (SP)+,SINGLE    ;SAVE IT
1417  007330  116767  013022  016705          MOVB    SINGLE,EC2      ;ECHO IT
1418  007336  104401                          TYPE
1419  007340  026243                          EC2
1420
1421  007342  022767  000003  013006          CMP     #3,SINGLE       ;WAS IT A CONTROL-C ??
1422  007350  001423                          BEQ     3$              ;BR IF YES
1423
1424  007352  020527  033777                  CMP     R5,#TBUF+513.   ;BUFFER FULL ??
1425  007356  001420                          BEQ     3$              ;BR IF YES
1426
1427  007360  022767  000012  012770          CMP     #12,SINGLE      ;WAS IT A LINE FEED ??
1428  007366  001011                          BNE     2$              ;BR IF NOT
1429
1430  007370  016700  013044                  MOV     FILLB,R0        ;LOAD LF PLUS FILLERS
1431  007374  116725  012756                  MOVB    SINGLE,(R5)+
1432  007400  116725  013032          11$:    MOVB    FILLA,(R5)+     ;LOAD A FILLER CHAR
1433  007404  005300                          DEC     R0
1434  007406  001374                          BNE     11$             ;BR TILL REQUIRED FILLERS LOADED
1435  007410  000744                          BR      1$              ;GO ASK FOR ANOTHER CHAR
1436
1437  007412  116725  012740          2$:     MOVB    SINGLE,(R5)+    ;LOAD IT IN BUFFER
1438  007416  000741                          BR      1$              ;GO GET NEXT CHAR
1439

1440  007420  112767  000136  016617  3$:     MOVB    #136,EC3                ;ECHO CONTROL-C
1441  007426  112767  000103  016612          MOVB    #103,EC3+1
1442  007434  104401                          TYPE
1443  007436  026245                          EC3
1444  007440  162705  032776                  SUB     #TBUF,R5        ;SET UP CHAR COUNT
1445  007444  005405                          NEG     R5
1446  007446  010567  012130                  MOV     R5,CHRCNT
1447  007452  004767  000050          4$:     JSR     PC,DHST2        ;GO EXECUTE PATTERN
1448  007456  005267  012666                  INC     DATCNT
1449  007462  026767  012674  012660          CMP     PATLIM,DATCNT   ;DONE REQUIRED ITERATIONS
1450  007470  001370                          BNE     4$              ;BR IF NOT
1451
1452  007472  016767  012652  171462          MOV     DATCNT,$REG0    ;SAVE ITERATION COUNT
1453  007500  005067  012644                  CLR     DATCNT          ;INIT ITERATION COUNTER
1454  007504  012767  007514  171376          MOV     #5$,$LPERR      ;RETURN TO 5$
1455  007512  104024                          ERROR   24              ;DONE REQUIRED ITERATIONS
1456  007514  105777  171420          5$:     TSTB    @SWR            ;LOCK ON THIS BUFFER ??
1457  007520  100754                          BMI     4$              ;BR IF YES
1458  007522  000167  176532                  JMP     EXPAT3          ;GO ASK FOR NEW PATTERN
1459
```

```
1460  007526  004767  005122          DHST2:  JSR     PC,DHSET1       ;GO SET UP THE DH11
1461  007532  056761  011776  000012          BIS     LINMSK,BAR(R1)  ;ACTIVATE THE LINE
1462  007540  004767  011662                  JSR     PC,CHPS1        ;GO CLEAR PSW
1463
1464  007544  012767  000100  012566  PTWAIT: MOV     #100,TIMEA      ;INIT TIMERS
1465  007552  005067  012564                  CLR     TIMEB
1466  007556  005767  012026          1$:     TST     RDONE           ;DONE ENTIRE PATTERN ?
1467  007562  001023                          BNE     3$              ;BR IF YES
1468  007564  004767  007324                  JSR     PC,TIMEIT       ;CALL THE TIMER
1469  007570  000772                          BR      1$              ;EXECUTED IF NO TIMEOUT
1470
1471  007572  012777  004000  011722          MOV     #BIT11,@DHADR   ;CLEAR THE DH11
1472  007600  016767  012544  171354          MOV     DATCNT,$REG0    ;SAVE ITERATION COUNTER
1473  007606  016767  012540  171350          MOV     DATPAT,$REG1    ;SAVE PATTERN TYPE
1474  007614  012767  007624  171266          MOV     #2$,$LPERR      ;SET UP ERROR RETURN
1475  007622  104025                          ERROR   25              ;DATA PATTERNS TEST TIMEOUT ERROR
1476
1477  007624  005726                  2$:     TST     (SP)+           ;FIX STACK SINCE WE ARE SKIPPING RTS
1478  007626  000167  176426                  JMP     EXPAT3          ;GO ASK FOR NEW PATTERN
1479
1480  007632  052711  004000          3$:     BIS     #BIT11,(R1)     ;CLEAR THE DH11
1481  007636  004767  000740                  JSR     PC,CKERDP       ;GO CHECK DATA BUFFERS
1482  007642  000207                          RTS     PC              ;RETURN TO CONTROL ROUTINE
1483
1484
```

```
1485
1486                                    ;TRANSMITTER INTERRUPT SERVICE ROUTINE THREE
1487
1488  007644  032711  002000          TIN3:  BIT     #BIT10,(R1)      ;NON EX MEM ERROR ??
1489  007650  001430                        BEQ     2$               ;BR IF NOT
1490
1491  007652  011103                        MOV     (R1),R3          ;SAVE THE SCR
1492  007654  004767  011562                JSR     PC,CHPS2         ;GO LOCK OUT INTRS
1493  007660  012711  004000                MOV     #BIT11,(R1)      ;CLEAR OUT THE DH11
1494  007664  116704  012262                MOVB    LINE,R4          ;SET UP THE S/B DATA
1495  007670  042703  175760                BIC     #175760,R3       ;CLEAR OUT SUPERFLUOUS BITS
1496  007674  010102                        MOV     R1,R2            ;SET UP REGADR
1497  007676  004767  005226                JSR     PC,SUER1         ;GO SET UP ERROR INFO
1498  007702  004567  005136                JSR     R5,SUNUM         ;GO SET UP LINE NO. IN MSG
1499  007706  022152                        LINE
1500  007710  022506                        EM1+44
1501  007712  012767  007722  171170        MOV     #1$,$LPERR       ;SET UP THE ERROR LOOP RETURN
1502  007720  104001                        ERROR   1                ;NON EX MEM ERROR
1503  007722  022626                  1$:   CMP     (SP)+,(SP)+      ;POP THE STACK
1504  007724  005726                        TST     (SP)+            ;FIX STACK SINCE NO RTS IS EXECUTED
1505  007726  000167  176326                JMP     EXPAT3           ;GO ASK FOR NEW PATTERN
1506
1507  007732  011103                  2$:   MOV     (R1),R3          ;GET THE SCR REG CONTENTS
1508  007734  100431                        BMI     4$               ;BR IF XMIT DONE SET
1509
1510  007736  004767  011500                JSR     PC,CHPS2         ;GO LOCK OUT INTRS
1511  007742  012711  004000                MOV     #BIT11,(R1)      ;CLEAR THE DH11 - FATAL ERROR
1512  007746  012704  100000                MOV     #BIT15,R4        ;SET UP S/B DATA
1513  007752  156704  012174                BISB    LINE,R4
1514  007756  042703  077760                BIC     #77760,R3        ;CLEAR OUT SUPERFLUOUS BITS
1515  007762  010102                        MOV     R1,R2            ;SET UP REGADR
1516  007764  004767  005140                JSR     PC,SUER1         ;GO SET UP ERROR INFO
1517  007770  004567  005050                JSR     R5,SUNUM         ;GO SET UP LINE NO. IN MSG
1518  007774  022152                        LINE
1519  007776  022670                        EM2+54
1520  010000  012767  010010  171102        MOV     #3$,$LPERR       ;SET UP ERROR LOOP RETURN
1521  010006  104002                        ERROR   2                ;XMITTR FALSE INTERRUPT
1522  010010  022626                  3$:   CMP     (SP)+,(SP)+      ;POP THE STACK
1523  010012  005726                        TST     (SP)+            ;FIX STACK SINCE NO RTS IS EXECUTED
1524  010014  000167  176240                JMP     EXPAT3           ;GO ASK FOR NEW PATTERN
1525
1526  010020  005761  000012          4$:   TST     BAR(R1)          ;DID BAR BIT CLEAR ??
1527  010024  001430                        BEQ     6$               ;BR IF YES
1528
1529  010026  004767  011410                JSR     PC,CHPS2         ;GO LOCK OUT INTRS
1530  010032  016103  000012                MOV     BAR(R1),R3       ;GET THE WAS DATA
1531  010036  012711  004000                MOV     #BIT11,(R1)      ;CLEAR THE DH11
1532  010042  005004                        CLR     R4               ;SET UP S/B DATA
1533  010044  010102                        MOV     R1,R2            ;SET UP REGADR
1534  010046  062702  000012                ADD     #BAR,R2
1535  010052  004767  005052                JSR     PC,SUER1         ;GO SET UP ERROR INFO
1536  010056  004567  004762                JSR     R5,SUNUM         ;GO SET UP LINE NO. IN MSG
1537  010062  022152                        LINE
1538  010064  022750                        EM3+55
1539  010066  012767  010076  171014        MOV     #5$,$LPERR       ;SAVE THE ERROR LOOP RETURN
1540  010074  104003                        ERROR   3                ;BUFFER ACTIVE REG FAILED TO CLEAR
```

```
1541  010076  022626                  5$:   CMP   (SP)+,(SP)+    ;POP GOES THE STACK
1542  010100  005726                        TST   (SP)+           ;FIX STACK SINCE NO RTS IS EXECUTED
1543  010102  000167  176152              JMP   EXPAT3            ;GO ASK FOR NEW PATTERN
1544
1545  010106  005761  000010        6$:   TST   BCR(R1)         ;DID BYTE COUNT GO TO ZERO ??
1546  010112  001430                      BEQ   8$              ;BR IF YES
1547
1548  010114  004767  011322              JSR   PC,CHPS2        ;GO LOCK OUT INTRS
1549  010120  016103  000010              MOV   BCR(R1),R3      ;GET THE WAS DATA
1550  010124  012711  004000              MOV   #BIT11,(R1)     ;CLEAR THE DH11
1551  010130  005004                      CLR   R4              ;SET UP S/B DATA
1552  010132  010102                      MOV   R1,R2           ;SET UP REGADR
1553  010134  062702  000010              ADD   #BCR,R2
1554  010140  004767  004764              JSR   PC,SUER1        ;GO SET UP THE ERROR INFO
1555  010144  004567  004674              JSR   R5,SUNUM        ;GO SET UP LINE NO. IN MSG
1556  010150  022152                      LINE
1557  010152  023025                      EM4+52
1558  010154  012767  010164  170726      MOV   #7$,$LPERR      ;SET UP ERROR LOOP RETURN
1559  010162  104004                      ERROR 4               ;BYTE COUNT REG FAILED TO GO TO 000000
1560  010164  022626              7$:     CMP   (SP)+,(SP)+     ;POP GOES THE STACK
1561  010166  005726                      TST   (SP)+           ;FIX STACK SINCE NO RTS IS EXECUTED
1562  010170  000167  176064              JMP   EXPAT3          ;GO ASK FOR NEW PATTERN
1563
1564  010174  016103  000006        8$:   MOV   CAR(R1),R3      ;GET THE WAS DATA
1565  010200  016704  011376              MOV   CHRCNT,R4       ;SET UP S/B DATA
1566  010204  005404                      NEG   R4
1567  010206  062704  032776              ADD   #TBUF,R4
1568  010212  020304                      CMP   R3,R4           ;WAS CAR CORRECT ??
1569  010214  001423                      BEQ   10$             ;BR IF YES
1570
1571  010216  004767  011220              JSR   PC,CHPS2        ;GO LOCK OUT INTRS
1572  010222  010102                      MOV   R1,R2           ;SET UP REGADR
1573  010224  062702  000006              ADD   #CAR,R2
1574  010230  004767  004674              JSR   PC,SUER1        ;GO SET UP ERROR INFO
1575  010234  004567  004604              JSR   R5,SUNUM        ;GO SET UP LINE NO. IN MSG
1576  010240  022152                      LINE
1577  010242  023107                      EM5+57
1578  010244  012767  010254  170636      MOV   #9$,$LPERR      ;SET UP THE ERROR RETURN
1579  010252  104005                      ERROR 5               ;CURRENT ADDRESS REG NOT CORRECT
1580  010254  022626              9$:     CMP   (SP)+,(SP)+     ;POP THE STACK
1581  010256  005726                      TST   (SP)+           ;FIX STACK SINCE NO RTS IS EXECUTED
1582  010260  000167  175774              JMP   EXPAT3          ;GO ASK FOR NEW PATTERN
1583
1584  010264  000002              10$:    RTI
```

```
                                        ;RECEIVER INTERRUPT SERVICE ROUTINE THREE
                                        ;
010266  032711  040000          RINT3:  BIT     #BIT14,(R1)     ;SILO OVERFLOW ERROR ??
010272  001427                  BEQ     2$              ;BR IF NOT
                                        ;
010274  004767  011142                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
010300  011103                  MOV     (R1),R3         ;GET THE WAS DATA
010302  012711  004000          MOV     #BIT11,(R1)     ;NOW CLEAR THE DH11
010306  042703  177760          BIC     #177760,R3      ;CLEAR JUNK
010312  116704  011634          MOVB    LINE,R4
010316  004767  004606          JSR     PC,SUER1        ;GO SET UP ERROR INFO
010322  004567  004516          JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG
010326  022152                  LINE
010330  023156                  EM6+44
010332  012767  010342  170550  MOV     #1$,$LPERR      ;SET UP ERROR LOOP RETURN
010340  104006                  ERROR   6               ;SILO OVERFLOW - BAD,BAD,BAD !!!
010342  022626          1$:     CMP     (SP)+,(SP)+     ;POP GOES THE STACK
010344  005726                  TST     (SP)+           ;FIX STACK SINCE NO RTS IS EXECUTED
010346  000167  175706          JMP     EXPAT3          ;GO ASK FOR NEW PATTERN
                                        ;
010352  105711          2$:     TSTB    (R1)            ;CHAR AVAIL SET ??
010354  100432                  BMI     4$              ;BR IF YES
                                        ;
010356  004767  011060                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
010362  011103                  MOV     (R1),R3         ;GET WAS DATA
010364  042703  177560          BIC     #177560,R3      ;CLEAN IT UP
010370  012711  004000          MOV     #BIT11,(R1)     ;NOW CLEAR DH11
010374  012704  000200          MOV     #BIT07,R4       ;SET UP S/B DATA
010400  156704  011546          BISB    LINE,R4
010404  010102                  MOV     R1,R2           ;SET UP REGADR
010406  004767  004516          JSR     PC,SUER1        ;GO SET UP ERROR INFO
010412  004567  004426          JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG
010416  022152                  LINE
010420  023232                  EM7+51
010422  012767  010432  170460  MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
010430  104007                  ERROR   7               ;RECEIVER FALSE INTERRUPT
010432  022626          3$:     CMP     (SP)+,(SP)+     ;POP GOES THE SP
010434  005726                  TST     (SP)+           ;FIX STACK SINCE NO RTS IS EXECUTED
010436  000167  175616          JMP     EXPAT3          ;GO ASK FOR NEW PATTERN
                                        ;
010442  016167  000002  170532  4$:     MOV     NRC(R1),$TMPO   ;SAVE THE DATA RECEIVED
010450  100427                  BMI     6$              ;BR IF IT WAS VALID DATA
                                        ;
010452  004767  010764                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
010456  012711  004000          MOV     #BIT11,(R1)     ;NOW CLEAR THE DH11
010462  162767  030516  020022  SUB     #RBUF,RBFPTR    ;WHICH CHAR WAS IT ??
010470  016702  020016          MOV     RBFPTR,R2       ;SAVE CHAR NUMBER
010474  004767  004424          JSR     PC,SUER2        ;GO SET UP ERROR INFO
010500  004567  004340          JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG
010504  022152                  LINE
010506  023302                  EM10+45
010510  012767  010520  170372  MOV     #5$,$LPERR      ;SET UP ERROR RETURN
010516  104010                  ERROR   10              ;RECEIVED INVALID DATA
010520  022626          5$:     CMP     (SP)+,(SP)+     ;POP GOES THE STACK
010522  005726                  TST     (SP)+           ;FIX STACK SINCE NO RTS IS EXECUTED
010524  000167  175530          JMP     EXPAT3          ;GO ASK FOR NEW PATTERN
```

```
 1641
 1642  010530  016777  170446  017754  6$:     MOV     $TMP0,@RBFPTR   ;STORE CHAR IN THE BUFFER
 1643  010536  062767  000001  017746          ADD     #1,RBFPTR       ;UPDATE THE POINTER
 1644  010544  026767  011042  017740          CMP     RBFEND,RBFPTR   ;END OF BUFFER ??
 1645  010552  001407                          BEQ     7$              ;BR IF YES
 1646  010554  062767  000001  017730          ADD     #1,RBFPTR
 1647  010562  026767  011024  017722          CMP     RBFEND,RBFPTR
 1648  010570  001003                          BNE     8$              ;BR IF NOT DONE
 1649  010572  052767  000001  011010  7$:     BIS     #1,RDONE        ;SET SOFTWARE DONE FLAG
 1650  010600  000002                  8$:     RTI                     ;RETURN TO WAIT LOOP
```

```
                                        ;THIS ROUTINE IS CALLED TO CHECK THE RECEIVED DATA AND REPORT ALL ERRORS
                                        ;FOR THE DATA PATTERNS TESTS
                                        
010602  012767  030516  017702  CKERDP: MOV     #RBUF,RBFPTR    ;SET UP POINTERS
010610  012767  032776  017676          MOV     #TBUF,TBFPTR
                                        
010616  117704  017672          1$:     MOVB    @TBFPTR,R4      ;GET THE S/B DATA
010622  000304                          SWAB    R4              ;PUT LINE NO. IN HIGH BYTE
010624  105004                          CLRB    R4
010626  156704  011320                  BISB    LINE,R4
010632  000304                          SWAB    R4
010634  052704  100000                  BIS     #BIT15,R4       ;AND FINALLY THE VALID DATA BIT
010640  046704  011524                  BIC     CLMSK,R4        ;MASK OFF BITS NOT XMITTED
010644  017703  017642                  MOV     @RBFPTR,R3      ;GET THE WAS DATA
010650  020304                          CMP     R3,R4           ;WAS = S/B ?????
010652  001425                          BEQ     3$              ;BR IF YES
                                        
010654  010367  170340                  MOV     R3,$TMP7        ;SAVE THE WAS DATA
010660  010146                          MOV     R1,-(SP)        ;SAVE THE DEVADR
010662  016701  017624                  MOV     RBFPTR,R1       ;GET THE SBADR
010666  016702  017622                  MOV     TBFPTR,R2       ;GET THE WASADR
010672  010200                          MOV     R2,R0           ;GET XMIT BUFFER ADDR
010674  162700  032776                  SUB     #TBUF,R0        ;GENERATE CHAR #
010700  004767  004220                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
010704  004567  004134                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
010710  022152                          LINE
010712  023435                          EM11+23
010714  012767  010724  170166          MOV     #2$,$LPERR      ;SET UP ERROR RETURN
010722  104011                          ERROR   11              ;DATA COMPARE ERROR OR PARITY,FRAMING
                                                                ;OR OVERRUN
010724  012601                  2$:     MOV     (SP)+,R1        ;RESTORE THE DEVADR
                                        
010726  005267  017562          3$:     INC     TBFPTR          ;UPDATE POINTERS
010732  062767  000001  017552          ADD     #1,RBFPTR
010740  026767  010646  017544          CMP     RBFEND,RBFPTR   ;COMPARED ALL CHARS ??
010746  001407                          BEQ     4$              ;BR IF YES
010750  062767  000001  017534          ADD     #1,RBFPTR       ;UPDATE IT AGAIN
010756  026767  010630  017526          CMP     RBFEND,RBFPTR   ;DONE YET ?
010764  001314                          BNE     1$              ;BR IF NOT
                                        
010766  000207                  4$:     RTS     PC              ;RETURN TO WAIT LOOP
                                        
```

```
                                        .SBTTL  END OF PASS ROUTINE
                                        ;;*****************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO START2
010770                                  $EOP:
010770  000004                                SCOPE
010772  005067  170104                       CLR     $ISTNM          ;;ZERO THE TEST NUMBER
010776  005067  170220                       CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
011002  005267  170232                       INC     $PASS           ;;INCREMENT THE PASS NUMBER
011006  042767  100000  170224               BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
011014  005327                               DEC     (PC)+           ;;LOOP?
011016  000001                        $EOPCT: .WORD   1
011020  003022                               BGT     $DOAGN          ;;YES
011022  012737                               MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
011024  000001                        $ENDCT: .WORD   1
011026  011016                                $EOPCT
011030  104401  011075                       TYPE    ,$ENDMG         ;;TYPE "END PASS #"
011034  016746  170200                       MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
011040  104405                               TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
011042  104401  011072                       TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
011046  013700  000042                $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
011052  001405                               BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
011054  000005                               RESET                   ;;CLEAR THE WORLD
011056  004710                        $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
011060  000240                               NOP                     ;;SAVE ROOM
011062  000240                               NOP                     ;;FOR
011064  000240                               NOP                     ;;ACT11
011066                                  $DOAGN:
011066  000137                               JMP     @(PC)+          ;;RETURN
011070  002336                        $RTNAD: .WORD   START2
011072     377     377     000        $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
011075     015  042412  042116        $ENDMG: .ASCIZ  <15><12>/END PASS #/
011102  050040  051501  020123
011110  000043
                                        .SBTTL  SCOPE HANDLER ROUTINE
                                        ;;*****************************************************************
                                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                        ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW14=1        LOOP ON TEST
                                        ;*SW11=1        INHIBIT ITERATIONS
                                        ;*SW09=1        LOOP ON ERROR
                                        ;*CALL
                                        ;*      SCOPE           ;;SCOPE=IOT
011112                                  $SCOPE:
011112  104407                               CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
011114  005067  011032                       CLR     LINE            ;INIT LINE COUNTER
011120  032777  040000  170012        1$:    BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
011126  001104                               BNE     $OVER           ;;YES IF SW14=1
```

```
                                        ;*****START OF CODE FOR THE XOR TESTER*****
011130  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
011132  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
011136  012737  011156  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
011144  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
011150  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
011154  000453                          BR      $SVLAD          ;;GO TO THE NEXT TEST
011156  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
011160  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
011164  000413                          BR      7$              ;;LOOP ON THE PRESENT TEST
011166                          6$:;*****END OF CODE FOR THE XOR TESTER*****
011166  105767  167711          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
011172  001421                          BEQ     3$              ;;BR IF NO
011174  126767  167715  167701          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
011202  101015                          BHI     3$              ;;BR IF NO
011204  032777  001000  167726          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
011212  001404                          BEQ     4$              ;;BR IF NO
011214  016767  167670  167664  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
011222  000446                          BR      $OVER
011224  105067  167653          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
011230  005067  167766                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
011234  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
011236  032777  004000  167674  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
011244  001011                          BNE     1$              ;;BR IF YES
011246  005767  167766                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
011252  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
011254  005267  167624                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
011260  026767  167736  167616          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
011266  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
011270  012767  000001  167606  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
011276  016767  000052  167716          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
011304  105267  167572          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
011310  116767  167566  167720          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
011316  011667  167564                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
011322  011667  167562                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
011326  005067  167672                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
011332  112767  000001  167555          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
011340  016777  167536  167574  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
011346  016716  167534                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
011352  000002                          RTI                     ;;FIXES PS
011354  000010                  $MXCNT: 10                      ;;MAX. NUMBER OF ITERATIONS
                                .SBTTL  ERROR HANDLER ROUTINE

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

011356                          $ERROR:
```

```
1805  011356  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
1806  011360  105267  167517          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
1807  011364  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
1808  011366  016777  167510  167546          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
1809  011374  005267  167512                  INC     $ERTTL          ;;INC THE ERROR COUNT
1810  011400  011667  167512                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
1811  011404  162767  000002  167504          SUB     #2,$ERRPC
1812  011412  117767  167500  167474          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
1813  011420  032777  020000  167512          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
1814  011426  001004                          BNE     20$             ;;SKIP TYPEOUTS
1815  011430  004767  000074                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
1816  011434  104401  001227                  TYPE    ,$CRLF
1817  011440                          20$:
1818  011440  122767  000001  167604          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
1819  011446  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
1820  011450  116767  167440  000004          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
1821  011456  004767  001174                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
1822  011462     000          21$:    .BYTE   0
1823  011463     000                  .BYTE   0
1824  011464  000777          22$:    BR      22$             ;;APT ERROR LOOP
1825  011466  005777  167446          2$:     TST     @SWR            ;;HALT ON ERROR
1826  011472  100002                  BPL     3$              ;;SKIP IF CONTINUE
1827  011474  000000                  HALT                    ;;HALT ON ERROR!
1828  011476  104407                  CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
1829  011500  032777  001000  167432  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
1830  011506  001402                  BEQ     4$              ;;BR IF NO
1831  011510  016716  167374                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
1832  011514  005767  167504          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
1833  011520  001402                  BEQ     5$              ;;BR IF NONE
1834  011522  016716  167476                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
1835  011526                          5$:
1836  011526  000002                  RTI                     ;;RETURN
1837                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
1838
1839                                  ;;**************************************************************
1840                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
1841                                  ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
1842                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
1843
1844  011530                          $ERRTYP:
1845  011530  104401  001227                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
1846  011534  010046                          MOV     R0,-(SP)        ;;SAVE R0
1847  011536  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
1848  011540  153700  001114                  BISB    @#$ITEMB,R0
1849  011544  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
1850                                                                  ;;TYPE THE PC OF THE ERROR
1851  011546  016746  167344                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
1852                                                                  ;;ERROR ADDRESS
1853  011552  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1854  011554  000445                          BR      10$             ;;GET OUT
1855  011556  005300          1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
1856  011560  006300                  ASL     R0              ;;      WORK FOR THE ERROR TABLE
1857  011562  006300                  ASL     R0
1858  011564  006300                  ASL     R0
1859  011566  062700  001356                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
1860  011572  012067  000004                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
```

```
1861  011576  001404                      BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
1862  011600  104401                      TYPE                    ;;TYPE THE "ERROR MESSAGE"
1863  011602  000000              2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
1864  011604  104401  001227              TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
1865  011610  012067  000004      3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
1866  011614  001404                      BEQ     5$              ;;SKIP TYPEOUT IF 0
1867  011616  104401                      TYPE                    ;;TYPE THE "DATA HEADER"
1868  011620  000000              4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
1869  011622  104401  001227              TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
1870  011626  010146              5$:     MOV     R1,-(SP)        ;;SAVE R1
1871  011630  012001                      MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
1872  011632  001415                      BEQ     9$              ;;BR IF NO DATA TO BE TYPED
1873  011634  012000                      MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
1874  011636  105720              6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
1875  011640  001003                      BNE     7$              ;;BR IF DECIMAL
1876  011642  013146                      MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
1877  011644  104402                      TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1878  011646  000402                      BR      8$
1879  011650              7$:
1880  011650  013146                      MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
1881  011652  104405                      TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
1882  011654  005711              8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
1883  011656  001403                      BEQ     9$              ;;BR IF NO
1884  011660  104401  011700              TYPE    ,11$            ;;TYPE TWO(2) SPACES
1885  011664  000764                      BR      6$              ;;LOOP
1886
1887  011666  012601              9$:     MOV     (SP)+,R1        ;;RESTORE R1
1888  011670  012600              10$:    MOV     (SP)+,R0        ;;RESTORE R0
1889  011672  104401  001227              TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
1890  011676  000207                      RTS     PC              ;;RETURN
1891  011700  020040      000     11$:    .ASCIZ  /  /            ;;TWO(2) SPACES
1892          011704                      .EVEN
1893                              .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
1894
1895                              ;;**************************************************************
1896                              ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
1897                              ;*OCTAL (ASCII) NUMBER AND TYPE IT.
1898                              ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
1899                              ;*CALL:
1900                              ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
1901                              ;*      TYPOS                   ;;CALL FOR TYPEOUT
1902                              ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
1903                              ;*      .BYTE   M               ;;M=1 OR 0
1904                              ;*                                      ;;1=TYPE LEADING ZEROS
1905                              ;*                                      ;;0=SUPPRESS LEADING ZEROS
1906                              ;*
1907                              ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
1908                              ;*$TYPOS OR $TYPOC
1909                              ;*CALL:
1910                              ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
1911                              ;*      TYPON                   ;;CALL FOR TYPEOUT
1912                              ;*
1913                              ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
1914                              ;*CALL:
1915                              ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
1916                              ;*      TYPOC                   ;;CALL FOR TYPEOUT
```

```
                                ;;PICKUP THE MODE
                                ;;LOAD ZERO FILL SWITCH
                                ;;NUMBER OF DIGITS TO TYPE
                                ;;ADJUST RETURN ADDRESS
                                ;;SET THE ZERO FILL SWITCH
                                ;;SET FOR SIX(6) DIGITS
                                ;;SET THE ITERATION COUNT
                                ;;SAVE R3
                                ;;SAVE R4
                                ;;SAVE R5
                                ;;GET THE NUMBER OF DIGITS TO TYPE
                                ;;SUBTRACT IT FOR MAX. ALLOWED
                                ;;SAVE IT FOR USE
                                ;;GET THE ZERO FILL SWITCH
                                ;;PICKUP THE INPUT NUMBER
                                ;;CLEAR THE OUTPUT WORD
                                ;;ROTATE MSB INTO "C"
                                ;;GO DO MSB
                                ;;FORM THIS DIGIT
                                ;;GET LSB OF THIS DIGIT
                                ;;TYPE THIS DIGIT?
                                ;;BR IF NO
                                ;;GET RID OF JUNK
                                ;;TEST FOR 0
                                ;;SUPPRESS THIS 0?
                                ;;BR IF YES
                                ;;DON'T SUPPRESS ANYMORE 0'S
                                ;;MAKE THIS DIGIT ASCII
                                ;;MAKE ASCII IF NOT ALREADY
                                ;;SAVE FOR TYPING
                                ;;GO TYPE THIS DIGIT
                                ;;COUNT BY 1
                                ;;BR IF MORE TO DO
                                ;;BR IF DONE
                                ;;INSURE LAST DIGIT ISN'T A BLANK
                                ;;GO DO THE LAST DIGIT
                                ;;RESTORE R5
                                ;;RESTORE R4
                                ;;RESTORE R3
                                ;;SET THE STACK FOR RETURNING
                                ;;RETURN
                                ;;STORAGE FOR ASCII DIGIT
                                ;;TERMINATOR FOR TYPE ROUTINE
                                ;;OCTAL DIGIT COUNTER
                                ;;ZERO FILL SWITCH
                                ;;NUMBER OF DIGITS TO TYPE
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

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

```
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                        ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                        ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                        ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                        ;*REPLACED WITH SPACES.
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                        ;*      TYPDS                   ;;GO TO THE ROUTINE

012132                                  $TYPDS:
012132  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
012134  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
012136  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
012140  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
012142  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
012144  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
012150  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
012154  100004                          BPL     1$              ;;BR IF INPUT IS POS.
012156  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
012160  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
012166  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
012170  012703  012346                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
012174  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
012200  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
012202  016001  012336                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
012206  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
012210  002402                          BLT     4$              ;;BR IF DONE
012212  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
012214  000774                          BR      3$
012216  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
012220  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
012222  001002                          BNE     5$              ;;FALL THROUGH IF 0
012224  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
012226  100407                          BMI     7$              ;;BR IF YES
012230  106316                  5$:     ASLB    (SP)            ;;MSD?
012232  103003                          BCC     6$              ;;BR IF NO
012234  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
012242  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
012246  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
012252  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
012254  005720                          TST     (R0)+           ;;JUST INCREMENTING
012256  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
012262  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
012264  003002                          BGT     8$              ;;GO TO EXIT
012266  010502                          MOV     R5,R2           ;;GET THE LSD
012270  000764                          BR      6$              ;;GO CHANGE TO ASCII
012272  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
012274  100003                          BPL     9$              ;;BR IF NO
012276  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
012304  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
012306  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
012310  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
012312  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
012314  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
012316  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
012320  104401  012346                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
```

```
 2029  012324  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 2030  012332  012616                          MOV     (SP)+,(SP)
 2031  012334  000002                          RTI                     ;;RETURN TO USER
 2032  012336  023420                  $DTBL:  10000.
 2033  012340  001750                          1000.
 2034  012342  000144                          100.
 2035  012344  000012                          10.
 2036  012346  000004                  $DBLK:  .BLKW   4
 2037                                  .SBTTL  TYPE ROUTINE
 2038
 2039                                  ;;**************************************************************
 2040                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 2041                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 2042                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 2043                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 2044                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 2045                                  ;*
 2046                                  ;*CALL:
 2047                                  ;*1) USING A TRAP INSTRUCTION
 2048                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 2049                                  ;*OR
 2050                                  ;*      TYPE
 2051                                  ;*      MESADR
 2052                                  ;*
 2053
 2054  012356  105767  166575  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 2055  012362  100002                  BPL     1$              ;;BR IF YES
 2056  012364  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
 2057  012366  000430                  BR      3$              ;;LEAVE
 2058  012370  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
 2059  012372  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 2060  012376  122767  000001  166646  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 2061  012404  001011                  BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
 2062  012406  132767  000100  166637  BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 2063  012414  001405                  BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
 2064  012416  010067  000004          MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
 2065  012422  004767  000220          JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
 2066  012426  000000          61$:    .WORD   0               ;;MESSAGE ADDRESS
 2067  012430  132767  000040  166615  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
 2068  012436  001003                  BNE     60$             ;;YES,SKIP TYPE OUT
 2069  012440  112046          2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 2070  012442  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 2071  012444  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 2072  012446  012600          60$:    MOV     (SP)+,R0        ;;RESTORE R0
 2073  012450  062716  000002  3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 2074  012454  000002                  RTI                     ;;RETURN
 2075  012456  122716  000011  4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 2076  012462  001430                  BEQ     8$
 2077  012464  122716  000200          CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 2078  012470  001006                  BNE     5$
 2079  012472  005726                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
 2080  012474  104401                  TYPE                    ;;TYPE A CR AND LF
 2081  012476  001227                  $CRLF
 2082  012500  105067  000130          CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 2083  012504  000755                  BR      2$              ;;GET NEXT CHARACTER
 2084  012506  004767  000056  5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
```

```
2085  012512  126726  166440          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
2086  012516  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
2087  012520  016746  166430                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
2088                                                                  ;;AND THE NULL CHAR.
2089  012524  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
2090  012530  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
2091  012532  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
2092  012536  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
2093  012542  000770                          BR      7$              ;;LOOP
2094
2095                                  ;HORIZONTAL TAB PROCESSOR
2096
2097  012544  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
2098  012550  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
2099  012554  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
2100  012562  001372                          BNE     9$              ;;TAB STOP
2101  012564  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
2102  012566  000724                          BR      2$              ;;GET NEXT CHARACTER
2103  012570  105777  166354          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
2104  012574  100375                          BPL     $TYPEC
2105  012576  116677  000002  166346          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
2106  012604  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
2107  012612  001003                          BNE     1$              ;;BRANCH IF NO
2108  012614  105067  000014                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
2109  012620  000406                          BR      $TYPEX          ;;EXIT
2110  012622  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
2111  012630  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
2112  012632  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
2113  012634  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
2114  012636  000207          $TYPEX: RTS     PC
2115
2116                                  .SBTTL  APT COMMUNICATIONS ROUTINE
2117
2118                                  ;;**************************************************************
2119  012640  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
2120  012646  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
2121  012654  000403                          BR      $ATYC
2122  012656  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
2123  012664                          $ATYC:
2124  012664  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
2125  012666  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2126  012670  105767  000206                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
2127  012674  001450                          BEQ     5$              ;;IF NOT:  BR
2128  012676  122767  000001  166346          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
2129  012704  001031                          BNE     3$              ;;IF NOT:  BR
2130  012706  132767  000100  166337          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
2131  012714  001425                          BEQ     3$              ;;IF NOT:  BR
2132  012716  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
2133  012722  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
2134  012730  005767  166276          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
2135  012734  001375                          BNE     1$              ;;IF NOT:  WAIT
2136  012736  010067  166304                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
2137  012742  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
2138  012744  001376                          BNE     2$
2139  012746  166700  166274                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
2140  012752  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
```

```
2141  012754  010067  166270                    MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
2142  012760  012767  000004  166244            MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
2143  012766  000413                            BR      5$
2144  012770  017667  000004  000016  3$:       MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
2145  012776  062766  000002  000004            ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
2146  013004  016746  164766                    MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
2147  013010  004767  177342                    JSR     PC,$TYPE        ;;CALL TYPE MACRO
2148  013014  000000                     4$:    .WORD   0
2149  013016                             5$:
2150  013016  105767  000062            10$:   TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
2151  013022  001416                            BEQ     12$             ;;IF NOT:  BR
2152  013024  005767  166222                    TST     $ENV            ;;RUNNING UNDER APT?
2153  013030  001413                            BEQ     12$             ;;IF NOT:  BR
2154  013032  005767  166174            11$:   TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
2155  013036  001375                            BNE     11$             ;;IF NOT:  WAIT
2156  013040  017667  000004  166166            MOV     @4(SP),$FATAL   ;;GET ERROR #
2157  013046  062766  000002  000004            ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
2158  013054  005267  166152                    INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
2159  013060  105067  000020            12$:   CLRB    $FFLG           ;;CLEAR FATAL FLAG
2160  013064  105067  000013                    CLRB    $LFLG           ;;CLEAR LOG FLAG
2161  013070  105067  000006                    CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
2162  013074  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
2163  013076  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
2164  013100  000207                            RTS     PC              ;;RETURN
2165  013102     000                     $MFLG: .BYTE   0               ;;MESSG. FLAG
2166  013103     000                     $LFLG: .BYTE   0               ;;LOG FLAG
2167  013104     000                     $FFLG: .BYTE   0               ;;FATAL FLAG
2168          013106                             .EVEN
2169          000200                     APTSIZE=200
2170          000001                     APTENV=001
2171          000100                     APTSPOOL=100
2172          000040                     APTCSUF=040
2173                                     .SBTTL  TTY INPUT ROUTINE
2174
2175                                     ;;**********************************************************************
2176                                     .ENABL  LSB
2177
2178                                     ;;**********************************************************************
2179                                     ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
2180                                     ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
2181                                     ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
2182                                     ;*WHEN OPERATING IN TTY FLAG MODE.
2183  013106  022767  000176  166024  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
2184  013114  001074                            BNE     15$             ;;BRANCH IF NO
2185  013116  105777  166022                    TSTB    @$TKS           ;;CHAR THERE?
2186  013122  100071                            BPL     15$             ;;IF NO, DON'T WAIT AROUND
2187  013124  117746  166016                    MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
2188  013130  042716  177600                    BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
2189  013134  022726  000007                    CMP     #7,(SP)+        ;;IS IT A CONTROL G?
2190  013140  001062                            BNE     15$             ;;NO, RETURN TO USER
2191  013142  126727  165766  000001            CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
2192  013150  001456                            BEQ     15$             ;;BRANCH IF YES
2193
2194  013152  104401  013761                    TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
2195  013156  104401  013766            $G^SWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
2196  013162  016746  165010                    MOV     $SWREG,-(SP)    ;;SAVE SWREG FOR TYPEOUT
```

```
2197  013166  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2198  013170  104401  013777                TYPE    .$MNEW          ;;PROMPT FOR NEW SWR
2199  013174  005046                19$:     CLR     -(SP)           ;;CLEAR COUNTER
2200  013176  005046                        CLR     -(SP)           ;;THE NEW SWR
2201  013200  105777  165740        7$:     TSTB    @$TKS           ;;CHAR THERE?
2202  013204  100375                        BPL     7$              ;;IF NOT TRY AGAIN
2203
2204  013206  117746  165734                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
2205  013212  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
2206
2207
2208
2209  013216  021627  000025        9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
2210  013222  001005                        BNE     10$             ;;BRANCH IF NOT
2211  013224  104401  013754                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
2212  013230  062706  000006        20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
2213  013234  000757                        BR      19$             ;;LET'S TRY IT AGAIN
2214
2215
2216  013236  021627  000015        10$:    CMP     (SP),#15        ;;IS IT A <CR>?
2217  013242  001022                        BNE     16$             ;;BRANCH IF NO
2218  013244  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
2219  013250  001403                        BEQ     11$             ;;BRANCH IF YES
2220  013252  016677  000002  165660        MOV     2(SP),@SWR      ;;SAVE NEW SWR
2221  013260  062706  000006        11$:    ADD     #6,SP           ;;CLEAR UP STACK
2222  013264  104401  001227        14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
2223  013270  126727  165641  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
2224  013276  001003                        BNE     15$             ;;BRANCH IF NOT
2225  013300  012777  000100  165636        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
2226  013306  000002                15$:    RTI                     ;;RETURN
2227  013310  004767  177254        16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
2228  013314  021627  000060                CMP     (SP),#60        ;;CHAR < 0?
2229  013320  002420                        BLT     18$             ;;BRANCH IF YES
2230  013322  021627  000067                CMP     (SP),#67        ;;CHAR > 7?
2231  013326  003015                        BGT     18$             ;;BRANCH IF YES
2232  013330  042726  000060                BIC     #60,(SP)+       ;;STRIP-OFF ASCII
2233  013334  005766  000002                TST     2(SP)           ;;IS THIS THE FIRST CHAR
2234  013340  001403                        BEQ     17$             ;;BRANCH IF YES
2235  013342  006316                        ASL     (SP)            ;;NO, SHIFT PRESENT
2236  013344  006316                        ASL     (SP)            ;;   CHAR OVER TO MAKE
2237  013346  006316                        ASL     (SP)            ;;   ROOM FOR NEW ONE.
2238  013350  005266  000002        17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
2239  013354  056616  177776                BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
2240  013360  000707                        BR      7$              ;;GET THE NEXT ONE
2241  013362  104401  001226        18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
2242  013366  000720                        BR      20$             ;;SIMULATE CONTROL-U
2243                                .DSABL  LSB
2244
2245
2246                                ;;*****************************************************************
2247                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
2248                                ;*CALL:
2249                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
2250                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
2251                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
2252                                ;
```

```
2253
2254  013370  011646                    $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
2255  013372  016666  000004  000002            MOV     4(SP),2(SP)     ;;SAVE THE PS
2256  013400  105777  165540            1$:     TSTB    @$TKS           ;;WAIT FOR
2257  013404  100375                            BPL     1$              ;;A CHARACTER
2258  013406  117766  165534  000004            MOVB    @$TKB,4(SP)     ;;READ THE TTY
2259  013414  042766  177600  000004            BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
2260  013422  026627  000004  000023            CMP     4(SP),#23       ;;IS IT A CONTROL-S?
2261  013430  001013                            BNE     3$              ;;BRANCH IF NO
2262  013432  105777  165506            2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
2263  013436  100375                            BPL     2$              ;;LOOP UNTIL ITS THERE
2264  013440  117746  165502                    MOVB    @$TKB,-(SP)     ;;GET CHARACTER
2265  013444  042716  177600                    BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
2266  013450  022627  000021                    CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
2267  013454  001366                            BNE     2$              ;;IF NOT DISCARD IT
2268  013456  000750                            BR      1$              ;;YES, RESUME
2269  013460  026627  000004  000140  3$:      CMP     4(SP),#140      ;;IS IT UPPER CASE?
2270  013466  002407                            BLT     4$              ;;BRANCH IF YES
2271  013470  026627  000004  000175            CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
2272  013476  003003                            BGT     4$              ;;BRANCH IF YES
2273  013500  042766  000040  000004            BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
2274  013506  000002                    4$:     RTI                     ;;GO BACK TO USER
2275                                    ;;**************************************************************
2276                                    ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
2277                                    ;*CALL:
2278                                    ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
2279                                    ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
2280                                    ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
2281
2282  013510  010346                    $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
2283  013512  005046                            CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
2284  013514  012703  013744            1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
2285  013520  022703  013754            2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
2286  013524  101456                            BLOS    4$              ;;BR IF YES
2287  013526  104410                            RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
2288  013530  112613                            MOVB    (SP)+,(R3)      ;;GET CHARACTER
2289  013532  122713  000177            10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
2290  013536  001022                            BNE     5$              ;;BR IF NO
2291  013540  005716                            TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
2292  013542  001007                            BNE     6$              ;;BR IF NO
2293  013544  112767  000134  000170            MOVB    #'\,9$          ;;TYPE A BACK SLASH
2294  013552  104401  013742                    TYPE    ,9$
2295  013556  012716  177777                    MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
2296  013562  005303                    6$:     DEC     R3              ;;BACKUP BY ONE
2297  013564  020327  013744                    CMP     R3,#$TTYIN      ;;STACK EMPTY?
2298  013570  103434                            BLO     4$              ;;BR IF YES
2299  013572  111367  000144                    MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
2300  013576  104401  013742                    TYPE    ,9$             ;;GO TYPE
2301  013602  000746                            BR      2$              ;;GO READ ANOTHER CHAR.
2302  013604  005716                    5$:     TST     (SP)            ;;RUBOUT KEY SET?
2303  013606  001406                            BEQ     7$              ;;BR IF NO
2304  013610  112767  000134  000124            MOVB    #'\,9$          ;;TYPE A BACK SLASH
2305  013616  104401  013742                    TYPE    ,9$
2306  013622  005016                            CLR     (SP)            ;;CLEAR THE RUBOUT KEY
2307  013624  122713  000025            7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
2308  013630  001003                            BNE     8$              ;;BR IF NO
```

```
2309  013632  104401  013754                TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
2310  013636  000726                        BR      1$              ;;GO START OVER
2311  013640  122713  000022        8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
2312  013644  001011                        BNE     3$              ;;BRANCH IF NO
2313  013646  105013                        CLRB    (R3)            ;;CLEAR THE CHARACTER
2314  013650  104401  001227                TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
2315  013654  104401  013744                TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
2316  013660  000717                        BR      2$              ;;GO PICKUP ANOTHER CHACTER
2317  013662  104401  001226        4$:     TYPE    ,$QUES          ;;TYPE A '?'
2318  013666  000712                        BR      1$              ;;CLEAR THE BUFFER AND LOOP
2319  013670  111367  000046        3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
2320  013674  104401  013742                TYPE    ,9$
2321  013700  122723  000015                CMPB    #15,(R3)+       ;;CHECK FOR RETURN
2322  013704  001305                        BNE     2$              ;;LOOP IF NOT RETURN
2323  013706  105063  177777                CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
2324  013712  104401  001230                TYPE    ,$LF            ;;TYPE A LINE FEED
2325  013716  005726                        TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
2326  013720  012603                        MOV     (SP)+,R3        ;;RESTORE R3
2327  013722  011646                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
2328  013724  016666  000004  000002        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
2329  013732  012766  013744  000004        MOV     #$TTYIN,4(SP)
2330  013740  000002                        RTI                     ;;RETURN
2331  013742     000                9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
2332  013743     000                        .BYTE   0               ;;TERMINATOR
2333  013744  000010                $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
2334  013754  052536  005015  000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
2335  013761     136  006507  000012 $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
2336  013766  005015  053523  020122 $MSWR:  .ASCIZ  <15><12>/SWR = /
2337  013774  020075     000
2338  013777     040  047040  053505 $MNEW:  .ASCIZ  /  NEW = /
2339  014004  036440  000040
2340                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
2341
2342                                  ;;******************************************************************
2343                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
2344                                  ;*CHANGE IT TO BINARY.
2345                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
2346                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
2347                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
2348                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
2349                                  ;*CALL:
2350                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
2351                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
2352                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
2353
2354  014010  011646                $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
2355  014012  016666  000004  000002        MOV     4(SP),2(SP)     ;;INPUT NUMBER
2356  014020  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
2357  014022  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2358  014024  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
2359  014026  104411                1$:     RDLIN                   ;;READ AN ASCIZ LINE
2360  014030  012600                        MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
2361  014032  010067  000100                MOV     R0,5$           ;;AND SAVE IT
2362  014036  005001                        CLR     R1              ;;CLEAR DATA WORD
2363  014040  005002                        CLR     R2
2364  014042  112046                2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
```

```
2365  014044  001420                        BEQ     3$              ;;IF ZERO GET OUT
2366  014046  122716  000060                CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
2367  014052  003026                        BGT     4$              ;;IS AN OCTAL DIGIT
2368  014054  122716  000067                CMPB    #'7,(SP)
2369  014060  002423                        BLT     4$
2370  014062  006301                        ASL     R1              ;;*2
2371  014064  006102                        ROL     R2
2372  014066  006301                        ASL     R1              ;;*4
2373  014070  006102                        ROL     R2
2374  014072  006301                        ASL     R1              ;;*8
2375  014074  006102                        ROL     R2
2376  014076  042716  177770                BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
2377  014102  062601                        ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
2378  014104  000756                        BR      2$              ;;LOOP
2379  014106  005726                3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
2380  014110  010166  000012                MOV     R1,12(SP)       ;;SAVE THE RESULT
2381  014114  010267  000026                MOV     R2,$HIOCT
2382  014120  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
2383  014122  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
2384  014124  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
2385  014126  000002                        RTI                     ;;RETURN
2386  014130  005726                4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
2387  014132  105010                        CLRB    (R0)            ;;SET A TERMINATOR
2388  014134  104401                        TYPE                    ;;TYPE UP THRU THE BAD CHAR.
2389  014136  000000                5$:     .WORD   0
2390  014140  104401  001226                TYPE    ,$QUES          ;;"?" "CR" & "LF"
2391  014144  000730                        BR      1$              ;;TRY AGAIN
2392  014146  000000                $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
2393                                .SBTTL  READ A DECIMAL NUMBER FROM THE TTY
2394
2395                                ;;**************************************************************
2396                                ;*THIS ROUTINE WILL READ A DECIMAL (ASCII) NUMBER FROM THE TTY AND
2397                                ;*CHANGE IT TO BINARY. IF TOO MANY CHARACTERS OR ANY ILLEGAL CHARACTERS
2398                                ;*ARE READ A "?" FOLLOWED BY A CARRIAGE RETURN-LINE FEED WILL BE TYPED.
2399                                ;*THE COMPLETE NUMBER MUST BE RETYPED. THE INPUT IS TERMINATED BY THE
2400                                ;*USER TYPING A CARRIAGE RETURN. THE RANGE OF THE INPUT NUMBER IS
2401                                ;*POSITIVE 32767 TO NEGATIVE 32768.
2402                                ;*CALL:
2403                                ;*      RDDEC                   ;;READ A DECIMAL NUMBER
2404                                ;*      RETURN HERE             ;;NUMBER IS ON TOP OF THE STACK
2405                                ;
2406
2407  014150  011646                $RDDEC: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR
2408  014152  016666  000004  000002        MOV     4(SP),2(SP)     ;;THE INPUT NUMBER
2409  014160  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
2410  014162  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2411  014164  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
2412  014166  104411                1$:     RDLIN                   ;;READ AN ASCIZ LINE
2413  014170  012600                        MOV     (SP)+,R0        ;;ADDRESS OF 1ST CHAR.
2414  014172  010067  000120                MOV     R0,6$           ;;SAVE INCASE OF BAD INPUT
2415  014176  005046                        CLR     -(SP)           ;;CLEAR DATA WORD
2416  014200  005002                        CLR     R2              ;;SIGN SET POSITIVE
2417  014202  122710  000055                CMPB    #'-,(R0)        ;;SEE IF A MINUS SIGN WAS TYPED
2418  014206  001001                        BNE     2$              ;;BR IF NO MINUS SIGN
2419  014210  112002                        MOVB    (R0)+,R2        ;;SAVE FOR LATER USE
2420  014212  112001                2$:     MOVB    (R0)+,R1        ;;PICKUP THIS CHARACTER
```

```
2421  014214  001424                          BEQ     3$              ;;GET OUT IF ZERO
2422  014216  122701  000060                  CMPB    #'0,R1          ;;MAKE SURE THIS CHARACTER
2423  014222  003032                          BGT     5$              ;;IS A DIGIT BETWEEN 0 & 9
2424  014224  122701  000071                  CMPB    #'9,R1
2425  014230  002427                          BLT     5$
2426  014232  032716  170000                  BIT     #^C7777,(SP)    ;;DON'T LET NUMBER GET TO BIG
2427  014236  001024                          BNE     5$              ;;BR IF NUMBER WOULD OVERFLOW
2428  014240  006316                          ASL     (SP)            ;;*2
2429  014242  011646                          MOV     (SP),-(SP)      ;;SAVE FOR LATER
2430  014244  006316                          ASL     (SP)            ;;*4
2431  014246  006316                          ASL     (SP)            ;;*8.
2432  014250  062616                          ADD     (SP)+,(SP)      ;;*10.
2433  014252  102416                          BVS     5$              ;;OVERFLOW ISN'T ALLOWED
2434  014254  162701  000060                  SUB     #'0,R1          ;;STRIP AWAY THE ASCII JUNK
2435  014260  060116                          ADD     R1,(SP)         ;;ADD IN THIS DIGIT
2436  014262  102412                          BVS     5$              ;;OVERFLOW ISN'T ALLOWED
2437  014264  000752                          BR      2$              ;;LOOP
2438  014266  005702                  3$:     TST     R2              ;;CHECK IF NUMBER IS NEG
2439  014270  001401                          BEQ     4$              ;;BR IF NO
2440  014272  005416                          NEG     (SP)            ;;YES--NEGATE THE NUMBER
2441  014274  012666  000012          4$:     MOV     (SP)+,12(SP)    ;;SAVE THE RESULT
2442  014300  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
2443  014302  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
2444  014304  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
2445  014306  000002                          RTI                     ;;RETURN
2446
2447  014310  005726                  5$:     TST     (SP)+           ;;CLEAN PARTIAL NUMBER FROM STACK
2448  014312  105010                          CLRB    (R0)            ;;SET A TERMINATOR
2449  014314  104401                          TYPE                    ;;TYPE THE INPUT UP TO BAD CHAR.
2450  014316  000000                  6$:     .WORD   0               ;;POINTER GOES HERE
2451  014320  104401  001226                  TYPE    ,$QUES          ;;"?" "CR" &"LF"
2452  014324  000720                          BR      1$              ;;TRY AGAIN
2453                                  .SBTTL  TRAP DECODER
2454
2455                                  ;;**************************************************************
2456                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
2457                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
2458                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
2459                                  ;*GO TO THAT ROUTINE.
2460
2461  014326  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
2462  014330  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
2463  014334  005740                          TST     -(R0)           ;;BACKUP BY 2
2464  014336  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
2465  014340  006300                          ASL     R0              ;;POSITION FOR INDEXING
2466  014342  016000  014362                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
2467  014346  000200                          RTS     R0              ;;GO TO ROUTINE
2468
2469
2470                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
2471
2472  014350  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
2473  014352  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
2474  014360  000002                          RTI                     ;;RESTORE THE PSW
2475
2476                                  .SBTTL  TRAP TABLE
```

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

                                ;       ROUTINE
                                ;       -------
014362  014350                  $TRPAD: .WORD   $TRAP2
014364  012356                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
014366  011730                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
014370  011704                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
014372  011744                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
014374  012132                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

014376  013156                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

014400  013106                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
014402  013370                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
014404  013510                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
014406  014010                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
014410  014150                          $RDDEC  ;;CALL=RDDEC    TRAP+13(104413) READ A DECIMAL NUMBER FROM TTY
                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;**************************************************************
                                ;POWER DOWN ROUTINE
014412  012737  014556  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
014420  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
014426  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
014430  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
014432  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
014434  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
014436  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
014440  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
014442  017746  164472                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
014446  010667  000110                  MOV     SP,$SAVR6       ;;SAVE SP
014452  012737  014464  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
014460  000000                          HALT
014462  000776                          BR      .-2             ;;HANG UP

                                ;;**************************************************************
                                ;POWER UP ROUTINE
014464  012737  014556  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
014472  016706  000064                  MOV     $SAVR6,SP       ;;GET SP
014476  005067  000060                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
014502  005267  000054          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
014506  001375                          BNE     1$              ;;OF  WORD
014510  012677  164424                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
014514  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
014516  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
014520  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
014522  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
014524  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
014526  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
014530  012737  014412  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
014536  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
014544  104401                          TYPE                    ;REPORT THE POWER FAILURE
014546  014564          $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
```

```
 2533  014550  012716                        MOV     (PC)+,(SP)      ;;RESTART AT CKRST1
 2534  014552  014574              $PWRAD:   .WORD   CKRST1          ;;RESTART ADDRESS
 2535  014554  000002                        RTI
 2536  014556  000000              $ILLUP:   HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
 2537  014560  000776                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
 2538  014562  000000              $SAVR6:   0                       ;;PUT THE SP HERE
 2539  014564  005015  047520  042527  $POWER:  .ASCIZ  <15><12>"POWER"
 2540  014572  000122
 2541                                        .EVEN
```

```
2542                                    ;;*****************************************************************
2543                                    ;COMMON DH11 SERVICE ROUTINES
2544                                    ;;*****************************************************************
2545
2546                                    ;THESE ROUTINES DETERMINE RESTART ADDRESS AFTER SYSTEM ERROR
2547                                    ;(BUS ERROR,RSVD INSTR ERROR, OR POWER FAIL)
2548
2549  014574  005767  005546       CKRST1: TST     DPFLG           ;IN PATTERNS TEST ?
2550  014600  001005                       BNE     1$              ;BR IF YES
2551  014602  005767  005552               TST     RETFLG          ;IN ECHO TEST ?
2552  014606  001004                       BNE     2$              ;BR IF YES
2553  014610  000167  165626               JMP     RSTRTA          ;GO RESTART RELIABILITY TESTS
2554  014614  000167  171266       1$:     JMP     EXPAT           ;GO TO PATTERNS TESTS
2555  014620  000167  170062       2$:     JMP     ECHO            ;GO TO ECHO TESTS
2556
2557  014624  005767  005516       CKRST2: TST     DPFLG           ;IN PATTERNS TEST ?
2558  014630  001005                       BNE     1$              ;BR IF YES
2559  014632  005767  005522               TST     RETFLG          ;IN ECHO TEST ?
2560  014636  001004                       BNE     2$              ;BR IF YES
2561  014640  000167  165566               JMP     REST1           ;GO RESTART RELIABILITY TESTS
2562  014644  000167  171236       1$:     JMP     EXPAT           ;GO TO PATTERNS TESTS
2563  014650  000167  170032       2$:     JMP     ECHO            ;GO TO ECHO TESTS
2564
2565
2566                                    ;THIS ROUTINE IS CALLED TO SET UP THE DH11 PARAMETERS PRIOR TO TEST
2567
2568  014654  012711  004000       DHSET1: MOV     #BIT11,(R1)     ;CLEAR THE DH11 UNDER TEST
2569  014660  004767  004556               JSR     PC,CHPS2        ;GO LOCK OUT INTRS
2570  014664  012711  030100               MOV     #30100,(R1)     ;ENABLE INTERRUPTS ON XMIT DONE
2571                                                                ;NON-EX MEM, DATA AVAIL, OR SILO OVFLW
2572  014670  156711  005256               BISB    LINE,(R1)       ;SELECT THE LINE NO.
2573  014674  005067  004710               CLR     RDONE           ;CLEAR SOFTWARE DONE FLAG
2574  014700  012767  030516  013604       MOV     #RBUF,RBFPTR    ;SET UP RCVR BUFFER POINTER
2575  014706  012767  030516  004676       MOV     #RBUF,RBFEND    ;MARK END OF THIS BUFFER
2576  014714  016705  004662               MOV     CHRCNT,R5       ;GET CHAR COUNT
2577  014720  005405                       NEG     R5              ;MAKE IT POSITIVE
2578  014722  060567  004664               ADD     R5,RBFEND
2579  014726  016761  004644  000004       MOV     CURLPR,LPR(R1)  ;LOAD THE LPR REG
2580  014734  016761  004642  000010       MOV     CHRCNT,BCR(R1)  ;LOAD THE BYTE COUNT REG
2581  014742  012761  032776  000006       MOV     #TBUF,CAR(R1)   ;LOAD CURRENT ADDRESS REG
2582  014750  000207                       RTS     PC              ;RETURN
2583
2584                                    ;THIS ROUTINE IS CALLED TO SELECT A NEW LINE NO. BASED ON THE
2585                                    ;VALUE OF THE LINE SELECTION PARAMETER
2586
2587                                            ;CALLING SEQUENCE:
2588
2589                                            ;JSR     PC,SELINE      ;CALL THE ROUTINE
2590                                            ;BR      1$             ;EXIT BRANCH-ROUTINE MOVES THE RETURN
2591                                                                    ;PC AROUND THIS BR IF MORE LINES ARE
2592                                                                    ;YET TO BE TESTED
2593
2594  014752  105767  005175       SELINE: TSTB    LINE+1          ;FIRST TIME THROUGH FOR ANY TEST ?
2595  014756  001010                       BNE     1$              ;BR IF NOT
2596  014760  105167  005167               COMB    LINE+1          ;SET ENTRY FLAG
2597  014764  012767  000001  004542       MOV     #1,LINMSK       ;INIT SELECT TEST MASK TO TEST LINE 00
```

```
2598  014772  105067  005154                    CLRB    LINE            ;START WITH LINE #00
2599  014776  000405                            BR      2$              ;GO TEST FOR LINE #00
2600  015000  105267  005146          1$:       INCB    LINE            ;GENERATE NEW LINE NO.
2601  015004  006367  004524                    ASL     LINMSK          ;SHIFT SELECT MASK TO TEST NXT LINE
2602  015010  001407                            BEQ     3$              ;RETURN TO EXIT BRANCH - ALL LINES DONE
2603  015012  036767  004516  004512  2$:       BIT     LINMSK,LINSEL   ;IS THE LINE SELECTED FOR TEST ??
2604  015020  001767                            BEQ     1$              ;BR IF NOT
2605  015022  062716  000002                    ADD     #2,(SP)         ;MOVE RETURN PC AROUND EXIT BRANCH
2606  015026  000402                            BR      4$              ;RETURN TO TEST SELECTED LINE
2607  015030  005067  005116          3$:       CLR     LINE            ;INIT ENTRY FLAG AND LINE NO. TO 000
2608  015034  142777  000017  004460  4$:       BICB    #17,@DHADR      ;INIT LINE SELECT BITS IN "SCR"
2609  015042  000207                            RTS     PC              ;RETURN TO CALLING TEST
2610
2611                                  ;THIS ROUTINE IS CALLED TO CONVERT EITHER THE "DH" NUMBER OR THE
2612                                  ;"LINE" NUMBER TO TWO ASCII CHARACTERS AND MOVE THEM INTO A
2613                                  ;PARTICULAR MESSAGE BUFFER FOR ERROR REPORTING
2614
2615                                          ;CALLING SEQUENCE
2616
2617                                          ;JSR     R5,SUNUM       ;CALL TO THIS ROUTINE
2618                                          ;ADDR1                  ;ADDRESS OF THE NUMBER TO BE CONVERTED
2619                                          ;ADDR2                  ;ADDRESS OF THE MSG BUFFER SLOT
2620
2621  015044                          SUNUM:
2622  015044  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
2623  015046  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2624  015050  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
2625  015052  012500                            MOV     (R5)+,R0        ;GET ADDRESS OF NUMBER
2626  015054  012501                            MOV     (R5)+,R1        ;GET MSG BUFFER ADDR
2627  015056  111000                            MOVB    (R0),R0         ;GET NO. TO BE CONVERTED
2628  015060  010002                            MOV     R0,R2           ;SAVE IT IN R2
2629  015062  006202                            ASR     R2              ;SHIFT MSD TO LSD POSITION
2630  015064  006202                            ASR     R2
2631  015066  006202                            ASR     R2
2632  015070  042702  177770                    BIC     #177770,R2      ;CLR JUNK BITS
2633  015074  062702  000060                    ADD     #60,R2          ;MAKE IT ASCII
2634  015100  110221                            MOVB    R2,(R1)+        ;PUT IT IN MSG BUFFER
2635  015102  042700  177770                    BIC     #177770,R0      ;CLR JUNK FROM LSD
2636  015106  062700  000060                    ADD     #60,R0          ;MAKE IT ASCII
2637  015112  110011                            MOVB    R0,(R1)         ;PUT LSD IN THE BUFFER
2638  015114  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
2639  015116  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
2640  015120  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
2641  015122  000205                            RTS     R5              ;RETURN TO CALLER
2642
2643                                  ;THIS ROUTINE IS CALLED TO SET UP THE ERROR INFORMATION IN THE
2644                                  ;MESSAGE BUFFERS
2645
2646  015124  010067  164032          SUER2:    MOV     R0,$REG0        ;STORE THE REGS IN CORE
2647  015130  010167  164030          SUER1:    MOV     R1,$REG1
2648  015134  010267  164026                    MOV     R2,$REG2
2649  015140  010367  164024                    MOV     R3,$REG3
2650  015144  010467  164022                    MOV     R4,$REG4
2651  015150  000207                            RTS     PC              ;RETURN TO REPORT ERROR
2652
2653                                  ;THIS ROUTINE AUTOSIZES THE SYSTEM TO DETERMINE THE ADDRESSES AND
```

```
                                        ;VECTORS OF THE DH11'S AND DM11-BB'S.
                                        
015152  010046                  AUTOSZ: MOV     R0,-(SP)
015154  005003                          CLR     R3
015156  012702  021756                  MOV     #DHADRS,R2
015162  005022                  25$:    CLR     (R2)+           ;CLEAR DH TABLES.
015164  005203                          INC     R3
015166  020327  000102                  CMP     R3,#102         ;HAVE WE CLEARED ALL ENTRIES?
015172  001373                          BNE     25$             ;BRANCH IF NOT.
015174  013746  000004                  MOV     @#4,-(SP)       ;SAVE TRAP VECTOR.
015200  012737  015306  000004          MOV     #4$,@#4         ;SETUP FOR NON-EXISTENT MEMORY TRAP.
015206  012703  022050                  MOV     #DMADRS,R3      ;SETUP DM ADDRESS TABLE POINTER.
015212  012702  021756                  MOV     #DHADRS,R2      ;SET UP DH ADDRESS TABLE POINTER.
                                        
015216  012701  160020                  MOV     #160020,R1      ;R1=FIRST ADDRESS TO BE TESTED.
                                        
015222  005711                  1$:     TST     (R1)            ;SEE IF ADDRESS IN R1 RESPONDS.
015224  005761  000016                  TST     16(R1)          ;CHECK TO SEE IF DEVICE IS MODULO 20.
015230  052711  004000                  BIS     #4000,(R1)      ;IF IT IS, CONTINUE
                                                                ;AND CHECK TO SEE
015234  052711  001000                  BIS     #1000,(R1)      ;IF THIS ADDRESS CONTAINS
015240  052711  002000                  BIS     #2000,(R1)      ;A DH-11.
015244  032711  003000                  BIT     #3000,(R1)      ;CHECK TO INSURE THESE BITS SET.
015250  001410                          BEQ     3$              ;IF NOT, BRANCH.
                                                                ;SET THE MAINTENANCE BIT, THE NON-
015252  052711  000400                  BIS     #400,(R1)       ;EXISTENT MEMORY BIT AND THE CLEAR
                                                                ;NON-EXISTENT MEMORY INTERRUPT BIT.
015256  032711  002400                  BIT     #2400,(R1)      ;IS THIS A DH-11? (BITS 8 AND 10 SHOULD
                                                                ;CLEAR IF THIS IS A DH11.)
                                        
015262  001003                          BNE     3$              ;IF NOT, CHECK TO SEE IF THIS IS A DM11-BB.
015264  042711  001000                  BIC     #1000,(R1)      ;CLEAR MAINTENANCE BIT.
015270  010122                          MOV     R1,(R2)+        ;SAVE THE ADDRESS IN THE DH ADR TABLE.
                                        
                                        
015272  020127  163760          3$:     CMP     R1,#163760      ;HAVE WE REACHED THE TOP OF THE FLOATING ADDRESSES.
015276  001406                          BEQ     5$              ;IF YES, GET OUT.
015300  062701  000020                  ADD     #20,R1          ;IF NOT, UPDATE ADDRESS AND
015304  000746                          BR      1$              ;GO CHECK IT.
                                        
015306  012716  015272          4$:     MOV     #3$,(SP)        ;IF DH ADDRESS DOES NOT RESPOND, GO TO 3$.
015312  000002                          RTI
                                        
                                        ;TEST FOR DM11 BB ADDRESS
                                        
015314  012737  015346  000004  5$:     MOV     #6$,@#4         ;SETUP FOR NON-EXISTENT MEMORY TRAP.
015322  012701  170500                  MOV     #170500,R1      ;R1=FIRST ADDRESS TO BE TESTED.
015326  005711                  21$:    TST     (R1)            ;SEE IF ADDRESS RESPONDS.
015330  010123                          MOV     R1,(R3)+        ;IF IT DOES, THIS IS A DM11-BB,
                                                                ;SO SAVE THE ADDRESS.
015332  020127  170670          23$:    CMP     R1,#170670      ;HAVE WE REACHED THE TOP OF THE DM11 ADDRESSES?
015336  001406                          BEQ     22$             ;IF YES, GET OUT.
015340  062701  000010                  ADD     #10,R1          ;IF NOT, UPDATE ADDRESS AND
015344  000770                          BR      21$             ;GO CHECK IT.
                                        
015346  012716  015332          6$:     MOV     #23$,(SP)       ;IF DM ADDRESS DOES NOT RESPOND, GO TO 23$.
```

```
2710  015352  000002                          RTI
2711
2712  015354  012637  000004          22$:    MOV     (SP)+,@#4       ;RESTORE TRAP VECTOR.
2713  015360  162702  021756                  SUB     #DHADRS,R2      ;HAVE WE FOUND ANY DH11'S AT ALL?
2714  015364  001003                          BNE     7$              ;IF YES, BRANCH
2715  015366  104401  025733                  TYPE    ,MSG1           ;NO DH11'S WERE FOUND,
2716  015372  000000                          HALT
2717
2718  015374  006202                  7$:     ASR     R2              ;R2 NOW CONTAINS THE NUMBER
2719  015376  005000                          CLR     R0              ;OF DH'S FOUND.
2720  015400  006100                  8$:     ROL     R0              ;FILL R0 WITH 1'S
2721  015402  005200                          INC     R0              ;CORRESPONDING TO
2722  015404  005302                          DEC     R2              ;THE NUMBER OF DH'S
2723  015406  005702                          TST     R2              ;FOUND.
2724  015410  001373                          BNE     8$
2725  015412  010067  004526                  MOV     R0,$DHSEL       ;$DHSEL CONTAINS THE DH SELECTION PARAMETER.
2726                                                                  ;IE. ALL DH'S FOUND WILL BE TESTED.
2727
2728                          ;FIND DH VECTOR:
2729  015416  012702  021756                  MOV     #DHADRS,R2      ;SETUP POINTER TO BEGINNING OF DH
2730  015422  012705  022014                  MOV     #DHVEC,R5       ;ADDRESS TABLE AND VECTOR TABLE.
2731  015426  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;SET IOT TRAP PRIORITY TO 7.
2732  015434  012737  015544  000020          MOV     #12$,@#IOTVEC   ;SETUP IOT TRAP VECTOR.
2733  015442  012703  000300                  MOV     #300,R3         ;START OF FLOATING VECTORS
2734  015446  012704  000302                  MOV     #302,R4         ;PC OF IOT INSTR.
2735
2736  015452  010423                  9$:     MOV     R4,(R3)+        ;FILL VECTOR AREA WITH ADDRESS
2737                                                                  ;OF NEXT INSTR (.+2)
2738  015454  012724  000004                  MOV     #4,(R4)+        ;NEXT INSTRUCTION IS AN IOT TRAP.
2739  015460  022324                          CMP     (R3)+,(R4)+     ;UPDATE R3+R4.
2740  015462  020427  001000                  CMP     R4,#1000        ;HAVE WE REACHED TO TOP OF THE
2741                                                                  ;VECTOR SPACE?
2742  015466  101771                          BLOS    9$              ;IF NOT, REPEAT PROCESS.
2743
2744  015470  005712                  10$:    TST     (R2)            ;HAVE WE CHECK ALL DH'S?
2745  015472  001441                          BEQ     13$             ;IF YES, GET OUT + CHECK FOR DM11 BB'S VECTORS.
2746
2747  015474  005067  162276                  CLR     PS              ;ZERO CPU PRIORITY.
2748  015500  052772  001000  000000          BIS     #1000,@(R2)     ;SET MAINTENANCE BIT
2749  015506  052772  000300  000000          BIS     #300,@(R2)      ;ATTEMPT TO CAUSE RECEIVER
2750                                                                  ;INTERRUPT.
2751  015514  005000                          CLR     R0
2752
2753  015516  005200                  11$:    INC     R0              ;WAIT...
2754  015520  001376                          BNE     11$
2755  015522  104401  025762                  TYPE    ,MSG2           ;ERROR MSG-NO DH RECEIVER INTERRUPT OCCURRED.
2756  015526  052772  004000  000000          BIS     #4000,@(R2)     ;DO A MASTER CLEAR
2757  015534  042772  001000  000000          BIC     #1000,@(R2)     ;CLEAR MAINTENANCE BIT
2758  015542  000752                          BR      10$
2759
2760  015544  011601                  12$:    MOV     (SP),R1
2761  015546  042701  000007                  BIC     #7,R1           ;CLEAR GARBAGE.
2762  015552  010125                          MOV     R1,(R5)+        ;SAVE VECTOR ADDRESS.
2763  015554  022626                          CMP     (SP)+,(SP)+     ;POP STACK
2764  015556  012716  015470                  MOV     #10$,(SP)       ;SETUP FOR RETURN.
2765  015562  052772  004000  000000          BIS     #4000,@(R2)     ;DO A MASTER CLEAR
```

```
2766  015570  042732  001000                    BIC     #1000,@(R2)+    ;CLEAR MAINTENANCE BIT.
2767  015574  000002                            RTI
2768
2769                                    ;FIND DM11 BB VECTORS:
2770
2771  015576  012702  022050           13$:     MOV     #DMADRS,R2      ;SET POINTERS TO BEGINNING OF
2772  015602  012705  022106                    MOV     #DMVEC,R5       ;ADR TABLE & VECTOR TABLE.
2773  015606  012737  015670  000020            MOV     #16$,@#IOTVEC   ;SET IOT TRAP VECTOR.
2774
2775  015614  005712                   14$:     TST     (R2)            ;HAVE WE CHECKED ALL DM'S?
2776  015616  001441                            BEQ     17$             ;IF YES, GET OUT.
2777  015620  005067  162152                    CLR     PS              ;ZERO CPU PRIORITY
2778  015624  052772  001000  000000            BIS     #1000,@(R2)     ;SET MAINTENANCE BIT.
2779  015632  052772  000300  000000            BIS     #300,@(R2)      ;ATTEMPT TO CAUSE INTERRLPT.
2780  015640  005000                            CLR     R0
2781
2782  015642  005200                   15$:     INC     R0              ;WAIT....
2783  015644  001376                            BNE     15$
2784  015646  104401  026026                    TYPE    ,MSG3           ;ERROR MSG - NO DM11-BB INTERRLPT OCCURRED.
2785  015652  052772  004000  000000            BIS     #4000,@(R2)     ;CLEAR BITS PREVIOUSLY SET.
2786  015660  042772  001000  000000            BIC     #1000,@(R2)     ;CLEAR MAINTENANCE BIT.
2787  015666  000752                            BR      14$
2788
2789  015670  011601                   16$:     MOV     (SP),R1
2790  015672  162701  000004                    SUB     #4,R1           ;CALCULATE VECTOR ADDRESS.
2791  015676  010125                            MOV     R1,(R5)+        ;SAVE VECTOR ADDRESS.
2792  015700  022626                            CMP     (SP)+,(SP)+     ;POP STACK.
2793  015702  012716  015614                    MOV     #14$,(SP)       ;SETUP FOR RETURN.
2794  015706  052772  004000  000000            BIS     #4000,@(R2)     ;CLEAR BITS PREVIOUSLY SET.
2795  015714  042732  001000                    BIC     #1000,@(R2)+    ;CLEAR MAINTENANCE BIT AND
2796                                                                    ;POINT TO NEXT DM11-BB ADDRESS.
2797  015720  000002                            RTI
2798
2799  015722  012737  011112  000020   17$:     MOV     #$SCOPE,@#IOTVEC  ;RESTORE IOT VECTOR FOR SCOPE ROUTINE.
2800  015730  012600                            MOV     (SP)+,R0        ;RESTORE R0.
2801  015732  012703  000300                    MOV     #300,R3         ;START OF FLOATING VECTORS.
2802  015736  012704  000302                    MOV     #302,R4
2803
2804  015742  010423                   18$:     MOV     R4,(R3)+        ;FILL VECTOR AREA WITH ADDRESS OF NEXT
2805                                                                    ;INSTRUCTION (.+2).
2806  015744  012724  000000                    MOV     #0,(R4)+        ;NEXT INSTRUCTION IS A HALT.
2807  015750  022324                            CMP     (R3)+,(R4)+     ;UPDATE R3 & R4.
2808  015752  020427  001000                    CMP     R4,#1000        ;ARE WE DONE?
2809  015756  101771                            BLOS    18$             ;IF NOT, REPEAT UNTIL ADDRESSES
2810                                                                    ;377 TO 777 ARE DONE.
2811  015760  013701  022014                    MOV     @#DHVEC,R1      ;LET R1 POINT TO 1ST DH VECTOR ADDRESS.
2812  015764  005737  022016                    TST     @#DHVEC+2       ;IS THERE MORE THAN ONE ENTRY?
2813  015770  001403                            BEQ     26$             ;BRANCH IF NO.
2814  015772  163701  022016                    SUB     @#DHVEC+2,R1    ;DETERMINE NUMBER OF ADDRESSES
2815                                                                    ;BETWEEN DH VECTORS (10(8) OR 20(8)).
2816  015776  005401                            NEG     R1              ;MAKE IT POSITIVE.
2817  016000  010167  004136           26$:     MOV     R1,ADRVEC       ;SAVE THAT NUMBER.
2818  016004  032777  000002  163126            BIT     #BIT1,@SWR      ;SHOULD DEVICE MAP BE TYPED OUT?
2819  016012  001441                            BEQ     20$             ;IF NOT, RETURN.
2820  016014  104401                            TYPE                    ;TYPEOUT MAP OF DH & DM11-BB'S
2821  016016  026073                            DEVMAP                  ;FOUND.
```

```
 2822  016020  012701  021756                MOV     #DHADRS,R1      ;R1=BEGINNING OF DH ADDRESS TABLE.
 2823  016024  012702  022014                MOV     #DHVEC,R2       ;R2=BEGINNING OF DH VECTOR TABLE.
 2824  016030  012703  022050                MOV     #DMADRS,R3      ;R3=BEGINNING OF DM11-BB ADDRESS TABLE.
 2825  016034  012704  022106                MOV     #DMVEC,R4       ;R4=BEGINNING OF DM11-BB VECTOR TABLE.
 2826  016040  005005                        CLR     R5              ;CLEAR TABLE LINE COUNTER
 2827
 2828  016042  012146              19$:      MOV     (R1)+,-(SP)     ;MOVE DATA TO BE TYPED
 2829  016044  104403                        TYPOS                   ;TYPE DATA
 2830  016046     006                        .BYTE   6
 2831  016047     001                        .BYTE   1
 2832  016050  012246                        MOV     (R2)+,-(SP)     ;MOVE DATA TO BE TYPED
 2833  016052  104403                        TYPOS                   ;TYPE DATA
 2834  016054     005                        .BYTE   5
 2835  016055     000                        .BYTE   0
 2836  016056  104401  026067                TYPE    ,SPACE
 2837  016062  012346                        MOV     (R3)+,-(SP)     ;MOVE DATA TO BE TYPED.
 2838  016064  104403                        TYPOS                   ;TYPE DATA.
 2839  016066     006                        .BYTE   6
 2840  016067     001                        .BYTE   1
 2841  016070  104401  026067                TYPE    ,SPACE
 2842  016074  012446                        MOV     (R4)+,-(SP)     ;MOVE DATA TO BE TYPED.
 2843  016076  104403                        TYPOS                   ;TYPE DATA.
 2844  016100     005                        .BYTE   5
 2845  016101     000                        .BYTE   0
 2846  016102  104401                        TYPE                    ;TYPE A CARRIAGE RETURN & LINE FEED.
 2847  016104  001227                        $CRLF
 2848  016106  005711                        TST     (R1)            ;HAVE WE TYPED ALL DH ENTRIES?
 2849  016110  001354                        BNE     19$             ;IF NOT, DO IT AGAIN.
 2850  016112  104401  001227                TYPE    ,$CRLF
 2851  016116  000207              20$:      RTS     PC              ;IF YES, GO BACK TO MAIN PROGRAM.
 2852
 2853                                        ;THIS ROUTINE IS USED TO ACCEPT INPUT PARAMETERS FROM THE CONSOLE
 2854                                        ;TELETYPE
 2855
 2856  016120  104401              INPARA:   TYPE
 2857  016122  025440                        VCWC                    ;"ASK FOR NO. ADDRESSES BETWEEN VECTORS"
 2858  016124  104412                        RDOCT                   ;READ OCTAL NO. FM TTY
 2859  016126  012600                        MOV     (SP)+,R0        ;GET THE NO. HE TYPED
 2860  016130  001407                        BEQ     3$              ;BR IF HE TYPED <CR>
 2861  016132  022700  000010                CMP     #10,R0          ;10(8) ADDRESSES BETWEEN VECTORS ?
 2862  016136  001406                        BEQ     4$              ;BR IF YES
 2863  016140  022700  000020                CMP     #20,R0          ;20(8) ADDRESSES BETWEEN VECTORS ??
 2864  016144  001403                        BEQ     4$              ;BR IF YES
 2865  016146  000764                        BR      INPARA          ;ASK ALL OVER AGAIN
 2866  016150  012700  000020              3$:       MOV     #20,R0          ;SET UP CONSTANT FOR 20(8) ADDRESSES
 2867  016154  005067  004166              4$:       CLR     DPFLG           ;CLEAR PATTERNS TESTS FLAG
 2868  016160  005067  004174                        CLR     RETFLG          ;INIT ECHO TEST RETURN FLAG
 2869  016164  000207                        RTS     PC              ;RETURN TO CALLER
 2870
 2871  016166  012767  177777  003520      INPARX:   MOV     #-1,VCFLG       ;SET START AT 200 FLAG
 2872  016174  000167  163444                        JMP     BEGINA          ;GO START UP
 2873  016200  012700  177777              INPARC:   MOV     #-1,R0          ;SET FLAG IN R0
 2874  016204  005067  003504                        CLR     VCFLG           ;INIT VECTOR SET UP FLAG
 2875  016210  005067  004132                        CLR     DPFLG           ;CLEAR PATTERNS TESTS FLAG
 2876  016214  005067  004140                        CLR     RETFLG          ;INIT ECHO TEST RETURN FLAG
 2877  016220  000167  163420                        JMP     BEGINA          ;GO ASK FOR SELECT PARAMETER
```

```
2878
2879  016224  013701  021756          INPAR:  MOV     @#DHADRS,R1     ;MOVE ADDRESS OF FIRST DH INTO R1.
2880  016230  032777  000001  162702          BIT     #BIT0,@SWR      ;ARE PARAMETERS TO BE INPUT MANUALLY?
2881  016236  001405                          BEQ     2$              ;BRANCH IF NOT.
2882  016240  104401                  1$:     TYPE                    ;ASK FOR DEVICE ADDRESS
2883  016242  024747                          INMSG1
2884  016244  104412                          RDOCT                   ;READ IN WHAT IS TYPED
2885  016246  012601                          MOV     (SP)+,R1        ;GET THE NO. HE TYPED
2886  016250  001403                          BEQ     INPAR1          ;BR IF DEFAULT
2887  016252  004767  000176          2$:     JSR     PC,CHKADR       ;GO CHECK VALIDITY OF THE ADDR
2888  016256  000770                          BR      1$              ;ERROR BRANCH
2889
2890  016260  013701  022014          INPAR1: MOV     @#DHVEC,R1      ;MOVE FIRST DH VECTOR INTO R1.
2891  016264  032777  000001  162646          BIT     #BIT0,@SWR      ;ARE PARAMETERS TO BE INPUT MANUALLY?
2892  016272  001405                          BEQ     2$              ;BRANCH IF NOT.
2893  016274  104401                  1$:     TYPE                    ;ASK FOR VECTOR ADDRESS
2894  016276  025013                          INMSG2
2895  016300  104412                          RDOCT                   ;READ IN WHAT HE TYPES
2896  016302  012601                          MOV     (SP)+,R1        ;GET THE ADDRESS
2897  016304  001403                          BEQ     INPAR3          ;BR IF DEFAULT
2898  016306  004767  000306          2$:     JSR     PC,CHKVCT       ;GO CHECK VALIDITY OF VECTOR
2899  016312  000770                          BR      1$              ;ERROR BRANCH
2900
2901  016314  005767  004040          INPAR3: TST     RETFLG          ;LINE ECHO TESTS ?
2902  016320  001402                          BEQ     1$              ;BR IF NOT
2903  016322  000167  166426                  JMP     ECHO2           ;RETURN TO LINE ECHO TESTS
2904  016326  005767  004014          1$:     TST     DPFLG           ;DATA PATTERNS TESTS ACTIVE ??
2905  016332  001402                          BEQ     2$              ;BR IF NOT
2906  016334  000167  167614                  JMP     EXPAT2          ;GO BACK TO PATTERNS TESTS
2907  016340  013701  022144          2$:     MOV     @#$DHSEL,R1     ;MOVE DEVICE SELECTION PARAMETER INTO R1.
2908  016344  005700                          TST     R0              ;START AT 210?
2909  016346  100404                          BMI     4$              ;BRANCH IF YES.
2910  016350  032777  000001  162562          BIT     #BIT0,@SWR      ;IS PARAMETER TO BE INPUT MANUALLY?
2911  016356  001405                          BEQ     3$              ;BRANCH IF NOT.
2912  016360  104401                  4$:     TYPE                    ;ASK FOR DEVICE SELECTION PARAMETER
2913  016362  025062                          INMSG3
2914  016364  104412                          RDOCT                   ;READ IN WHAT HE TYPES
2915  016366  012601                          MOV     (SP)+,R1        ;GET THE SELECT PARAMETER
2916  016370  001402                          BEQ     INPAR4          ;BR IF  DEFAULT
2917  016372  010167  003132          3$:     MOV     R1,DHSEL        ;SET UP DH11 SELECTION PARAMETER
2918  016376  012767  177777  003126  INPAR4: MOV     #-1,LINSEL      ;INIT FOR ALL 16. LINES
2919  016404  032777  000001  162526          BIT     #BIT0,@SWR      ;IS LINE SELECT PARAMETER TO BE INPUT MANUALLY?
2920  016412  001407                          BEQ     1$              ;BRANCH IF NO.
2921  016414  104401                          TYPE                    ;ASK FOR LINE SELECT PARAMETER
2922  016416  025376                          INMSGA
2923  016420  104412                          RDOCT                   ;READ WHAT HE TYPES
2924  016422  012601                          MOV     (SP)+,R1        ;GET IT OFF STACK
2925  016424  001402                          BEQ     1$              ;BR IF DEFAULT
2926  016426  010167  003100                  MOV     R1,LINSEL       ;SET LINE SELECT PARAMETER
2927  016432  032777  000400  162500  1$:     BIT     #BIT8,@SWR      ;HALT AFTER SET UP ??
2928  016440  001403                          BEQ     EXPAR           ;BR IF  NOT
2929  016442  104401                          TYPE                    ;TYPE CONTINUE MESSAGE PRIOR TO HALTING
2930  016444  025226                          INMSG7
2931  016446  000000                          HALT                    ;DEPRESS CONTINUE TO RESUME TESTING
2932  016450  000167  163662          EXPAR:  JMP     START2          ;GO START UP THE PROGRAM
2933
```

```
2934
2935  016454  020127  160020          CHKADR: CMP     R1,#160020      ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
2936  016460  002001                          BGE     1$              ;BR IF YES
2937  016462  000453                          BR      4$              ;BR IF NOT
2938  016464  020127  160420          1$:     CMP     R1,#160420      ;IS IT BELOW THE HIGH LIMIT?
2939  016470  002401                          BLT     2$              ;BR IF YES
2940  016472  000447                          BR      4$              ;BR IF NOT
2941  016474  032701  000017          2$:     BIT     #17,R1          ;CORRECT BOUNDARY ?
2942  016500  001044                          BNE     4$              ;BR IF NOT
2943  016502  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC AROUND ERROR BRANCH
2944  016506  005767  003646                  TST     RETFLG          ;ARE WE IN ECHO TESTS ?
2945  016512  001403                          BEQ     21$             ;BR IF NOT
2946  016514  010167  003002                  MOV     R1,DHADR        ;SET UP DH11 DEVICE ADDRESS
2947  016520  000436                          BR      5$              ;CONTINUE
2948  016522  005767  003620          21$:    TST     DPFLG           ;PATTERNS TESTS ACTIVE ??
2949  016526  001403                          BEQ     22$             ;BR IF NOT
2950  016530  010167  002766                  MOV     R1,DHADR        ;SET UP DEVICE ADDRESS
2951  016534  000430                          BR      5$              ;CONTINUE
2952  016536  012702  021614          22$:    MOV     #DHADTB,R2      ;POINT TO BEGIN OF ADDR TABLE
2953  016542  032777  000001  162370          BIT     #BIT0,@$SWR     ;ARE WE AUTOSIZING?
2954  016550  001011                          BNE     3$              ;BRANCH IF NOT.
2955  016552  012703  021756                  MOV     #DHADRS,R3      ;POINT TO BEGINNING OF AUTOSIZER
2956                                                                  ;DH ADDRESS TABLE.
2957  016556  016704  003362                  MOV     $DHSEL,R4
2958  016562  012322                  6$:     MOV     (R3)+,(R2)+     ;MOVE CONTENTS OF AUTOSIZER DH TABLE
2959                                                                  ;TO THE TABLE USED BY PROGRAM.
2960  016564  006204                          ASR     R4
2961  016566  005704                          TST     R4              ;HAVE WE MOVED ALL TABLE ENTRIES?
2962  016570  001374                          BNE     6$              ;BRANCH IF NOT--ONE MORE TIME.
2963  016572  000411                          BR      5$              ;RETURN TO INPUT ROUTINES.
2964  016574  010122                  3$:     MOV     R1,(R2)+        ;SET UP A TABLE ENTRY
2965  016576  062701  000020                  ADD     #20,R1          ;GENERATE NEXT DH11 ADDR
2966  016602  022702  021654                  CMP     #DHADTB+40,R2   ;END OF TABLE ?
2967  016606  001372                          BNE     3$              ;BR IF NOT
2968  016610  000402                          BR      5$              ;RETURN TO INPUT ROUTINES
2969  016612  104401                  4$:     TYPE                    ;TELL HIM HE GOOFED
2970  016614  025133                          INMSG4
2971  016616  000207                  5$:     RTS     PC              ;RETURN TO INPUT ROUTINES
2972
2973  016620  020127  000300          CHKVCT: CMP     R1,#300         ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
2974  016624  002001                          BGE     1$              ;BR IF YES
2975  016626  000452                          BR      4$              ;BR IF NOT
2976  016630  020127  001000          1$:     CMP     R1,#1000        ;IS IT BELOW THE HIGH LIMIT?
2977  016634  002401                          BLT     2$              ;BR IF YES
2978  016636  000446                          BR      4$              ;BR IF NOT
2979  016640  032701  000007          2$:     BIT     #7,R1           ;CORRECT BOUNDARY ?
2980  016644  001043                          BNE     4$              ;BR IF NOT
2981  016646  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC AROUND ERROR BRANCH
2982  016652  005767  003502                  TST     RETFLG          ;ARE WE IN ECHO TESTS ?
2983  016656  001403                          BEQ     21$             ;BR IF NOT
2984  016660  010167  002640                  MOV     R1,DHVCT        ;SET UP DH11 VECTOR ADDR
2985  016664  000435                          BR      5$              ;CONTINUE
2986  016666  005767  003454          21$:    TST     DPFLG           ;PATTERNS TESTS ACTIVE ??
2987  016672  001403                          BEQ     22$             ;BR IF NOT
2988  016674  010167  002624                  MOV     R1,DHVCT        ;SET UP DEVICE VECTOR
2989  016700  000427                          BR      5$              ;CONTINUE
```

```
2990  016702  012702  021654          22$:    MOV     #DHVCTB,R2      ;POINT TO BEGIN OF VECTOR TABLE
2991  016706  032777  000001  162224          BIT     #BIT0,@SWR      ;ARE WE AUTOSIZING?
2992  016714  001011                          BNE     3$              ;BRANCH IF NOT.
2993  016716  012703  022014                  MOV     #DHVEC,R3       ;POINT TO BEGINING OF AUTOSIZER
2994                                                                  ;DH VECTOR TABLE.
2995  016722  016704  003216                  MOV     $DHSEL,R4
2996  016726  012322                  6$:     MOV     (R3)+,(R2)+     ;MOVE CONTENTS OF AUTOSIZER VECTOR
2997                                                                  ;TABLE TO TABLE USED BY PROGRAM.
2998  016730  006204                          ASR     R4
2999  016732  005704                          TST     R4              ;HAVE WE MOVED ALL TABLE ENTRIES?
3000  016734  001374                          BNE     6$              ;BRANCH IF NOT--ONE MORE TIME.
3001  016736  000410                          BR      5$              ;RETURN TO INPUT ROUTINES.
3002  016740  010122                  3$:     MOV     R1,(R2)+        ;SET UP A TABLE ENTRY
3003  016742  060001                          ADD     R0,R1           ;GENERATE NEXT DH11 ADDR
3004  016744  022702  021714                  CMP     #DHVCTB+40,R2   ;END OF TABLE ?
3005  016750  001373                          BNE     3$              ;BR IF NOT
3006  016752  000402                          BR      5$              ;RETURN TO INPUT ROUTINES
3007  016754  104401                  4$:     TYPE                    ;TELL HIM HE GOOFED
3008  016756  025204                          INMSG5
3009  016760  000207                  5$:     RTS     PC              ;RETURN TO INPUT ROUTINES
3010
3011                                  ;THESE TWO ROUTINES SERVICE UNEXPECTED BUS ERROR AND RSVD INSTR TRAPS
3012
3013  016762  010667  162210          BUSER:  MOV     SP,$REG6        ;SAVE THE SP
3014  016766  012667  162172                  MOV     (SP)+,$REG1     ;GET THE TRAP PC
3015  016772  012667  162170                  MOV     (SP)+,$REG2     ;GET THE TRAP PSW
3016  016776  012706  001100                  MOV     #STACK,SP       ;RESET THE STACK POINTER
3017  017002  012767  017012  162100          MOV     #1$,$LPERR      ;ALWAYS COME BACK TO 1$
3018  017010  104014                          ERROR   14              ;UNEXPECTED BUS ERROR TRAP
3019  017012  000005                  1$:     RESET                   ;PREPARE TO RESTART
3020  017014  004767  002406                  JSR     PC,CHPS1        ;GO CLEAR PSW
3021  017020  000167  175600                  JMP     CKRST2          ;GO RESTART THE PROGRAM
3022
3023  017024  010667  162146          RESERR: MOV     SP,$REG6        ;SAVE THE SP
3024  017030  012667  162130                  MOV     (SP)+,$REG1     ;GET THE TRAP PC
3025  017034  012667  162126                  MOV     (SP)+,$REG2     ;GET THE TRAP PSW
3026  017040  012706  001100                  MOV     #STACK,SP       ;RESET THE STACK POINTER
3027  017044  012767  017054  162036          MOV     #1$,$LPERR      ;ALWAYS COME BACK TO 1$
3028  017052  104015                          ERROR   15              ;UNEXPECTED RSVD INSTR ERROR TRAP
3029  017054  000005                  1$:     RESET                   ;PREPARE TO RESTART
3030  017056  004767  002344                  JSR     PC,CHPS1        ;GO CLEAR PSW
3031  017062  000167  175536                  JMP     CKRST2          ;GO RESTART THE PROGRAM
3032
3033                                  ;THIS ROUTINE IS CALLED WHEN A TEST NEEDS TO RESTORE THE TRAP
3034                                  ;CATCHER IN THE DH11 VECTOR
3035
3036  017066  016703  002432          RESTRP: MOV     DHVCT,R3        ;GET VECTOR ADDRESS
3037  017072  010313                          MOV     R3,(R3)         ;RESTORE THE TRAP CATCHER
3038  017074  062723  000002                  ADD     #2,(R3)+
3039  017100  005023                          CLR     (R3)+
3040  017102  010313                          MOV     R3,(R3)
3041  017104  062723  000002                  ADD     #2,(R3)+
3042  017110  005023                          CLR     (R3)+
3043  017112  000207                          RTS     PC              ;RETURN TO CALLING TEST
3044
3045                                  ;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

017114  005267  003222          TIMEIT: INC     TIMEB           ;COUNT B
017120  001005                          BNE     1$              ;BR IF NOT ZERO
017122  005367  003212                  DEC     TIMEA           ;COUNT TIME A
017126  001002                          BNE     1$              ;BR IF NO TIMEOUT
017130  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC TO ALLOW ERROR REPORT
017134  000207                  1$:     RTS     PC              ;RETURN TO THE CALLING TEST


                                ;THIS ROUTINE IS CALLED TO CLEAR ALL ENTRIES IN THE STATISTICS TABLES

017136  012705  030202          CLSTAT: MOV     #RTOTAL,R5      ;SET UP POINTER TO BEGINNING
017142  005025                  1$:     CLR     (R5)+           ;CLEAR ONE WORD
017144  022705  030502                  CMP     #RTOTAL+192.,R5 ;CLEARED ALL ENTRIES ??
017150  001374                          BNE     1$              ;BR IF NOT
017152  000207                          RTS     PC

                                ;THIS ROUTINE IS CALLED TO RETRIEVE A NEW LPR CONSTANT
                                ;FROM THE LPR TABLE (LPRTAB)

                                ;CALLING SEQUENCE:

                                ;       JSR     R5,SETLPR       ;CALL
                                ;       BR      NEWLIN          ;EXIT BRANCH - EXECUTED AFTER ALL
                                                                ;13 BAUD RATES EXERCISED

017154  022767  021576  002416  SETLPR: CMP     #CURLPR,LPRPTR  ;DONE ALL 13. ENTRIES ??
017162  001425                          BEQ     3$              ;BR IF YES
017164  017767  002410  002404          MOV     @LPRPTR,CURLPR  ;GET THE LPR CONSTANT
017172  105777  161742                  TSTB    @SWR            ; QUICK TEST ?
017176  100410                          BMI     1$              ;BR IF NOT - SUPPLY THE WHOLE THING
017200  022767  033500  002370          CMP     #33500,CURLPR   ;9600 BAUD TEST ??
017206  001404                          BEQ     1$              ;BR IF YES
017210  012767  177777  002322          MOV     #-1,QUICK       ;SET QUICK TEST FLAG
017216  000402                          BR      2$              ;CONTINUE
017220  005067  002314          1$:     CLR     QUICK           ;DO FULL TESTING AT 9600. BAUD
017224  062767  000002  002346  2$:     ADD     #2.LPRPTR       ;UPDATE THE TABLE POINTER
017232  062705  000002                  ADD     #2,R5           ;MOVE PC AROUND ERROR BRANCH
017236  000205                  3$:     RTS     R5              ;RETURN

                                ;THIS ROUTINE IS CALLED TO SETUP THE CHAR LENGTH SELECT BITS AND
                                ;LOAD THE OUTPUT DATA BUFFER

                                ;CALLING SEQUENCE:

                                ;       JSR     R5,SETCL        ;CALL
                                ;       BR      NEWLPR          ;EXIT BRANCH AFTER ALL FOUR LNGTHS TESTED

017240  005767  002276          SETCL:  TST     QUICKX          ;EXIT AFTER ONLY ONE CHAR LNGTH ?
017244  001034                          BNE     2$              ;BR IF YES
017246  005267  002332                  INC     CLSEL           ;GENERATE NEW CHAR LNGTH SELECT CODE
```

```
3102  017252  022767  000004  002324          CMP     #4,CLSEL        ;DONE FOUR OF THEM ??
3103  017260  001426                          BEQ     2$              ;BR IF YES
3104  017262  005767  002252                  TST     QUICK           ;QUICK TEST FLAG SET ?
3105  017266  001407                          BEQ     1$              ;BR IF NOT
3106  017270  005267  002246                  INC     QUICKX          ;SET QUICK TEST EXIT FLAG
3107  017274  005067  002304                  CLR     CLSEL           ;DO ONLY 5 BIT CHARS
3108  017300  012767  177760  002274          MOV     #177760,CHRCNT  ;DO  ONLY 32. CHAR BUFFER
3109  017306  042767  000003  002262  1$:     BIC     #3,CURLPR       ;SET UP THE CURRENT LPR
3110  017314  056767  002264  002254          BIS     CLSEL,CURLPR
3111  017322  006367  002254                  ASL     CHRCNT          ;GENERATE CHAR COUNT
3112  017326  004767  000006                  JSR     PC,SUBUF1       ;GO SET UP THE OUTPUT BUFFER
3113  017332  062705  000002                  ADD     #2,R5           ;MOVE PC AROUND EXIT BRANCH
3114  017336  000205                  2$:     RTS     R5              ;RETURN
3115
3116                                          ;THIS ROUTINE IS CALLED TO LOAD THE OUTPUT DATA BUFFER WITH THE
3117                                          ;REQUIRED BINARY COUNT PATTERN
3118
3119                                          ;CALLING SEQUENCE:
3120
3121                                          ;       JSR     PC,SUBUF1       ;CALL
3122
3123  017340                                  SUBUF1:
3124  017340  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
3125  017342  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
3126  017344  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
3127  017346  005004                          CLR     R4              ;INIT CHAR GENERATOR
3128  017350  016703  002226                  MOV     CHRCNT,R3       ;SET UP LOAD COUNT
3129  017354  012702  032776                  MOV     #TBUF,R2        ;SET UP BUFFER POINTER
3130  017360  110422                  1$:     MOVB    R4,(R2)+        ;LOAD A CHAR
3131  017362  005204                          INC     R4              ;GENERATE NEXT CHAR
3132  017364  005203                          INC     R3              ;COUNT ONE LOADED
3133  017366  001374                          BNE     1$              ;BR TIL BUFFER FULL
3134  017370  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
3135  017372  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
3136  017374  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
3137  017376  000207                          RTS     PC              ;RETURN
3138
3139                                          ;THIS ROUTINE IS CALLED TO SET UP THE PARITY SELECT BITS
3140                                          ;IN THE CURRENT LPR TEST CONSTANT
3141
3142                                          ;CALLING SEQUENCE:
3143
3144                                          ;       JSR     R5,SETPAR       ;CALL
3145                                          ;       BR      NEWCL           ;EXIT BRANCH
3146
3147  017400  022767  177777  002200  SETPAR: CMP     #-1,PARBIT      ;DONE ALL PARITY COMBOS ?
3148  017406  001444                          BEQ     5$              ;BR IF YES
3149  017410  005767  002124                  TST     QUICK           ;QUICK TEST FLAG SET ?
3150  017414  001403                          BEQ     1$              ;BR IF NOT
3151  017416  012767  000060  002162          MOV     #60,PARBIT      ;CHECK ODD PARITY ONLY
3152  017424  042767  000060  002144  1$:     BIC     #60,CURLPR      ;SET PARITY SELECT BITS
3153  017432  056767  002150  002136          BIS     PARBIT,CURLPR
3154  017440  005767  002142                  TST     PARBIT          ;SELECT BITS 00 ?
3155  017444  001004                          BNE     2$              ;BR IF NOT
3156  017446  012767  000020  002132          MOV     #20,PARBIT      ;SET SELECT BITS TO 01
3157  017454  000417                          BR      4$              ;EXIT
```

```
3158  017456  022767  000020  002122  2$:     CMP     #20,PARBIT      ;SELECT BITS 10 ?
3159  017464  001004                          BNE     3$              ;BR IF NOT
3160  017466  012767  000060  002112          MOV     #60,PARBIT      ;MAKE SELECT BITS 11
3161  017474  000407                          BR      4$              ;EXIT
3162  017476  022767  000060  002102  3$:     CMP     #60,PARBIT      ;SELECT BITS 11 ?
3163  017504  001005                          BNE     5$              ;BR IF NOT
3164  017506  012767  177777  002072          MOV     #-1,PARBIT      ;SET EXIT FLAG
3165  017514  062705  000002          4$:     ADD     #2,R5           ;MOVE RETURN PC AROUND EXIT BRANCH
3166  017520  000205                  5$:     RTS     R5              ;RETURN
3167
3168                                  ;THIS ROUTINE IS CALLED TO SET UP FOR KEYBOARD INTERRUPTS
3169
3170  017522  012767  017546  160330  KYBD1:  MOV     #KYBD2,60       ;SET UP THE INPUT VECTOR
3171  017530  012767  000340  160324          MOV     #340,62
3172  017536  012767  000100  160014          MOV     #100,177560     ;ENABLE KYBD INTRS
3173  017544  000207                          RTS     PC              ;RETURN TO START TESTING
3174
3175                                  ;THIS ROUTINE SERVICES THE KEYBOARD INTERRUPT AND LOOKS FOR AN "S"
3176                                  ;BEING TYPED TO INDICATE ABORT AND PRINT STATISTICS
3177
3178  017546  117746  161374          KYBD2:  MOVB    @$TKB,-(SP)     ;GET CHAR TYPED
3179  017552  142716  000200                  BICB    #200,(SP)       ;CLEAR UNWANTED BITS
3180  017556  122716  000123                  CMPB    #123,(SP)       ;WAS AN "S" TYPED ?
3181  017562  001420                          BEQ     1$              ;BR IF YES
3182  017564  022767  000176  161346          CMP     #SWREG,SWR      ;USING SOFTWARE SWR?
3183  017572  001024                          BNE     2$              ;BRANCH IF YES
3184  017574  126727  161334  000001          CMPB    $AUTOB,#1       ;RUNNING IN AUTO MODE?
3185  017602  001420                          BEQ     2$              ;BRANCH IF YES
3186  017604  122716  000007                  CMPB    #7,(SP)         ;IS IT A <↑G>
3187  017610  001015                          BNE     2$              ;BRANCH IF NO
3188  017612  005726                          TST     (SP)+           ;POP
3189  017614  104401  013761                  TYPE    ,$CNTLG         ;TYPE <↑G>
3190  017620  000167  173332                  JMP     $GTSWR
3191  017624  005726                  1$:     TST     (SP)+           ;POP
3192  017626  000005                          RESET                   ;ZAP THE WORLD
3193  017630  012706  001100                  MOV     #STACK,SP       ;RESET THE SP
3194  017634  004767  001566                  JSR     PC,CHPS1        ;GO CLEAR PSW
3195  017640  000167  164562                  JMP     PRSTAT          ;GO DUMP THE STATISTICS
3196  017644  005726                  2$:     TST     (SP)+           ;POP
3197  017646  000002                          RTI                     ;RETURN AND FORGET IT
3198
3199                                  ;THIS ROUTINE SENDS A TEST BUFFER TO REMOTE DH11 LINE
3200
3201  017650  016701  001646          SENDP2: MOV     DHADR,R1                ;SET UP DH SCR ADDR
3202  017654  012711  004000                  MOV     #BIT11,(R1)             ;CLEAR THE DH11
3203  017660  016711  002266                  MOV     LINE,(R1)               ;SET LINE SELECT
3204  017664  162705  032776                  SUB     #TBUF,R5                ;SET UP BYTE COUNT
3205  017670  005405                          NEG     R5
3206  017672  010561  000010                  MOV     R5,BCR(R1)
3207  017676  012761  032776  000006          MOV     #TBUF,CAR(R1)           ;SET CURRENT ADDRESS
3208  017704  016761  001666  000004          MOV     CURLPR,LPR(R1)          ;SET LINE PARAMETERS
3209  017712  016761  001616  000012          MOV     LINMSK,BAR(R1)          ;ACTIVATE THE LINE
3210
3211  017720  005711                  1$:     TST     (R1)                    ;DONE TRANSMITTING ??
3212  017722  100376                          BPL     1$                      ;BR IF NOT
3213  017724  000207                          RTS     PC                      ;RETURN TO CONTROL ROUTINE "SENDP1"
```

```
3214
3215                                    ;THIS ROUTINE IS CALLED TO LOAD FILLERS INTO ECHO BUFFER
3216
3217  017726  116704  002506          LDFILL: MOVB    FILLB,R4                ;GET COUNT OF FILLERS
3218  017732  012703  022375                  MOV     #ECBUF+1,R3             ;SET UP BUFFER POINTER
3219  017736  116767  161240  002430          MOVB    $TMPO,ECBUF             ;STORE LF CHAR
3220  017744  116722  161232                  MOVB    $TMPO,(R2)+             ;IN ECHO BUFFER TOO
3221  017750  116723  002462          1$:     MOVB    FILLA,(R3)+             ;LOAD A FILLER CHAR
3222  017754  116722  002456                  MOVB    FILLA,(R2)+
3223  017760  005304                          DEC     R4                      ;COUNT IT
3224  017762  001372                          BNE     1$                      ;BR TIL REQUIRED COUNT LOADED
3225  017764  116704  002450                  MOVB    FILLB,R4                ;SET UP BYTE COUNT REG
3226  017770  005204                          INC     R4
3227  017772  005404                          NEG     R4
3228  017774  010461  000010                  MOV     R4,BCR(R1)              ;LOAD BCR REG
3229  020000  000207                          RTS     PC                      ;RETURN TO RINT2
3230
3231                                    ;THIS ROUTINE IS CALLED TO SET UP XMITTER SPEED
3232
3233  020002  104401                  INXSP:  TYPE                            ;ASK USER TO TYPE SPEED
3234  020004  026631                          XSMSG1                          ;"TRANSMITTER SPEED ?"
3235  020006  012767  022156  002140  1$:     MOV     #XSPTAB,XSPTR           ;SET UP TABLE POINTER
3236  020014  042767  036000  001554          BIC     #36000,CURLPR           ;INIT SPEED SELECT BITS
3237  020022  104413                          RDDEC                           ;READ SPEED HE TYPED
3238  020024  005716                          TST     (SP)                    ;DEFAULT TO 9600. BAUD ?
3239  020026  001426                          BEQ     4$                      ;BR IF YES
3240  020030  027716  002120          2$:     CMP     @XSPTR,(SP)             ;TYPED ENTRY MATCH TABLE ENTRY ?
3241  020034  001010                          BNE     3$                      ;BR IF NOT
3242  020036  062767  000002  002110          ADD     #2,XSPTR                ;POINT TO SELECT BITS IN TABLE
3243  020044  057767  002104  001524          BIS     @XSPTR,CURLPR           ;SET SPEED SELECT BITS
3244  020052  005726                          TST     (SP)+                   ;FIX STACK
3245  020054  000417                          BR      5$                      ;CONTINUE
3246
3247  020056  062767  000004  002070  3$:     ADD     #4,XSPTR                ;POINT TO NEXT ENTRY
3248  020064  022767  022242  002062          CMP     #XSPTAB+52.,XSPTR       ;END OF TABLE ??
3249  020072  001356                          BNE     2$                      ;BR IF NOT
3250  020074  104401                          TYPE                            ;ERROR MESSAGE
3251  020076  026657                          XSMSG2                          ;"INVALID XMITR SPEED - TRY AGAIN"
3252  020100  005726                          TST     (SP)+                   ;FIX THE SP
3253  020102  000741                          BR      1$                      ;GO TRY AGAIN
3254
3255  020104  052767  032000  001464  4$:     BIS     #32000,CURLPR           ;SET UP DEFAULT TO 9600. BAUD
3256  020112  005726                          TST     (SP)+                   ;FIX STACK POINTER
3257
3258  020114  000207                  5$:     RTS     PC                      ;RETURN TO CALLER
3259
3260                                    ;THIS ROUTINE IS CALLED TO SET UP RECEIVER SPEED
3261
3262  020116  104401                  INRSP:  TYPE                            ;ASK USER TO TYPE SPEED
3263  020120  026722                          RSMSG1                          ;"RECEIVER SPEED ?"
3264  020122  012767  022244  002112  1$:     MOV     #RSPTAB,RSPTR           ;SET UP TABLE POINTER
3265  020130  042767  001700  001440          BIC     #1700,CURLPR            ;INIT SPEED SELECT BITS
3266  020136  104413                          RDDEC                           ;READ SPEED HE TYPED
3267  020140  005716                          TST     (SP)                    ;DEFAULT TO 9600. BAUD ?
3268  020142  001426                          BEQ     4$                      ;BR IF YES
3269  020144  027716  002072          2$:     CMP     @RSPTR,(SP)             ;TYPED ENTRY MATCH TABLE ENTRY ?
```

```
3270  020150  001010                        BNE     3$              ;BR IF NOT
3271  020152  062767  000002  002062        ADD     #2,RSPTR        ;POINT TO SELECT BITS IN TABLE
3272  020160  057767  002056  001410        BIS     @RSPTR,CURLPR   ;SET SPEED SELECT BITS
3273  020166  005726                        TST     (SP)+           ;FIX STACK
3274  020170  000417                        BR      5$              ;CONTINUE
3275
3276  020172  062767  000004  002042  3$:   ADD     #4,RSPTR        ;POINT TO NEXT ENTRY
3277  020200  022767  022330  002034        CMP     #RSPTAB+52.,RSPTR;END OF TABLE ??
3278  020206  001356                        BNE     2$              ;BR IF NOT
3279  020210  104401                        TYPE                    ;ERROR MESSAGE
3280  020212  026745                        RSMSG2                  ;"INVALID RCVR SPEED - TRY AGAIN"
3281  020214  005726                        TST     (SP)+           ;FIX THE SP
3282  020216  000741                        BR      1$              ;GO TRY AGAIN
3283
3284  020220  052767  001500  001350  4$:   BIS     #1500,CURLPR    ;SET UP DEFAULT TO 9600. BAUD
3285  020226  005726                        TST     (SP)+           ;FIX STACK POINTER
3286
3287  020230  000207                5$:     RTS     PC              ;RETURN TO CALLER
3288
3289
3290                                ;THIS ROUTINE IS CALLED TO SET UP LINE PARAMETERS FM KYBD
3291
3292  020232  105067  006005        LPRIN:  CLRB    EC2             ;CLEAR ECHO BUFFER
3293  020236  104401                        TYPE
3294  020240  026557                        LPMSG                   ;"DO YOU WANT TO CHANGE "LPR"?"
3295  020242  104410                1$:     RDCHR
3296  020244  012600                        MOV     (SP)+,R0        ;GET WHAT HE TYPED
3297  020246  122700  000015              CMPB    #15,R0          ;WAS IT A <CR> ??
3298  020252  001405                        BEQ     2$              ;BR IF YES
3299  020254  110067  005763              MOVB    R0,EC2          ;ECHO WHAT HE TYPED
3300  020260  104401                        TYPE
3301  020262  026243                        EC2
3302  020264  000766                        BR      1$              ;GO WAIT FOR TERMINATOR
3303
3304  020266  105767  005751        2$:     TSTB    EC2             ;<CR> ONLY ??
3305  020272  001411                        BEQ     3$              ;BR IF YES
3306  020274  122767  000116  005741        CMPB    #116,EC2        ;WAS IT A "NO" ??
3307  020302  001405                        BEQ     3$              ;BR IF IT WAS
3308  020304  122767  000131  005731        CMPB    #131,EC2        ;WAS IT A "YES" ??
3309  020312  001347                        BNE     LPRIN           ;GO ASK ALL OVER AGAIN
3310  020314  000407                        BR      4$              ;BR IF IT WAS "YES"
3311  020316  005767  001254        3$:     TST     CURLPR          ;HAS LPR BEEN SET UP AT ALL ?
3312  020322  001016                        BNE     5$              ;BR IF YES USE PREVIOUS LPR
3313  020324  012767  033503  001244        MOV     #33503,CURLPR   ;SET DEFAULT 9600 BAUD,8 BITS NO PARITY
3314  020332  000412                        BR      5$              ;CONTINUE
3315  020334  004767  177442        4$:     JSR     PC,INXSP        ;GO INPUT AND SET UP XMIT SPEED
3316  020340  004767  177552                JSR     PC,INRSP        ;GO INPUT AND SET UP RCVR SPEED
3317  020344  004767  000022                JSR     PC,INCL         ;GO INPUT AND SET UP CHAR LENGTH
3318  020350  004767  000162                JSR     PC,INSB         ;GO INPUT AND SET UP NO. OF STOP BITS
3319  020354  004767  000274                JSR     PC,INPB         ;GO INPUT AND SET UP PARITY SELECTION
3320  020360  004767  000410        5$:     JSR     PC,INFCHR       ;GO INPUT AND SET UP FILLER CHAR
3321  020364  004767  000474                JSR     PC,INFCNT       ;GO INPUT AND SET UP FILLER COUNT
3322  020370  000207                        RTS     PC              ;RETURN TO CALLER
3323
3324                                ;THIS ROUTINE IS CALLED TO SET UP CHAR LENCTH BITS
3325
```

```
3326  020372  105067  005645          INCL:   CLRB    EC2             ;CLEAR THE ECHO BUFFER
3327  020376  104401                          TYPE                    ;ASK FOR INPUT
3328  020400  027010                          CLMSG1                  ;"CHAR LENGTH - 6,7, OR 8 ?"
3329  020402  042767  000003  001166  1$:     BIC     #3,CURLPR       ;INIT CHAR LENGTH SELECT BITR
3330  020410  104410                          RDCHR                   ;GET THE CHAR HE TYPED
3331  020412  012600                          MOV     (SP)+,R0        ;GET WHAT HE TYPED
3332  020414  122700  000015                  CMPB    #15,R0          ;WAS IT A <CR> ??
3333  020420  001405                          BEQ     11$             ;BR IF IT WAS
3334  020422  110067  005615                  MOVB    R0,EC2          ;ECHO WHAT HE TYPED
3335  020426  104401                          TYPE
3336  020430  026243                          EC2
3337  020432  000763                          BR      1$              ;GO WAIT FOR TERMINATOR
3338  020434  105767  005603          11$:    TSTB    EC2             ;<CR> ONLY ??
3339  020440  001432                          BEQ     4$              ;BR IF YES
3340  020442  142767  000060  005573          BICB    #60,EC2         ;STRIP ASCII
3341  020450  122767  000006  005565          CMPB    #6,EC2          ;6 BITS ?
3342  020456  001004                          BNE     2$              ;BR IF NOT
3343  020460  052767  000001  001110          BIS     #1,CURLPR       ;SET UP FOR 6 BIT CHARS
3344  020466  000422                          BR      5$              ;CONTINUE
3345  020470  122767  000007  005545  2$:     CMPB    #7,EC2          ;7 BITS ?
3346  020476  001004                          BNE     3$              ;BR IF NOT
3347  020500  052767  000002  001070          BIS     #2,CURLPR       ;SET UP FOR 7 BIT CHARS
3348  020506  000412                          BR      5$              ;CONTINUE
3349  020510  122767  000010  005525  3$:     CMPB    #8.,EC2         ;8 BITS ?
3350  020516  001403                          BEQ     4$              ;BR IF YES
3351  020520  104401                          TYPE                    ;ERROR MESSAGE
3352  020522  027046                          CLMSG2                  ;"INVALID CHAR LENGTH = TRY AGAIN"
3353  020524  000722                          BR      INCL            ;GO TRY AGAIN
3354  020526  052767  000003  001042  4$:     BIS     #3,CURLPR       ;SET UP FOR 8 BIT CHARS
3355  020534  000207                  5$:     RTS     PC              ;RETURN TO CALLER
3356
3357                                  ;THIS ROUTINE IS CALLED TO SET UP NO. OF STOP BITS
3358
3359  020536  105067  005501          INSB:   CLRB    EC2             ;CLEAR ECHO BUFFER
3360  020542  104401                          TYPE                    ;ASK FOR INPUT
3361  020544  027112                          SBMSG1                  ;"NO. OF STOP BITS - 1 OR 2 ?"
3362  020546  104410                  1$:     RDCHR                   ;GET CHAR TYPED
3363  020550  012600                          MOV     (SP)+,R0        ;GET WHAT HE TYPED
3364  020552  122700  000015                  CMPB    #15,R0          ;WAS IT A <CR>
3365  020556  001405                          BEQ     11$             ;BR IF YES
3366  020560  110067  005457                  MOVB    R0,EC2          ;ECHO WHAT HE TYPED
3367  020564  104401                          TYPE
3368  020566  026243                          EC2
3369  020570  000766                          BR      1$              ;GO WAIT FOR TERMINATOR
3370  020572  105767  005445          11$:    TSTB    EC2             ;<CR> ONLY ??
3371  020576  001422                          BEQ     3$              ;BR IF YES
3372  020600  142767  000060  005435          BICB    #60,EC2         ;CLEAR ASCII JUNK
3373  020606  122767  000002  005427          CMPB    #2,EC2          ;2 STOP BITS ?
3374  020614  001004                          BNE     2$              ;BR IF NOT
3375  020616  052767  000004  000752          BIS     #4,CURLPR       ;SET UP FOR TWO STOP BITS
3376  020624  000412                          BR      4$              ;CONTINUE
3377  020626  122767  000001  005407  2$:     CMPB    #1,EC2          ;ONE STOP BIT ?
3378  020634  001403                          BEQ     3$              ;BR IF YES
3379  020636  104401                          TYPE                    ;ERROR MESSAGE
3380  020640  027151                          SBMSG2                  ;"INVALID NO. STOP BITS - TRY AGAIN"
3381  020642  000735                          BR      INSB            ;GO TRY AGAIN
```

```
3382  020644  042767  000004  000724  3$:     BIC     #4,CURLPR       ;SET UP FOR ONE STOP BIT
3383  020652  000207                  4$:     RTS     PC              ;RETURN TO CALLER
3384
3385                                  ;THIS ROUTINE IS CALLED TO SET UP PARITY SELECT BITS
3386
3387  020654  105067  005363          INPB:   CLRB    EC2             ;CLEAR ECHO BUFFER
3388  020660  104401                          TYPE                    ;ASK FOR INPUT
3389  020662  027217                          PBMSG1                  ;"PARITY - E,O, OR <CR> ?"
3390  020664  042767  000060  000704  1$:     BIC     #60,CURLPR      ;INIT FOR NO PARITY CHECKING
3391  020672  104410                          RDCHR                   ;GET CHAR TYPED
3392  020674  012600                          MOV     (SP)+,R0        ;GET WHAT HE TYPED
3393  020676  122700  000015                  CMPB    #15,R0          ;WAS IT A <CR> ??
3394  020702  001405                          BEQ     11$             ;BR IF IT WAS
3395  020704  110067  005333                  MOVB    R0,EC2          ;ECHO THE CHAR TYPED
3396  020710  104401                          TYPE
3397  020712  026243                          EC2
3398  020714  000763                          BR      1$              ;GO WAIT FOR TERMINATOR
3399  020716  105767  005321          11$:    TSTB    EC2             ;<CR> ONLY ??
3400  020722  001423                          BEQ     4$              ;BR IF YES
3401  020724  122767  000105  005311          CMPB    #105,EC2        ;EVEN PARITY ??
3402  020732  001004                          BNE     2$              ;BR IF NOT
3403  020734  052767  000060  000634          BIS     #60,CURLPR      ;SET UP FOR EVEN PARITY
3404  020742  000413                          BR      4$              ;CONTINUE
3405  020744  122767  000117  005271  2$:     CMPB    #117,EC2        ;ODD PARITY
3406  020752  001004                          BNE     3$              ;BR IF NOT
3407  020754  052767  000020  000614          BIS     #20,CURLPR      ;SET UP FOR ODD PARITY
3408  020762  000403                          BR      4$              ;CONTINUE
3409  020764  104401                  3$:     TYPE                    ;ERROR MESSAGE
3410  020766  027265                          PBMSG2                  ;"INVALID PARITY - TRY AGAIN"
3411  020770  000731                          BR      INPB            ;GO TRY AGAIN
3412  020772  000207                  4$:     RTS     PC              ;RETURN TO CALLER
3413
3414                                  ;THIS ROUTINE IS CALLED TO SET UP "FILL" CHAR
3415
3416  020774  105067  005243          INFCHR: CLRB    EC2             ;CLEAR ECHO BUFFER
3417  021000  005067  001432                  CLR     FILLA           ;INIT TEMP STORAGE FOR CHAR
3418  021004  104401                          TYPE                    ;GO ASK FOR FILLER CHAR
3419  021006  027324                          FILC1                   ;"FILL CHAR ?"
3420  021010  005067  001420          1$:     CLR     DHFILL          ;ININ FILL LOCATION
3421  021014  104410                          RDCHR                   ;GET CHAR TYPED
3422  021016  012600                          MOV     (SP)+,R0        ;GET WHAT HE TYPED
3423  021020  122700  000015                  CMPB    #15,R0          ;WAS IT A <CR> ??
3424  021024  001405                          BEQ     2$              ;BR IF YES
3425  021026  110067  005211                  MOVB    R0,EC2          ;ECHO WHAT HE TYPED
3426  021032  104401                          TYPE
3427  021034  026243                          EC2
3428  021036  000764                          BR      1$              ;GO WAIT FOR TERMINATOR
3429
3430  021040  105767  005177          2$:     TSTB    EC2             ;<CR> ONLY ??
3431  021044  001403                          BEQ     3$              ;BR IF YES
3432  021046  116767  005171  001361          MOVB    EC2,DHFILL+1    ;SET UP FILL CHAR
3433  021054  116767  001355  001354  3$:     MOVB    DHFILL+1,FILLA  ;SAVE FILL CHAR
3434  021062  000207                          RTS     PC              ;RETURN TO CALLER
3435
3436                                  ;THIS ROUTINE IS CALLED TO SET UP "FILL" COUNT
3437
```

```
3438  021064  005067  001350          INFCNT: CLR     FILLB           ;INIT TEMP. STORAGE FOR COUNT
3439  021070  104401                          TYPE                    ;ASK FOR COUNT
3440  021072  027352                          FILC2                   ;"FILL COUNT ?"
3441  021074  104412                          RDOCT                   ;GET OCIAL NO. TYPED
3442  021076  005716                          TST     (SP)            ;DEFAULT TO ONE ?
3443  021100  001403                          BEQ     1$              ;BR IF YES
3444  021102  111667  001326                  MOVB    (SP),DHFILL     ;SET UP COUNT TYPED
3445  021106  000403                          BR      2$              ;CONTINUE
3446  021110  112767  000001  001316  1$:     MOVB    #1,DHFILL       ;SET UP FOR  1 FILLER
3447  021116  005726                  2$:     TST     (SP)+           ;FIX THE SP
3448  021120  142767  000360  001306          BICB    #360,DHFILL     ;LIMIT COUNT TO 15. MAX
3449  021126  116767  001302  001304          MOVB    DHFILL,FILLB    ;SAVE IT FOR LATER
3450  021134  000207                          RTS     PC              ;RETURN TO CALLER
3451                                  ;THIS ROUTINE CALLED TO SET UP ALTERNATING 1/0 PATTERN
3452
3453  021136  004767  000246          SUPATA: JSR     PC,CLALL        ;GO CLEAR XMIT AND RCV BUFFERS
3454  021142  016700  000434                  MOV     CHRCNT,R0               ;GET CHAR COUNT
3455  021146  012705  032776                  MOV     #TBUF,R5        ;POINT TO XMIT BUFFER
3456  021152  112725  000252          1$:     MOVB    #252,(R5)+      ;LOAD A BYTE
3457  021156  005200                          INC     R0              ;COUNT IT
3458  021160  001374                          BNE     1$              ;BR TILL BUFFER FULL
3459  021162  000207                          RTS     PC              ;RETURN TO "DPATA" ROUTINE
3460
3461                                  ;THIS ROUTINE IS CALLED TO SET UP UP COUNT PATTERN
3462
3463  021164  004767  000220          SUPATU: JSR     PC,CLALL        ;GO CLEAR BUFFERS
3464  021170  016700  000406                  MOV     CHRCNT,R0       ;GET COUNT OF CHARS TO LOAD
3465  021174  012705  032776                  MOV     #TBUF,R5        ;POINT TO XMITTR BUFFER
3466  021200  005004                          CLR     R4              ;INIT CHAR GENERATOR
3467  021202  110425                  1$:     MOVB    R4,(R5)+        ;LOAD ONE BYTE
3468  021204  105204                          INCB    R4              ;GENERATE NEXT BYTE
3469  021206  005200                          INC     R0              ;COUNT IT
3470  021210  001374                          BNE     1$              ;BR TIL BUFFER FULL
3471  021212  000207                          RTS     PC              ;RETURN TO "DPATU" ROUTINE
3472
3473                                  ;THIS ROUTINE IS CALLED TO SET UP DOWN COUNT PATTERN
3474
3475  021214  004767  000170          SUPATD: JSR     PC,CLALL        ;CLEAR THE BUFFERS
3476  021220  016700  000356                  MOV     CHRCNT,R0       ;SET UP COUNT TO LOAD
3477  021224  012705  032776                  MOV     #TBUF,R5        ;POINT TO XMIT BUFFER
3478  021230  012704  000377                  MOV     #377,R4         ;INIT CHAR GENERATOR
3479  021234  110425                  1$:     MOVB    R4,(R5)+        ;LOAD ONE BYTE
3480  021236  105304                          DECB    R4              ;GENERATE NEW CHAR
3481  021240  005200                          INC     R0              ;COUNT IT
3482  021242  001374                          BNE     1$              ;BR TIL BUFFER FULL
3483  021244  000207                          RTS     PC              ;RETURN TO "DPATA" ROUTINE
3484
3485                                  ;THIS ROUTINE CALLED TO LOAD RANDOM DATA PATTERN
3486
3487  021246  004767  000136          SJPATR: JSR     PC,CLALL        ;GO CLEAR BUFFERS
3488  021252  016700  000324                  MOV     CHRCNT,R0       ;SET UP COUNT TO LOAD
3489  021256  012705  032776                  MOV     #TBUF,R5        ;POINT TO XMITTR BUFFER
3490  021262  012767  125252  001074          MOV     #125252,RANA    ;INIT RANDOM NUMBER GENERATOR
3491
3492  021270  066767  001070  001070  1$:     ADD     RANA,RANB       ;GENERATE RANDOM NO.
3493  021276  005567  001062                  ADC     RANA
```

```
3494  021302  066767  001060  001054          ADD     RANB,RANA
3495  021310  005567  001052                  ADC     RANB
3496
3497  021314  116725  001044                  MOVB    RANA,(R5)+      ;LOAD A BYTE
3498  021320  005200                          INC     R0              ;COUNT IT
3499  021322  001362                          BNE     1$              ;BR TIL BUFFER FULL
3500  021324  000207                          RTS     PC              ;RETURN TO "DPATR" ROUTINE
3501
3502                                  ;THIS ROUTINE LOADS A SINGLE CHAR THROUGHOUT BUFFER
3503
3504  021326  004767  000056          SUPATS: JSR     PC,CLALL        ;GO CLEAR BUFFERS
3505  021332  016700  000244                  MOV     CHRCNT,R0       ;INIT CHAR COUNTER
3506  021336  012705  032776                  MOV     #TBUF,R5        ;POINT TO XMIT BUFFER
3507  021342  116725  001010          1$:     MOVB    SINGLE,(R5)+    ;LOAD ONE CHAR
3508  021346  005200                          INC     R0              ;COUNT IT
3509  021350  001374                          BNE     1$              ;BR TIL BUFFER FULL
3510  021352  000207                          RTS     PC              ;RETURN TO "DPATS" ROUTINE
3511
3512                                  ;THIS ROUTINE CALLED TO INIT CHAR LENGTH MASK FOR PATTERNS TESTS
3513
3514  021354  016700  000216          SUCLMK: MOV     CURLPR,R0       ;GET CURRENT "LPR"
3515  021360  012767  000340  001002          MOV     #340,CLMSK      ;INIT FOR 5 BIT CHARS
3516  021366  042700  177774                  BIC     #177774,R0      ;MASK OFF ALL BUT CL BITS
3517  021372  005700                  1$:     TST     R0              ;DONE SETUP ?
3518  021374  001404                          BEQ     2$              ;BR IF YES
3519  021376  106367  000766                  ASLB    CLMSK           ;SHIFT MASK LEFT
3520  021402  005300                          DEC     R0              ;COUNT IT
3521  021404  000772                          BR      1$              ;GO SEE IF ITS RIGHT ON
3522  021406  000207                  2$:     RTS     PC              ;RETURN TO CALLER
3523                                  ;ROUTINE TO CLEAR XMIT AND RECEIVER BUFFERS
3524
3525  021410  012700  032776          CLALL:  MOV     #TBUF,R0        ;SET UP POINTER
3526  021414  005020                  1$:     CLR     (R0)+           ;CLEAR A WORD
3527  021416  022700  034126                  CMP     #ENBUFS,R0      ;DONE ALL LOCATIONS ?
3528  021422  001374                          BNE     1$              ;BR IF NO
3529  021424  000207                          RTS     PC
```

```
                                        ;THIS ROUTINE IS CALLED TO SET PSW PRIORITY TO 000 IN ORDER
                                        ;TO BE LSI11 COMPATIBLE
                                
021426  012746  000000          CHPS1:  MOV     #0,-(SP)        ;NEW PSW
021432  012746  021440                  MOV     #1$,-(SP)       ;NEW PC
021436  000002                          RTI                     ;CHANGE PSW
021440  000207                  1$:     RTS     PC              ;RETURN TO CALLING TEST

                                        ;THIS ROUTINE DOES THE SAME THING EXCEPT IT SET THE PSW
                                        ;PRIORITY TO 340 (LEVEL 7 ) TO LOCK OUT INTRS

021442  012746  000340          CHPS2:  MOV     #340,-(SP)      ;NEW PSW
021446  012746  021454                  MOV     #1$,-(SP)       ;NEW PC
021452  000002                          RTI                     ;CHANGE THE PSW
021454  000207                  1$:     RTS     PC              ;RETURN TO CALLING TEST

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

021456  005046                  SAPS:   CLR     -(SP)           ;TEMP STORAGE TO SAVE PSW
021460  016746  156350                  MOV     34,-(SP)        ;SAVE TRAP VECTOR POINTER
021464  012767  021474  156342          MOV     #1$,34          ;GO TO 1$ ON TRAP
021472  104400                          TRAP                    ;GO TO IT
021474  016666  000002  000006  1$:     MOV     2(SP),6(SP)     ;GET PSW SAVED
021502  012716  021510                  MOV     #2$,(SP)        ;GO TO 2$ ON RTI
021506  000002                          RTI
021510  012667  156320          2$:     MOV     (SP)+,34        ;RESTORE VECTOR
021514  012667  157462                  MOV     (SP)+,$TMPO     ;FINALLY SAVE PSW IN $TMPO
021520  000207                          RTS     PC
```

```
                                .SBTTL DH11 PROGRAM CONSTANTS AND VARIABLES
                                ;;*****************************************************************
                                ;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.

021522      000000  DHADR:  0           ;HOLDS THE "SCR" ADDRESS OF THE DH11 UNDER TEST
021524      000000  DHVCT:  0           ;HOLDS THE 1ST VECTOR ADDRESS OF THE DH11 UNDER TEST
021526      000000  SELMSK: 0           ;BIT TST MARKER FOR SELECTING DH11'S
021530      000001  DHSEL:  1           ;SPECIFIES DH11'S SELECTED FOR TEST
021532      177777  LINSEL: 177777      ;SPECIFIES LINES TO TEST
021534      000000  LINMSK: 0           ;MARKER USED TO TEST FOR LINES TO TEST
021536      000000  DRPLIN: 0           ;DROPPED LINE FLAGS

021540      000000  QUICK:  0           ;QUICK TEST FLAG - ALLOWS SINGLE PATTERN TEST
                                        ;ON ALL TESTS NOT USING 9600. BAUD
021542      000000  QUICKX: 0           ;ALLOWS SUB-TEST EXIT DURING QUICK TEST


                                ;THIS TABLE CONTAINS THIRTEEN CONSTANTS USED TO ESTABLISH
                                ;THE INITIAL LINE PARAMETERS FOR THE THIRTEEN PROGRAMMABLE BAUD
                                ;RATES - EACH PARAMETER INITIALLY SPECIFIES NO PARITY CHECKING
                                ;AND A CHARACTER LENGTH OF FIVE BITS

021544      033500  LPRTAB: 33500               ;9600 BAUD
021546      004200          4200                ;75 BAUD
021550      006300          6300                ;110 BAUD
021552      010400          10400               ;134.5 BAUD
021554      012500          12500               ;150 BAUD
021556      014600          14600               ;200 BAUD
021560      016700          16700               ;300 BAUD
021562      021000          21000               ;600 BAUD
021564      023100          23100               ;1200 BAUD
021566      025200          25200               ;1800 BAUD
021570      027300          27300               ;2400 BAUD
021572      031400          31400               ;4800 BAUD
021574      002100          2100                ;50 BAUD

021576      000000  CJRLPR: 0                   ;CONTAINS CURRENT "LPR" CONSTANT

021600      000000  LPRPTR: 0                   ;CONTAINS POINTER TO LPR TABLE
021602      000000  CHRCNT: 0                   ;LOADED WITH CURRENT CHAR COUNT

021604      000000  CLSEL:  0                   ;CHAR LENGTH SELECT PARAMETER
021606      000000  PARBIT: 0                   ;PARITY SELECT PARAMETER

021610      000000  RDONE:  0                   ;SOFTWARE DONE FLAG
```

```
3617  021612  000000                RBFEND: 0                       ;HOLDS END OF BUFFER ADDRESS
3618
3619                                ;DH11 ADDRESS TABLE - THIS TABLE CONTAINS THE "SCR" ADDRESS FOR UP TO
3620                                ;SIXTEEN DH11'S
3621
3622  021614  160020                DHADTB: 160020          ;ADDRESS OF FIRST DH11
3623  021616  160040                        160040          ;ADDRESS OF SECOND DH11
3624  021620  160060                        160060
3625  021622  160100                        160100
3626  021624  160120                        160120
3627  021626  160140                        160140
3628  021630  160160                        160160
3629  021632  160200                        160200
3630  021634  160220                        160220
3631  021636  160240                        160240
3632  021640  160260                        160260
3633  021642  160300                        160300
3634  021644  160320                        160320
3635  021646  160340                        160340
3636  021650  160360                        160360
3637  021652  160400                        160400          ;ADDRESS OF THE LAST DH11
3638
3639                                ;DH11 VECTOR TABLE - THIS TABLE CONTAINS THE VECTOR ADDRESSES FOR UP
3640                                ;TO SIXTEEN DH11'S
3641
3642  021654  000330                DHVCTB: 330             ;ADDRESS OF VECTOR FOR FIRST DH11
3643  021656  000350                        350             ;ADDRESS OF VECTOR FOR SECOND DH11
3644  021660  000370                        370
3645  021662  000410                        410
3646  021664  000430                        430
3647  021666  000450                        450
3648  021670  000470                        470
3649  021672  000510                        510
3650  021674  000530                        530
3651  021676  000550                        550
3652  021700  000570                        570
3653  021702  000610                        610
3654  021704  000630                        630
3655  021706  000650                        650
3656  021710  000670                        670
3657  021712  000710                        710             ;ADDRESS OF VECTOR FOR LAST DH11
3658
3659  021714  003000                VCFLG:  0               ;VECTOR DISPLACEMENT FLAG
3660
3661
3662                                ;BR PRIORITY LEVEL TABLE - THIS TABLE CONTAINS THE PRIORITY LEVELS
3663                                ;FOR UP TO SIXTEEN DH11'S - THE RCVR LEVEL IS STORED IN THE LOW BYTE
3664                                ;AND THE XMTTR LEVEL IN THE HIGH BYTE
3665
3666  021716  120240                BRLVL:  120240          ;BRLEVELS FOR FIRST DH11
3667  021720  120240                        120240          ;BR LEVELS FOR SECOND DH11
3668  021722  120240                        120240
3669  021724  120240                        120240
3670  021726  120240                        120240
3671  021730  120240                        120240
3672  021732  120240                        120240
```

```
3673  021734  120240                          120240
3674  021736  120240                          120240
3675  021740  120240                          120240
3676  021742  120240                          120240
3677  021744  120240                          120240
3678  021746  120240                          120240
3679  021750  120240                          120240
3680  021752  120240                          120240
3681  021754  120240                          120240          ;BR LEVELS FOR LAST DH11
3682
3683                                  ;THIS DH ADDRESS TABLE IS FILLED BY THE AUTOSIZER.
3684
3685  021756                          DHADRS:
3686  021756  000000                  .WORD   0
3687  021760  000000                  .WORD   0
3688  021762  000000                  .WORD   0
3689  021764  000000                  .WORD   0
3690  021766  000000                  .WORD   0
3691  021770  000000                  .WORD   0
3692  021772  000000                  .WORD   0
3693  021774  000000                  .WORD   0
3694  021776  000000                  .WORD   0
3695  022000  000000                  .WORD   0
3696  022002  000000                  .WORD   0
3697  022004  000000                  .WORD   0
3698  022006  000000                  .WORD   0
3699  022010  000000                  .WORD   0
3700  022012  000000                  .WORD   0
3701
3702                                  ;THIS DH VECTOR TABLE IS FILLED BY THE AUTOSIZER.
3703
3704  022014                          DHVEC:
3705  022014  000000                  .WORD   0
3706  022016  000000                  .WORD   0
3707  022020  000000                  .WORD   0
3708  022022  000000                  .WORD   0
3709  022024  000000                  .WORD   0
3710  022026  000000                  .WORD   0
3711  022030  000000                  .WORD   0
3712  022032  000000                  .WORD   0
3713  022034  000000                  .WORD   0
3714  022036  000000                  .WORD   0
3715  022040  000000                  .WORD   0
3716  022042  000000                  .WORD   0
3717  022044  000000                  .WORD   0
3718  022046  000000                  .WORD   0
3719
3720                                  ;THIS DM ADDRESS TABLE IF FILLED BY THE AUTOSIZER.
3721
3722  022050                          DMADRS:
3723  022050  000000                  .WORD   0
3724  022052  000000                  .WORD   0
3725  022054  000000                  .WORD   0
3726  022056  000000                  .WORD   0
3727  022060  000000                  .WORD   0
3728  022062  000000                  .WORD   0
```

```
3729  022064  000000                  .WORD   0
3730  022066  000000                  .WORD   0
3731  022070  000000                  .WORD   C
3732  022072  000000                  .WORD   0
3733  022074  000000                  .WORD   0
3734  022076  000000                  .WORD   0
3735  022100  000000                  .WORD   0
3736  022102  000000                  .WORD   0
3737  022104  000000                  .WORD   0
3738
3739                                  ;THIS DM VECTOR TABLE IS FILLED BY THE AUTOSIZER.
3740
3741  022106                          DMVEC:
3742  022106  000000                  .WORD   0
3743  022110  000000                  .WORD   0
3744  022112  000000                  .WORD   0
3745  022114  000000                  .WORD   0
3746  022116  000000                  .WORD   0
3747  022120  000000                  .WORD   0
3748  022122  000000                  .WORD   0
3749  022124  000000                  .WORD   0
3750  022126  000000                  .WORD   0
3751  022130  000000                  .WORD   0
3752  022132  000000                  .WORD   0
3753  022134  000000                  .WORD   0
3754  022136  000000                  .WORD   0
3755  022140  000000                  .WORD   0
3756
3757  022142  000000          ADRVEC: 0               ;ADDRESSES BETWEEN VECTORS - FILLED BY THE AUTOSIZER.
3758  022144  000000          $DHSEL: 0               ;DEVICE SELECT PARAMETER - FILLED BY THE AUTOSIZER.
3759  022146     000          DHRLVL: .BYTE   0               ;BR LEVEL FOR RCVR
3760  022147     000          DHTLVL: .BYTE   0               ;BR LEVEL FOR XMITTER
3761  022150  000000          DHNUM:  0               ;CONTAINS NUMBER OF THE DH11 UNDER TEST
3762  022152  000000          LINE:   0               ;CONTINES NUMBER OF THE LINE UNDER TEST
3763                          ;TABLES USED TO SELECT XMITTR AND RCVR SPEEDS
3764
3765                          ;THE TABLES CONSIST OF 13. TWO WORD ENTRIES - ONE FOR EACH
3766                          ;ALLOWABLE BAUD RATE. THE FIRST WORD IS THE ACTUAL BAUD RATE
3767                          ;IN DECIMAL AND THE SECOND WORD IS THE ENCODED BINARY WORD
3768                          ;THAT SETS THAT BAUD RATE IN THE "LPR"
3769
3770  022154  000000          XSPTR:  0                       ;CONTAINS POINTER TO FOLLOWING TABLE
3771
3772  022156  000062          XSPTAB: 50.                     ;50. BAUD
3773  022160  002000                  2000
3774  022162  000113                  75.                     ;75. BAUD
3775  022164  004000                  4000
3776  022166  000156                  110.                    ;110. BAUD
3777  022170  006000                  6000
3778  022172  002501                  1345.                   ;134.5 BAUD
3779  022174  010000                  10000
3780  022176  000226                  150.                    ;150. BAUD
3781  022200  012000                  12000
3782  022202  000310                  200.                    ;200. BAUD
3783  022204  014000                  14000
3784  022206  000454                  300.                    ;300 BAUD
```

```
3785  022210  016000                          16000
3786  022212  001130                          600.          ;600. BAUD
3787  022214  020000                          20000
3788  022216  002260                          1200.         ;1200. BAUD
3789  022220  022000                          22000
3790  022222  003410                          1800.         ;1800. BAUD
3791  022224  024000                          24000
3792  022226  004540                          2400.         ;2400. BAUD
3793  022230  026000                          26000
3794  022232  011300                          4800.         ;4800. BAUD
3795  022234  030000                          30000
3796  022236  022600                          9600.         ;9600. BAUD
3797  022240  032000                          32000
3798  022242  000000              RSPTR:      0             ;CONTAINS POINTER TO FOLLOWING TABLE
3799
3800  022244  000062              RSPTAB:     50.           ;50. BAUD
3801  022246  000100                          100
3802  022250  000113                          75.           ;75. BAUD
3803  022252  000200                          200
3804  022254  000156                          110.          ;110. BAUD
3805  022256  000300                          300
3806  022260  002501                          1345.         ;134.5 BAUD
3807  022262  000400                          400
3808  022264  000226                          150.          ;150. BAUD
3809  022266  000500                          500
3810  022270  000310                          200.          :200. BAUD
3811  022272  000600                          600
3812  022274  000454                          300.          ;300 BAUD
3813  022276  000700                          700
3814  022300  001130                          600.          ;600. BAUD
3815  022302  001000                          1000
3816  022304  002260                          1200.         ;1200. BAUD
3817  022306  001100                          1100
3818  022310  003410                          1800.         ;1800. BAUD
3819  022312  001200                          1200
3820  022314  004540                          2400.         ;2400. BAUD
3821  022316  001300                          1300
3822  022320  011300                          4800.         ;4800. BAUD
3823  022322  001400                          1400
3824  022324  022600                          9600.         ;9600. BAUD
3825  022326  001500                          1500
3826
3827                          ;ADDRESS POINTERS TO SET UP TABLES WHEN INPUTTING PARAMETERS
3828
3829  022330  000000              ADPTR:      0             ;POINTS TO ADDRESS TABLE
3830  022332  000000              VCPTR:      0             ;POINTS TO VECTOR TABLE
3831  022334  000000              BRPTR:      0             ;POINTS TO BR LEVEL TABLE
3832
3833  022336  000000              TITFLG:     0             ;FLAG TO ALLOW PRINTING TITLE ONLY ONCE
3834  022340  000000              TIMEA:      0             ;GENERAL PURPOSE TIMERS
3835  022342  000000              TIMEB:      0
3836
3837  022344  000000              CEXIT:      0             ;CONTROL-C EXIT FLAG FM ECHO TESTS
3838  022346  000000              DPFLG:      0             ;PATTERNS TEST FLAG
3839  022350  000000              DATCNT:     0             ;ITERATION COUNTER FOR PATTERNS TEST
3840  022352  000000              DATPAT:     0             ;FLAGS TYPE PATTERN
```

```
3841  022354  000000                PATFLG: 0                       ;DATA PATTERNS <CR> SEQUENCE FLAG
3842  022356  000000                SINGLE: 0                       ;HOLDS SINGLE CHAR TEST PATTERN
3843  022360  000000                RETFLG: 0                       ;ECHO TEST RETURN FLAG FM SETUP
3844  022362  000012                PATLIM: 10.                     ;PATTERNS TESTS ITERATION COUNT
3845  022364  000000                RANA:   0                       ;RANDOM NO. ACCUMULATORS
3846  022366  000000                RANB:   0
3847  022370  000000                CLMSK:  0                       ;CHAR LENGTH BIT CLR MASK
3848  022372  000000                EXFLAG: 0                       ;ECHO TEST EXIT FLAGS
3849  022374  000020                ECBUF:  .BLKW   16.             ;DATA BUFFER FOR SINGLE LINE ECHO TEST
3850  022434  000000                DHFILL: 0                       ;FILL CHAR AND COUNT FOR SINGLE LINE
3851                                                                ;ECHO TESTS
3852  022436  000000                FILLA:  0                       ;TEMP STORAGE FOR FILLER CHAR
3853  022440  000000                FILLB:  0                       ;SAME FOR COUNT
3854
3955
```

```
                                        .SBTTL STANDARD ERROR MESSAG BUFFERS
                                        ;;*****************************************************************
                                        ;ERROR MESSAGE INFORMATION - MESSAGE BUFFERS AND POINTERS
                                        ;;*****************************************************************

                                        ;INFORMATION FOR MESSAGE 1

 022442  047516  020116  054105  EM1:    .ASCIZ  'NON EX MEMORY ERROR - DROPPED LINE #  '
 022450  046440  046505  051117
 022456  020131  051105  047522
 022464  020122  020055  051104
 022472  050117  042520  020104
 022500  044514  042516  021440
 022506  020040     000
 022511     040  050050  024503  DH1:    .ASCIZ  ' (PC)   CURLPR  DEVADR  REGADR   WAS     S/R'
 022516  020040  041440  051125
 022524  050114  020122  042040
 022532  053105  042101  020122
 022540  051040  043505  042101
 022546  020122  020040  040527
 022554  020123  020040  020040
 022562  027523  000102
                                        .EVEN
 022566  001116  021576  001164  DT1:    .WORD   $ERRPC,CURLPR,$REG1,$REG2,$REG3,$REG4,0
 022574  001166  001170  001172
 022602  000000
 022604     000     000     000  DF2:    .BYTE   0,0,0,0,0,0,0,0
 022607     000     000     000
 022612     000     000

                                        ;INFORMATION FOR MESSAGE 2

 022614  051124  047101  046523  EM2:    .ASCIZ  'TRANSMITTER FALSE INTERRUPT - DROPPED LINE #  '
 022622  052111  042524  020122
 022630  040506  051514  020105
 022636  047111  042524  051122
 022644  050125  020124  020055
 022652  051104  050117  042520
 022660  020104  044514  042516
 022666  021440  020040     000

                                        ;INFORMATION FOR MESSAGE 3

 022673     102  043125  042506  EM3:    .ASCIZ  'BUFFER ACTIVE REGISTER ERROR - DROPPED LINE #  '
 022700  020122  041501  044524
 022706  042526  051040  043505
 022714  051511  042524  020122
 022722  051105  047522  020122
 022730  020055  051104  050117
 022736  042520  020104  044514
 022744  042516  021440  020040
 022752     000

                                        ;INFORMATION FOR MESSAGE 4
```

```
3912  022753  102     052131  020105  EM4:    .ASCIZ  'BYTE COUNT REGISTER ERROR - DROPPED LINE #  '
3913  022760  047503  047125  020124
3914  022766  042522  044507  052123
3915  022774  051105  042440  051122
3916  023002  051117  026440  042040
3917  023010  047522  050120  042105
3918  023016  046040  047111  020105
3919  023024  020043  000040
3920
3921                                  ;INFORMATION FOR MESSAGE 5
3922
3923  023030  052503  051122  047105  EM5:    .ASCIZ  'CURRENT ADDRESS REGISTER ERROR - DROPPED LINE #  '
3924  023036  020124  042101  051104
3925  023044  051505  020123  042522
3926  023052  044507  052123  051105
3927  023060  042440  051122  051117
3928  023066  026440  042040  047522
3929  023074  050120  042105  046040
3930  023102  047111  020105  020043
3931  023110  000040
3932
3933                                  ;INFORMATION FOR MESSAGE 6
3934
3935  023112  044523  047514  047440  EM6:    .ASCIZ  'SILO OVERFLOW ERROR - DROPPED LINE #  '
3936  023120  042526  043122  047514
3937  023126  020127  051105  047522
3938  023134  020122  020055  051104
3939  023142  050117  042520  020104
3940  023150  044514  042516  021440
3941  023156  020040  000
3942
3943                                  ;INFORMATION FOR MESSAGE 7
3944
3945  023161  122     041505  044505  EM7:    .ASCIZ  'RECEIVER FALSE INTERRUPT - DROPPED LINE #  '
3946  023166  042526  020122  040506
3947  023174  051514  020105  047111
3948  023202  042524  051122  050125
3949  023210  020124  020055  051104
3950  023216  050117  042520  020104
3951  023224  044514  042516  021440
3952  023232  020040  000
3953
3954                                  ;INFORMATION FOR MESSAGE 10
3955
3956  023235  111     053116  046101  EM10:   .ASCIZ  'INVALID DATA IN SILO - DROPPED LINE #  '
3957  023242  042111  042040  052101
3958  023250  020101  047111  051440
3959  023256  046111  020117  020055
3960  023264  051104  050117  042520
3961  023272  020104  044514  042516
3962  023300  021440  020040  000
3963  023305  040     050050  024503  DH2:    .ASCIZ  ' (PC)   CURLPR  CHAR #  WASADR  SHBADR   WAS     S/B'
3964  023312  020040  041440  051125
3965  023320  050114  020122  041440
3966  023326  040510  020122  020043
3967  023334  053440  051501  042101
```

```
3968  023342  020122  051440  041110
3969  023350  042101  020122  020040
3970  023356  040527  020123  020040
3971  023364  020040  027523  000102
3972                                  .EVEN
3973  023372  001116  021576  001162  DT2:    .WORD   $ERRPC,CURLPR,$REG0,$REG1,$REG2,$REG3,$REG4,0
3974  023400  001164  001166  001170
3975  023406  001172  000000
3976
3977                                  ;INFORMATION FOR MESSAGE 11
3978
3979  023412  040504  040524  042440  EM11:   .ASCIZ  'DATA ERROR - LINE #  '
3980  023420  051122  051117  026440
3981  023426  046040  047111  020105
3982  023434  020043  000040
3983
3984                                  ;INFORMATION FOR MESSAGE 12
3985
3986  023440  042524  052123  052040  EM12:   .ASCIZ  'TEST TIMEOUT - DROPPED LINE #  '
3987  023446  046511  047505  052125
3988  023454  026440  042040  047522
3989  023462  050120  042105  046040
3990  023470  047111  020105  020043
3991  023476  000040
3992  023500  024040  041520  020051  DH3:    .ASCIZ  ' (PC)   CURLPR  RTOTAL  XTOTAL  RDONE'
3993  023506  020040  052503  046122
3994  023514  051120  020040  052122
3995  023522  052117  046101  020040
3996  023530  052130  052117  046101
3997  023536  020040  042122  047117
3998  023544  000105
3999                                  .EVEN
4000  023546  001116  021576  001202  DT3:    .WORD   $ERRPC,CURLPR,$TMP0,$TMP1,RDONE,0
4001  023554  001204  021610  000000
4002
4003                                  ;INFORMATION FOR MESSAGE 13
4004
4005  023562  001202  001204  001206  DT4:    .WORD   $TMP0,$TMP1,$TMP2,$TMP3,$TMP4,$TMP5,$TMP6,0
4006  023570  001210  001212  001214
4007  023576  001216  000000
4008  023602     000     001     001  DF1:    .BYTE   0,1,1,1,1,1,1,0
4009  023605     001     001     001
4010  023610     001     000
4011
4012                                  ;INFORMATION FOR MESSAGE 14
4013
4014  023612  052502  020123  051105  EM14:   .ASCIZ  'BUS ERROR TRAP TO 04'
4015  023620  047522  020122  051124
4016  023626  050101  052040  020117
4017  023634  032060     000
4018  023637     040  050050  024503  DH4:    .ASCIZ  ' (PC)    (PS)    (SP)   TRAPPC  TRAPPS'
4019  023644  020040  020040  050050
4020  023652  024523  020040  020040
4021  023660  051450  024520  020040
4022  023666  052040  040522  050120
4023  023674  020103  052040  040522
```

```
4024  023702  050120  000123
4025                                  .EVEN
4026  023706  001116  001202  001176  DT5:    .WORD   $ERRPC,$TMPO,$REG6,$REG1,$REG2,0
4027  023714  001164  001166  000000
4028
4029                                  ;INFORMATION FOR MESSAGE 15
4030
4031  023722  051522  042126  044440  EM15:   .ASCIZ  'RSVD INSTR TRAP TO 10'
4032  023730  051516  051124  052040
4033  023736  040522  020120  047524
4034  023744  030440  000060
4035
4036                                  ;INFORMATION FOR MESSAGE 16
4037
4038  023750  044523  043516  042514  EM16:   .ASCIZ  'SINGLE LINE ECHO TEST - INTR WAIT TIMEOUT'
4039  023756  046040  047111  020105
4040  023764  041505  047510  052040
4041  023772  051505  020124  020055
4042  024000  047111  051124  053440
4043  024006  044501  020124  044524
4044  024014  042515  052517  000124
4045  024022  024040  041520  020051  DH5:    .ASCIZ  ' (PC)   DEVADR   LINE    (SCR)  CURLPR  EXFLAG'
4046  024030  020040  042504  040526
4047  024036  051104  020040  046040
4048  024044  047111  020105  020040
4049  024052  024040  041523  024522
4050  024060  020040  052503  046122
4051  024066  051120  020040  054105
4052  024074  046106  043501      000
4053          024102                  .EVEN
4054  024102  001116  001164  022152  DT6:    .WORD   $ERRPC,$REG1,LINE,$TMPO,CURLPR,EXFLAG,0
4055  024110  001202  021576  022372
4056  024116  000000
4057
4058
4059                                  ;INFORMATION FOR MESSAGE 17
4060
4061  024120  046101  042524  047122  EM17:   .ASCIZ  'ALTERNATING 1/0 PATTERN TEST DONE'
4062  024126  052101  047111  020107
4063  024134  027461  020060  040520
4064  024142  052124  051105  020116
4065  024150  042524  052123  042040
4066  024156  047117  000105
4067  024162  024040  041520  020051  DH6:    .ASCIZ  ' (PC)   DEVADR   LINE   CURLPR  ICOUNT'
4068  024170  020040  042504  040526
4069  024176  051104  020040  046040
4070  024204  047111  020105  020040
4071  024212  052503  046122  051120
4072  024220  020040  041511  052517
4073  024226  052116      000
4074          024232                  .EVEN
4075  024232  001116  021522  022152  DT7:    .WORD   $ERRPC,DHADR,LINE,CURLPR,$REG0,0
4076  024240  021576  001162  000000
4077
4078                                  ;INFORMATION FOR MESSAGE 20
4079
```

```
4080  024246  044502  040516  054522  EM20:   .ASCIZ  'BINARY UP COUNT PATTERN TEST DONE'
4081  024254  052440  020120  047503
4032  024262  047125  020124  040520
4083  024270  052124  051105  020116
4084  024276  042524  052123  042040
4085  024304  047117  000105
4086
4087                                  ;INFORMATION FOR MESSAGE 21
4088
4089  024310  044502  040516  054522  EM21:   .ASCIZ  'BINARY DOWN COUNT PATTERN TEST DONE'
4090  024316  042040  053517  020116
4091  024324  047503  047125  020124
4092  024332  040520  052124  051105
4093  024340  020116  042524  052123
4094  024346  042040  047117  000105
4095
4096                                  ;INFORMATION FOR MESSAGE 22
4097
4098  024354  040522  042116  046517  EM22:   .ASCIZ  'RANDOM DATA PATTERN TEST DONE'
4099  024362  042040  052101  020101
4100  024370  040520  052124  051105
4101  024376  020116  042524  052123
4102  024404  042040  047117  000105
4103
4104                                  ;INFORMATION FOR MESSAGE 23
4105
4106  024412  044523  043516  042514  EM23:   .ASCIZ  'SINGLE CHAR PATTERN TEST DONE'
4107  024420  041440  040510  020122
4108  024426  040520  052124  051105
4109  024434  020116  042524  052123
4110  024442  042040  047117  000105
4111
4112                                  ;INFORMATION    FOR MESSAGE 24
4113
4114  024450  054524  042520  020104  EM24:   .ASCIZ  'TYPED BUFFER PATTERN TEST DONE'
4115  024456  052502  043106  051105
4116  024464  050040  052101  042524
4117  024472  047122  052040  051505
4118  024500  020124  047504  042516
4119  024506     000
4120
4121
4122                                  ;INFORMATION FOR MESSAGE 25
4123
4124  024507     104  052101  020101  EM25:   .ASCIZ  'DATA PATTERNS TEST TIMEOUT'
4125  024514  040520  052124  051105
4126  024522  051516  052040  051505
4127  024530  020124  044524  042515
4128  024536  052517  000124
4129  024542  024040  041520  020051  D47:    .ASCIZ  ' (PC)   DEVADR   LINE   CURLPR  ICOUNT  PATCDE'
4130  024550  020040  042504  040526
4131  024556  051104  020040  046040
4132  024564  047111  020105  020040
4133  024572  052503  046122  051120
4134  024600  020040  041511  052517
4135  024606  052116  020040  040520
```

```
4136  024614  041524  042504     000
4137          024622                    .EVEN
4138  024622  001116  021522  022152  DT10:   .WORD   $ERRPC,DHADR,LINE,CURLPR,$REG0,$REG1,0
4139  024630  021576  001162  001164
4140  024636  000000
4141
4142                                    .EVEN
```

```
                                        .SBTTL MISCELANEOUS TABLES AND  MESSAGE AND DATA BUFFERS
                                        ;;*****************************************************************
                                        ;MISCELLANEOUS MESSAGES
                                        ;;*****************************************************************

024640  005015  040515  047111  TITLE:  .ASCIZ  <15><12>'MAINDEC-11-DZDHN-B DH11 DATA RELIABILITY TEST'<15><12>
024646  042504  026503  030461
024654  042055  042132  047110
024662  041055  042040  030510
024670  020061  040504  040524
024676  051040  046105  040511
024704  044502  044514  054524
024712  052040  051505  006524
024720  000012
024722  005015  042524  052123  TITLE2: .ASCIZ  <15><12>'TESTING DH11 #  '<15><12>
024730  047111  020107  044104
024736  030461  021440  020040
024744  005015     000
024747     015  052012  050131  INMSG1: .ASCIZ  <15><12>'TYPE SCR ADDRESS FOR FIRST DH11'<15><12>
024754  020105  041523  020122
024762  042101  051104  051505
024770  020123  047506  020122
024776  044506  051522  020124
025004  044104  030461  005015
025012     000
025013     015  052012  050131  INMSG2: .ASCIZ  <15><12>'TYPE VECTOR ADDRESS FOR FIRST DH11'<15><12>
025020  020105  042526  052103
025026  051117  040440  042104
025034  042522  051523  043040
025042  051117  043040  051111
025050  052123  042040  030510
025056  006461  000012
025062  005015  054524  042520  INMSG3: .ASCIZ  <15><12>'TYPE DH11 DEVICE SELECTION PARAMETER'<15><12>
025070  042040  030510  020061
025076  042504  044526  042503
025104  051440  046105  041505
025112  044524  047117  050040
025120  051101  046501  052105
025126  051105  005015     000
025133     015  044412  053116  INMSG4: .ASCIZ  <15><12>'INVALID DH11 SCR ADDRESS - TRY AGAIN'<15><12>
025140  046101  042111  042040
025146  030510  020061  041523
025154  020122  042101  051104
025162  051505  020123  020055
025170  051124  020131  043501
025176  044501  006516  000012
025204  005015  047111  040526  INMSG5: .ASCIZ  <15><12>'INVALID DH11 VECTOR ADDRESS - TRY AGAIN'<15><12>
025212  044514  020104  044104
025220  030461  053040  041505
025226  047524  020122  042101
025234  051104  051505  020123
025242  020055  051124  020131
025250  043501  044501  006516
025256  000012
025260  005015  047531  020125  INMSG6: .ASCIZ  <15><12>'YOU MUST SELECT AT LEAST ONE DH11'<15><12>
025266  052515  052123  051440
```

```
4199  025274  046105  041505  020124
4200  025302  052101  046040  040505
4201  025310  052123  047440  042516
4202  025316  042040  030510  006461
4203  025324  000012
4204  025326  005015  042504  051120  INMSG7: .ASCIZ  <15><12>'DEPRESS "CONTINUE" TO START TESTING'<15><12>
4205  025334  051505  020123  041442
4206  025342  047117  044524  052516
4207  025350  021105  052040  020117
4208  025356  052123  051101  020124
4209  025364  042524  052123  047111
4210  025372  006507  000012
4211  025376  005015  054524  042520  INMSGA: .ASCIZ  <15><12>'TYPE LINE SELECTION PARAMETER'<15><12>
4212  025404  046040  047111  020105
4213  025412  042523  042514  052103
4214  025420  047511  020116  040520
4215  025426  040522  042515  042524
4216  025434  006522  000012
4217
4218  025440  005015  054524  042520  VCWC:   .ASCIZ  <15><12>'TYPE NO. OF ADDRESSES (OCTAL) BETWEEN VECTORS (10 OR 20)'<15><1
4219  025446  047040  027117  047440
4220  025454  020106  042101  051104
4221  025462  051505  042523  020123
4222  025470  047450  052103  046101
4223  025476  020051  042502  053524
4224  025504  042505  020116  042526
4225  025512  052103  051117  020123
4226  025520  030450  020060  051117
4227  025526  031040  024460  005015
4228  025534     000
4229  025535     015  042012  020110  STMSG1: .ASCIZ  <15><12>'DH #   STATISTICS:'
4230  025542  020043  020040  052123
4231  025550  052101  051511  044524
4232  025556  051503  000072
4233  025562  005015  042524  052123  STMSG2: .ASCIZ  <15><12>'TESTING LINE #  '<15><12>
4234  025570  047111  020107  044514
4235  025576  042516  021440  020040
4236  025604  005015     000
4237  025607     015  046012  047111  STMSG3: .ASCIZ  <15><12>'LINE #   WAS DROPPED'<15><12>
4238  025614  020105  020043  020040
4239  025622  040527  020123  051104
4240  025630  050117  042520  006504
4241  025636  000012
4242  025640  005015  044514  042516  STMSG4: .ASCIZ  <15><12>'LINE #  RTOTAL  XTOTAL  DATERR  PARERR  FRMERR  OVRERR'<15><12>
4243  025646  021440  020040  052122
4244  025654  052117  046101  020040
4245  025662  052130  052117  046101
4246  025670  020040  040504  042524
4247  025676  051122  020040  040520
4248  025704  042522  051122  020040
4249  025712  051106  042515  051122
4250  025720  020040  053117  042522
4251  025726  051122  005015     000
4252                          ;MESSAGES USED BY THE AUTOSIZER.
4253  025733     116  020117  044104  MSG1:   .ASCIZ  /NO DH11'S WERE FOUND/<15><12>
4254  025740  030461  051447  053440
```

```
4255  025746  051105  020105  047506
4256  025754  047125  006504  000012
4257  025762  047516  042040  020110  MSG2:   .ASCIZ  /NO DH RECEIVER INTERRUPT OCCURRED/<15><12>
4258  025770  042522  042503  053111
4259  025776  051105  044440  052116
4260  026004  051105  052522  052120
4261  026012  047440  041503  051125
4262  026020  042522  006504  000012
4263  026026  047516  042040  030515  MSG3:   .ASCIZ  /NO DM11-BB INTERRUPT OCCURRED./<15><12>
4264  026034  026461  041102  044440
4265  026042  052116  051105  052522
4266  026050  052120  047440  041503
4267  026056  051125  042522  027104
4268  026064  005015     000
4269  026067     040  020040     000  SPACE:  .ASCIZ  /  /
4270  026073     015  042012  030510  DEVMAP: .ASCII  <15><12>/DH11, DM11-BB DEVICE MAP:/<15><12>
4271  026100  026061  042040  030515
4272  026106  026461  041102  042040
4273  026114  053105  041511  020105
4274  026122  040515  035120  005015
4275  026130  005015  044104  030461          .ASCII  <15><12>/DH11   DH11   DM11-BB   DM11-BB/
4276  026136  020040  042040  030510
4277  026144  020061  020040  046504
4278  026152  030461  041055  020102
4279  026160  020040  046504  030461
4280  026166  041055     102
4281  026171     015  040412  051104          .ASCIZ  <15><12>/ADRS   VECT    ADRS      VECT/<15><12><15><12>
4282  026176  020123  020040  042526
4283  026204  052103  020040  020040
4284  026212  042101  051522  020040
4285  026220  020040  020040  042526
4286  026226  052103  005015  005015
4287  026234     000
4288                                    ;MESSAGES FOR INPUTTING PARAMETERS TO ECHO TESTS
4289
4290  026235     040  020040  005015  EC:     .ASCIZ  '   '<15><12>
4291  026242     000
4292  026243     040     000          EC2:    .ASCIZ  ' '
4293  026245     040  006440  000012  EC3:    .ASCIZ  '  '<15><12>
4294  026252  005015  044523  043516  ECMSG1: .ASCIZ  <15><12>'SINGLE LINE ECHO TEST - CONNECT TERMINAL TO DH11 TEST LINE'<15>
4295  026260  042514  046040  047111
4296  026266  020105  041505  047510
4297  026274  052040  051505  020124
4298  026302  020055  047503  047116
4299  026310  041505  020124  042524
4300  026316  046522  047111  046101
4301  026324  052040  020117  044104
4302  026332  030461  052040  051505
4303  026340  020124  044514  042516
4304  026346  005015     000
4305
4306
4307  026351     015  052012  050131  ECMSG2: .ASCIZ  <15><12>'TYPE LINE # (00 - 17 OCTAL)'
4308  026356  020105  044514  042516
4309  026364  021440  024040  030060
4310  026372  026440  030440  020067
```

```
4311  026400  041460  040524  024514
4312  026406     000
4313
4314  026407     015  052012  051505  ECMSG3: .ASCIZ  <15><12>'TESTING LINE #   - GO TYPE IN ON TEST LINE'<15><12>
4315  026414  044524  043516  046040
4316  026422  047111  020105  020043
4317  026430  020040  020055  047507
4318  026436  052040  050131  020105
4319  026444  047111  047440  020116
4320  026452  042524  052123  046040
4321  026460  047111  006505  000012
4322
4323  026466  005015  054524  042520  ECMSG4: .ASCIZ  <15><12>'TYPE: [CONTROL-C TO EXIT] [CONTROL-E TO ECHO BUFFER]'<15><12>
4324  026474  020072  041533  047117
4325  026502  051124  046117  041455
4326  026510  052040  020117  054105
4327  026516  052111  020135  041533
4328  026524  047117  051124  046117
4329  026532  042455  052040  020117
4330  026540  041505  047510  041040
4331  026546  043125  042506  056522
4332  026554  005015     000
4333  026557     015  042012  020117  LPMSG:  .ASCIZ  <15><12>'DO YOU WANT TO CHANGE "LPR" (Y OR N) ? '
4334  026564  047531  020125  040527
4335  026572  052116  052040  020117
4336  026600  044103  047101  042507
4337  026606  021040  050114  021122
4338  026614  024040  020131  051117
4339  026622  047040  020051  020077
4340  026630     000
4341  026631     015  052012  040522  XSMSG1: .ASCIZ  <15><12>'TRANSMITTER SPEED ?'
4342  026636  051516  044515  052124
4343  026644  051105  051440  042520
4344  026652  042105  037440     000
4345
4346  026657     015  044412  053116  XSMSG2: .ASCIZ  <15><12>'INVALID XMIT SPEED - TRY AGAIN'<15><12>
4347  026664  046101  042111  054040
4348  026672  044515  020124  050123
4349  026700  042505  020104  020055
4350  026706  051124  020131  043501
4351  026714  044501  006516  000012
4352
4353  026722  005015  042522  042503  RSMSG1: .ASCIZ  <15><12>'RECEIVER SPEED ?'
4354  026730  053111  051105  051440
4355  026736  042520  042105  037440
4356  026744     000
4357
4358  026745     015  044412  053116  RSMSG2: .ASCIZ  <15><12>'INVALID RCVR SPEED - TRY AGAIN'<15><12>
4359  026752  046101  042111  051040
4360  026760  053103  020122  050123
4361  026766  042505  020104  020055
4362  026774  051124  020131  043501
4363  027002  044501  006516  000012
4364
4365  027010  005015  044103  051101  CLMSG1: .ASCIZ  <15><12>'CHAR LENGTH (6, 7, OR 8) ? '
4366  027016  046040  047105  052107
```

```
 4367  027024  020110  033050  020054
 4368  027032  026067  047440  020122
 4369  027040  024470  037440  000040
 4370
 4371  027046  005015  047111  040526  CLMSG2: .ASCIZ  <15><12>'INVALID CHAR LENGTH - TRY AGAIN'<15><12>
 4372  027054  044514  020104  044103
 4373  027062  051101  046040  047105
 4374  027070  052107  020110  020055
 4375  027076  051124  020131  043501
 4376  027104  044501  006516  000012
 4377
 4378  027112  005015  047516  020056  SBMSG1: .ASCIZ  <15><12>'NO. OF STOP BITS (1 OR 2) ? '
 4379  027120  043117  051440  047524
 4380  027126  020120  044502  051524
 4381  027134  024040  020061  051117
 4382  027142  031040  020051  020077
 4383  027150     000
 4384
 4385  027151     015  044412  053116  SBMSG2: .ASCIZ  <15><12>'INVALID NO. STOP BITS - TRY AGAIN'<15><12>
 4386  027156  046101  042111  047040
 4387  027164  027117  051440  047524
 4388  027172  020120  044502  051524
 4389  027200  026440  052040  054522
 4390  027206  040440  040507  047111
 4391  027214  005015     000
 4392
 4393  027217     015  050012  051101  PBMSG1: .ASCIZ  <15><12>'PARITY SELECTION (E, O, OR <CR>) ? '
 4394  027224  052111  020131  042523
 4395  027232  042514  052103  047511
 4396  027240  020116  042450  020054
 4397  027246  026117  047440  020122
 4398  027254  041474  037122  020051
 4399  027262  020077     000
 4400
 4401  027265     015  044412  053116  PBMSG2: .ASCIZ  <15><12>'INVALID PARITY - TRY AGAIN'<15><12>
 4402  027272  046101  042111  050040
 4403  027300  051101  052111  020131
 4404  027306  020055  051124  020131
 4405  027314  043501  044501  006516
 4406  027322  000012
 4407
 4408  027324  005015  044506  046114  FILC1:  .ASCIZ  <15><12>'FILLER CHARACTER ? '
 4409  027332  051105  041440  040510
 4410  027340  040522  052103  051105
 4411  027346  037440  000040
 4412
 4413  027352  005015  044506  046114  FILC2:  .ASCIZ  <15><12>'FILLER COUNT ?'
 4414  027360  051105  041440  052517
 4415  027366  052116  037440     000
 4416  027373     015  051412  047105  SNMSG1: .ASCIZ  <15><12>'SEND MODE -(Y OR N)  '
 4417  027400  020104  047515  042504
 4418  027406  026440  054450  047440
 4419  027414  020122  024516  020040
 4420  027422     000
 4421  027423     015  052012  050131  SNMSG2: .ASCIZ  <15><12>'TYPE SEND BUFFER - TERMINATE WITH CONTROL-C'<15><12>
 4422  027430  020105  042523  042116
```

```
 4423  027436  041040  043125  042506
 4424  027444  020122  020055  042524
 4425  027452  046522  047111  052101
 4426  027460  020105  044527  044124
 4427  027466  041440  047117  051124
 4428  027474  046117  041455  005015
 4429  027502     000
 4430  027503     015  041412  040510  SNMSG3:  .ASCIZ  <15><12>'CHANGE PARAMETERS (Y OR N)? '
 4431  027510  043516  020105  040520
 4432  027516  040522  042515  042524
 4433  027524  051522  024040  020131
 4434  027532  051117  047040  037451
 4435  027540  000040
 4436
```

```
                                        ;MESSAGES FO INPUTTING PATTERNS TEST PARAMETERS
027542  005015  040504  040524  DPMSG1: .ASCIZ  <15><12>'DATA PATTERNS TESTS - CONNECT TEST JUMPER'<15><12>
027550  050040  052101  042524
027556  047122  020123  042524
027564  052123  020123  020055
027572  047503  047116  041505
027600  020124  042524  052123
027606  045040  046525  042520
027614  006522  000012
027620  005015  052502  043106  DPMSG2: .ASCIZ  <15><12>'BUFFER SIZE ? (1 - 512)'
027626  051105  051440  055111
027634  020105  020077  030450
027642  026440  032440  031061
027650  000051
027652  005015  047111  040526  DPMSG3: .ASCIZ  <15><12>'INVALID SIZE - TRY AGAIN'<15><12>
027660  044514  020104  044523
027666  042532  026440  052040
027674  054522  040440  040507
027702  047111  005015    000
027707    015  050012  052101  DPMSG4: .ASCIZ  <15><12>'PATTERN TYPE ? (A,U,D,R,S,9 OR <CR>) '<15><12>
027714  042524  047122  052040
027722  050131  020105  020077
027730  040450  052454  042054
027736  051054  051454  041054
027744  047440  020122  041474
027752  037122  020051  005015
027760    000
027761    015  051412  052105  DPMSG5: .ASCIZ  <15><12>'SET SR07=1 TO LOCK ON PATTERN'<15><12>
027766  051440  030122  036467
027774  020061  047524  046040
030002  041517  020113  047117
030010  050040  052101  042524
030016  047122  005015    000
030023    015  044412  053116  DPMSG6: .ASCIZ  <15><12>'INVALID PATTERN - TRY AGAIN'<15><12>
030030  046101  042111  050040
030036  052101  042524  047122
030044  026440  052040  054522
030052  040440  040507  047111
030060  005015    000
030063    015  052012  050131  DPMSG7: .ASCIZ  <15><12>'TYPE SINGLE TEST CHAR '<15><12>
030070  020105  044523  043516
030076  042514  052040  051505
030104  020124  044103  051101
030112  006440  000012
030116  005015  054524  042520  DPMSGA: .ASCIZ  <15><12>'TYPE IN TEST BUFFER - TERMINATE WITH CONTROL-C'<15><12>
030124  044440  020116  042524
030132  052123  041040  043125
030140  042506  020122  020055
030146  042524  046522  047111
030154  052101  020105  044527
030162  044124  041440  047117
030170  051124  046117  041455
030176  005015    000
030202                  .EVEN
                        ;ERROR STATISTICS TABLES
```

```
030202  000020          RTOTAL: .BLKW   16.          ;TOTAL CHARS RCVD PER LINE
030242  000020          XTOTAL: .BLKW   16.          ;TOTAL CHARS XMITTED PER LINE
030302  000020          DATERR: .BLKW   16.          ;TOTAL DATA COMPARE ERRORS PER LINE
030342  000020          PARERR: .BLKW   16.          ;TOTAL PARITY ERRORS PER LINE
030402  000020          OVRERR: .BLKW   16.          ;TOTAL OVERRUN ERRORS PER LINE
030442  000020          FRMERR: .BLKW   16.          ;TOTAL FRAMING ERRORS PER LINE

                        ;STATISTICS TABLES POINTERS

030502  000000          DEPTR:  0                    ;CONTAINS POINTERS TO STAT TABLES
030504  000000          PEPTR:  0
030506  000000          ORPTR:  0
030510  000000          FRPTR:  0

030512  000000          RBFPTR: 0                    ;CONTAINS INPUT BUFFER POINTER
030514  000000          TBFPTR: 0                    ;CONTAINS OUTPUT BUFFER POINTER

                        ;600. WORD RECEIVER INPUT BUFFER

030516  001130          RBUF:   .BLKW   600.

                        ;600(10) BYTE TRANSMITTER OUTPUT DATA BUFFER

                        .EVEN
032776  001130          TBUF:   .BLKB   600.

034126  000000          ENBUFS: 0                    ;MARK END OF BUFFERS

        000001          .END
```

```
ABASE = 000000            258     299
ACDW1 = 000000            258     301
ACDW2 = 000000            258     302
ACPUOP= 000000            258     273
ADDW0 = 000000            258     303
ADDW1 = 000000            258     304
ADDW10= 000000            258     313
ADDW11= 000000            258     314
ADDW12= 000000            258     315
ADDW13= 000000            258     316
ADDW14= 000000            258     317
ADDW15= 000000            258     318
ADDW2 = 000000            258     305
ADDW3 = 000000            258     306
ADDW4 = 000000            258     307
ADDW5 = 000000            258     308
ADDW6 = 000000            258     309
ADDW7 = 000000            258     310
ADDW8 = 000000            258     311
ADDW9 = 000000            258     312
ADEVCT= 000000            258     264
ADEVM = 000000            258     300
ADPTR   022330            580*    587*    595     979*   3829*
ADRVEC  022142            563    2817*   3757*
AENV  = 000000            258     269
AENVM = 000000            258     270
AFATAL= 000000            258     261
AMADR1= 000000            258     286
AMADR2= 000000            258     290
AMADR3= 000000            258     293
AMADR4= 000000            258     296
AMAMS1= 000000            258     280
AMAMS2= 000000            258     288
AMAMS3= 000000            258     291
AMAMS4= 000000            258     294
AMSGAD= 000000            258     266
AMSGLG= 000000            258     267
AMSGTY= 000000            258     260
AMTYP1= 000000            258     281
AMTYP2= 000000            258     289
AMTYP3= 000000            258     292
AMTYP4= 000000            258     295
APASS = 000000            258     263
APRIOR= 000000            258
APTCSU= 000040           2067    2172*
APTENV= 000001           1818    2060    2128    2170*
APTSIZ= 000200            532    2169*
APTSPO= 000100           2062    2130    2171*
ASWREG= 000000            258     271
ATESTN= 000000            258     262
AUNIT = 000000            258     265
AUSWR = 000000            258     272
AUTOSZ  015152            558    2656*
AVECT1= 000000            258     297
AVECT2= 000000            258     298
BAR   = 000012            647*    718     722     726    1102*   1152*   1461*   1526    1530    1534    3209*   3570*
```

```
BCR    = 000010   737  741  745  1098* 1126* 1137* 1148* 1545 1549 1553 2580* 3206* 3228*
                  3569*
BEGIN    001626    80  489*
BEGINA   001644   493*  997  1212  2872  2877
BIT0   = 000001   179*  556   561   786  2880  2891  2910  2919  2953  2991
BIT00  = 000001   169*  179
BIT01  = 000002   168*  178
BIT02  = 000004   167*  177
BIT03  = 000010   166*  176
BIT04  = 000020   165*  175
BIT05  = 000040   164*  174
BIT06  = 000100   163*  173  1058  1111  1118
BIT07  = 000200   162*  172   815  1612
BIT08  = 000400   161*  171
BIT09  = 001000   160*  170  1765  1829
BIT1   = 000002   178*  857  2818
BIT10  = 002000   159*  680  1488
BIT11  = 004000   158*  659   674   685   703   723   742   795   814   832  1049  1078  1471
                 1480 1493  1511  1531  1550  1592  1611  1629  1772  2568  3202
BIT12  = 010000   157*
BIT13  = 020000   156* 1086  1101  1151  1813
BIT14  = 040000   155*  790  1587  1747
BIT15  = 100000   154*  704   883  1086  1512  1662
BIT2   = 000004   177*
BIT3   = 000010   176*
BIT4   = 000020   175*
BIT5   = 000040   174*
BIT6   = 000100   173*
BIT7   = 000200   172*
BIT8   = 000400   171*  2927
BIT9   = 001000   170*
BKR    = 000014  3571*
BPTVEC= 000014   186*
BRLVL    021716   582  3666*
BRPTR    022334   582*  589*  597   981*  3831*
BUSER    016762   547  3013*
CAR    = 000006   756  765  1099* 1149* 1564  1573  2581* 3207* 3568*
CEXIT    022344  1045* 1071  1105* 3837*
CHKADR   016454  2887  2935*
CHKVCT   016620  2898  2973*
CHPS1    021426   648  1067  1462  3020  3030  3194  3533*
CHPS2    021442   684   702   721   740   763   793   811   831  1044  1492  1510  1529  1548
                 1571  1590  1608  1628  2569  3541*
CHRCNT   021602   635*  757   779   852  1168  1236* 1446* 1565  2576  2580  3108* 3111* 3128
                 3454  3464  3476  3488  3505  3611*
CKER     004112   675   865*
CKERDP   010602  1481  1654*
CKRST1   014574  2534  2549*
CKRST2   014624  2557* 3021  3031
CKSWR  = 104407  1245  1745  1805  1828  2492*
CLALL    021410  3453  3463  3475  3487  3504  3525*
CLMSG1   027010  3328  4365*
CLMSG2   027046  3352  4371*
CLMSK    022370  1663  3515* 3519* 3847*
CLSEL    021604   636* 3101* 3102  3107* 3110  3613*
CLSTAT   017136   609  3062*
```

```
CR     = 000015            94*   2106   2116
CRLF   = 000200            95*   2077   2116
CURLPR   021576          1059    2579   3077   3079*  3082   3109*  3110*  3152*  3153*  3208   3236*  3243*  3255*
                         3265*   3272*  3284*  3311   3313*  3329*  3343*  3347*  3354*  3375*  3382*  3390*  3403*
                         3407*   3514   3608*  3879   3973   4000   4054   4075   4138
DATCNT   022350          1296*   1299*  1300   1303   1304*  1314*  1317*  1318   1321   1322*  1332*  1335*  1336
                         1339    1340*  1350*  1353*  1354   1357   1358*  1381*  1399*  1400   1403   1404*  1411*
                         1448*   1449   1452   1453*  1472   3839*
DATERR   030302           867     964   4495*
DATPAT   022352          1255*   1268   1272   1276   1280   1284   1288   1368*  1370*  1372*  1374*  1473   3840*
DDISP  = 177570           101*    222    520
DEPTR    030502           867*    873*   917*  4502*
DEVMAP   026073          2821    4270*
DF1      023602           414    4008*
DF2      022604           344     351    358    365    372    379    386    393    400    407    421    428    435
                          442     449    456    463    470    477    484   3882*
DHADR    021522           595*    620    659*   999*  1048   1214*  1248   1471*  2608*  2946*  2950*  3201   3574*
                         4075    4138
DHADRS   021756          2658    2666   2713   2729   2822   2879   2955   3685*
DHADTB   021614           580    2952   2966   3622*
DHFILL   022434          3420*   3432*  3433   3444*  3446*  3448*  3449   3850*
DHNUM    022150           583*    586*   599    938    978*  3761*
DHRLVL   022146           597*    616   1047*  1052   1238*  1241   3759*
DHSEL    021530           590     985   2917*  3577*
DHSET1   014654           646    1460   2568*
DHST1    002700           646*
DHST2    007526          1298    1316   1334   1352   1398   1447   1460*
DHTLVL   022147           619    1055   1244   3760*
DHVCT    021524           596*    614   1000*  1050   1215*  1239   2984*  2988*  3036   3575*
DHVCTB   021654           581    2990   3004   3642*
DHVEC    022014          2730    2811   2812   2814   2823   2890   2993   3704*
DHWAIT   005252          1069*
DH1      022511           342     349    356    363    370    377    384   3870*
DH2      023305           391     398   3963*
DH3      023500           405    3992*
DH4      023637           419     426   4018*
DH5      024022           433    4045*
DH6      024162           440     447    454    461    468    475   4067*
DH7      024542           482    4129*
DISPLA   001142           222*    520*   528*  1787*  1808*
DISPRE   000174            75*    528
DMADRS   022050          2665    2771   2824   3722*
DMVEC    022106          2772    2825   3741*
DPATA    006474          1270    1296*  1369
DPATB    007306          1290    1411*
DPATCR   007064          1266    1368*  1377
DPATD    006670          1278    1332*  1373
DPATR    006766          1282    1350*  1375
DPATS    007146          1286    1380*
DPATU    006572          1274    1314*  1371
DPFLG    022346           491*    570    995*  1021   1209*  2549   2557   2867*  2875*  2904   2948   2986   3838*
DPMSGA   030116          1413    4481*
DPMSG1   027542          1217    4438*
DPMSG2   027620          1223    4446*
DPMSG3   027652          1231    4451*
DPMSG4   027707          1250    4456*
```

```
DPMSG5  027761          1262    4464#
DPMSG6  030023          1293    4470#
DPMSG7  030063          1382    4476#
DRPLIN  021536           612*    669*    696*    715*    734*    753*    773*    805*    825*    842*    945    3580#
DSWR  = 177570           100#    221     519
DT1     022566           343     350     357     364     371     378     385    3879#
DT10    024622           483    4138#
DT2     023372           392     399    3973#
DT3     023546           406    4000#
DT4     023562           413    4005#
DT5     023706           420     427    4026#
DT6     024102           434    4054#
DT7     024232           441     448     455     462     469     476    4075#
EC      026235          4290#
ECBUF   022374          1124*   1125*   1133*   1134*   1146*   1149    3218    3219*   3849#
ECHO    004706            82     994#   2555    2563
ECHO1   004730           569     999#
ECHO2   004754          1007#   1081    1202    1220    2903
ECHO3   005124          1037    1039    1044#
ECMSG1  026252          1002    4294#
ECMSG2  026351          1008    4307#
ECMSG3  026407          1062    1064    4314#
ECMSG4  026466          1066    4323#
EC2     026243          1025*   1032*   1034    1036    1038    1040    1162*   1164    1184*   1191*   1193    1196    1198
                        1200    1247*   1256*   1258    1263    1380*   1389*   1391    1394    1417*   1419    3292*   3299*
                        3301    3304    3306    3308    3326*   3334*   3336    3338    3340*   3341    3345    3349    3359*
                        3366*   3368    3370    3372*   3373    3377    3387*   3395*   3397    3399    3401    3405    3416*
                        3425*   3427    3430    3432    4292#
EC3     026245          1440*   1441*   1443    4293#
EMTVEC= 000030           189#    504*    505*
EM1     022442           341     692    1500    3863#
EM10    023235           390     838    1635    3956#
EM11    023412           397     898    1677    3979#
EM12    023440           404     666    3986#
EM14    023612           418    4014#
EM15    023722           425    4031#
EM16    023750           432    4038#
EM17    024120           439    4061#
EM2     022614           348     711    1519    3888#
EM20    024246           446    4080#
EM21    024310           453    4089#
EM22    024354           460    4098#
EM23    024412           467    4106#
EM24    024450           474    4114#
EM25    024507           481    4124#
EM3     022673           355     730    1538    3900#
EM4     022753           362     749    1557    3912#
EM5     023030           369     769    1577    3923#
EM6     023112           376     801    1598    3935#
EM7     023161           383     821    1618    3945#
ENBUFS  034126          3527    4521#
ENDA    004616           972     975#    986
ERRVEC= 000004           182#    517     518*    529*    547*    548*   1752    1753*   1755*   1758*
EXFLAG  022372          1046*   1087    1089    1091    1095*   1109*   1127*   1139*   3848#   4054
EXPAR   016450          2928    2932#
EXPAT   006106            83    1209#   2554    2562
```

```
EXPAT1  006130            572    1214*
EXPAT2  006154           1220*   2906
EXPAT3  006260           1245*   1294    1312    1330    1348    1366    1378    1409    1458    1478    1505    1524    1543
                         1562    1582    1603    1623    1640
FILC1   027324           3419    4408*
FILC2   027352           3440    4413*
FILLA   022436           1175    1432    3221    3222    3417*   3433*   3852*
FILLB   022440           1174    1430    3217    3225    3438*   3449*   3853*
FRMERR  030442            870     967    4498*
FRPTR   030510            870*    876*    925*   4505*
GNS   = ****** U           74    2484    2485    2486    2487    2488    2490    2492    2493    2494    2495    2496
GTSWR = 104406            543    2490*
HT    = 000011             92*   2075    2116
INCL    020372           3317    3326*   3353
INFCHR  020774           3320    3416*
INFCNT  021064           3321    3438*
INMSGA  025376           2922    4211*
INMSG1  024747           2883    4161*
INMSG2  025013           2894    4168*
INMSG3  025062           2913    4175*
INMSG4  025133           2970    4182*
INMSG5  025204           3008    4189*
INMSG6  025260           4197*
INMSG7  025326           2930    4204*
INPAR   016224            578    1004    1218    2879*
INPARA  016120            565    2856*   2865
INPARC  016200             81    2873*
INPARX  016166             78    2871*
INPAR1  016260           2886    2890*
INPAR3  016314            577    2897    2901*
INPAR4  016376           2916    2918*
INPB    020654           3319    3387*   3411
INRSP   020116           3262*   3316
INSB    020536           3318    3359*   3381
INXSP   020002           3233*   3315
IOTVEC= 000020            187*    502*    503*   2731*   2732*   2773*   2799*
KYBD1   017522            610    3170*
KYBD2   017546           3170    3178*
LDFILL  017726           1143    3217*
LF    = 000012             93*   2110    2116
LINE    022152            613*    627     660     665     686     691     705     710     729     748     768     791     797
                          800     816     820     837     854     871     881     897    1010*   1011*   1012    1056    1061
                         1494    1499    1513    1518    1537    1556    1576    1594    1597    1613    1617    1634    1660
                         1676    1746*   2572    2594    2596*   2598*   2600*   2607*   3203    3762*   4054    4075    4138
LINMSK  021534            647     669     696     715     734     753     773     805     825     842     934*    945     958
                          971*   1014*   1017*   1102    1152    1461    2597*   2601*   2603    3209    3579*
LINSEL  021532            958    2603    2918*   2926*   3578*
LPMSG   026557           3294    4333*
LPR   = 000004           1059*   2579*   3208*   3567*
LPRIN   020232           1019    3292*   3309
LPRPTR  021600            631*   3077    3079    3087*   3610*
LPRTAB  021544            631    3594*
MSG1    025733           2715    4253*
MSG2    025762           2755    4257*
MSG3    026026           2784    4263*
NEWCL   002660            639*    644
```

```
NEWLIN  002574  622* 634 671 697 716 735 754 774 806 826 843
NEWLPR  002632  633* 640
NEWPAR  002672  643* 676
NRC   = 000002  828 1119 1625 3566*
ORPTR   030506  869* 875* 922* 4504*
OVRERR  030402  869 966 4497*
PARBIT  021606  641* 3147 3151* 3153 3154 3156* 3158 3160* 3162 3164* 3614*
PARERR  030342  868 965 4496*
PATFLG  022354 1246* 1265* 1307 1325 1343 1361 3841*
PATLIM  022362 1300 1318 1336 1354 1400 1449 3844*
PBMSG1  027217 3389 4393*
PBMSG2  027265 3410 4401*
PC    =%000007  113* 558* 565* 609* 610* 622* 646* 648* 655* 675* 684* 689* 702*
                708* 721* 727* 740* 746* 763* 766* 793* 798* 811* 818* 831* 835*
                894* 895* 906* 913* 918* 929* 1019* 1023* 1044* 1067* 1073* 1143* 1183*
               1220* 1221* 1297* 1298* 1309* 1315* 1316* 1327* 1333* 1334* 1345* 1351* 1352*
               1363* 1369* 1371* 1373* 1375* 1397* 1398* 1447* 1460* 1462* 1468* 1481* 1482*
               1492* 1497* 1510* 1516* 1529* 1535* 1548* 1554* 1571* 1574* 1590* 1595* 1608*
               1615* 1628* 1632* 1674* 1691* 1707* 1710* 1720* 1725 1815* 1821* 1890* 2065*
               2084* 2091* 2098* 2112* 2114* 2147* 2164* 2227* 2533 2569* 2582* 2609* 2651*
               2851* 2869* 2887* 2898* 2971* 3009* 3020* 3030* 3043* 3056* 3066* 3112* 3137*
               3173* 3194* 3213* 3229* 3258* 3287* 3315* 3316* 3317* 3318* 3319* 3320* 3321*
               3322* 3355* 3383* 3412* 3434* 3450* 3453* 3459* 3463* 3471* 3475* 3483* 3487*
               3500* 3504* 3510* 3522* 3529* 3536* 3544* 3558*
PEPTR   030504  868* 874* 928* 4503*
PIRQ  = 177772  99*
PIRQVE= 000240  193*
PRSTAT  004426  625 934* 3195
PR0   = 000000  116*
PR1   = 000040  117*
PR2   = 000100  118*
PR3   = 000140  119*
PR4   = 000200  120*
PR5   = 000240  121*
PR6   = 000300  122*
PR7   = 000340  123*
PS    = 177776  96* 97 2747* 2777*
PSW   = 177776  97*
PTWAIT  007544 1464*
PWRVEC= 000024  188* 508* 509* 2501* 2502* 2511* 2517* 2529* 2530*
QUICK   021540  611* 3084* 3086* 3104 3149 3582*
QUICKX  021542  638* 3099 3106* 3584*
RANA    022364 3490* 3492 3493* 3494* 3497 3845*
RANB    022366 3492* 3494 3495* 3846*
RBFEND  021612  850 910 1644 1647 1685 1688 2575* 2578* 3617*
RBFPTR  030512  833* 834 848* 849* 850 865* 884 890 909* 910 1630* 1631 1642*
               1643* 1644 1646* 1647 1654* 1664 1670 1684* 1685 1687* 1688 2574* 4507*
RBUF    030516  833 865 1630 1654 2574 2575 4513*
RDCHR = 104410 1028 1160 1187 1251 1384 1415 2287 2493* 3295 3330 3362 3391 3421
RDDEC = 104413 1224 2496* 3237 3266
RDLIN = 104411 2359 2412 2494*
RDOCT = 104412 1009 2495* 2858 2884 2895 2914 2923 3441
RDONE   021610  653 786* 857* 1466 1649* 2573* 3616* 4000
RESERR  017024  549 3023*
RESTRP  017066 3036*
RESTRT  002374  586* 594
```

```
REST1   002432   592*  2561
RESVEC= 000010   183*   549*   550*
RETFLG  022360   492*   567    994*  1210*  2551  2559  2868*  2876*  2901  2944  2982  3843*
RINT1   003550   615    790*
RINT2   005476  1051   1118*
RINT3   010266  1240   1587*
RSMSG1  026722  3263   4353*
RSMSG2  026745  3280   4358*
RSPTAB  022244  3264   3277   3800*
RSPTR   022242  3264*  3269   3271*  3272  3276*  3277  3798*
RSTRTA  002442   591    595*   987   2553
RTOTAL  030202   662    856*   962   3062  3064  4493*
R0      =%000000   104*   489*   563*   573   575   660*   661*   662   663   892*   893*  1029*  1030
                  1032  1050*  1051*  1052*  1053  1054*  1055*  1161*  1162  1165  1170  1173  1180
                  1188*  1189   1191  1225*  1228  1234*  1235*  1236  1239*  1240*  1241*  1242  1243*
                  1244*  1252*  1253  1255  1256  1385*  1386  1388  1389  1430*  1433*  1672*  1673*
                  1717*  1720  1846  1847*  1848*  1855*  1856*  1857*  1858*  1859*  1860  1865  1871
                  1873*  1874  1888*  1983  1993*  1997  2013  2014  2027*  2058  2059*  2064  2069
                  2072*  2124  2132*  2136  2137  2139*  2140*  2141  2163*  2356  2360*  2361  2364
                  2384*  2387*  2409  2413*  2414  2417  2419  2420  2444*  2448*  2461  2462*  2463
                  2464*  2465*  2466*  2467*  2503  2528*  2622  2625*  2627*  2628  2635*  2636*  2637
                  2640*  2646  2656  2719*  2720*  2721*  2725  2751*  2753*  2780*  2782*  2800*  2859*
                  2861  2863  2866*  2873*  2908  3003  3296*  3297  3299  3331*  3332  3334  3363*
                  3364  3366  3392*  3393  3395  3422*  3423  3425  3454*  3457*  3464*  3469*  3476*
                  3481*  3488*  3498*  3505*  3508*  3514*  3516*  3517  3520*  3525*  3526*  3527
R1      =%000001   105*   620*   647*   674*   680   683   685*   688   699   703*   707   718   722
                  723*   725   737   741   742*   744   756   764   790   794   795*   808   812
                  814*   817   828   832*   889   890*   902*   935*   948   956   960   970*  1048*
                  1049*  1056*  1058*  1059*  1076  1077  1078*  1086*  1098*  1099*  1101*  1102*  1111*
                  1118*  1119  1126*  1137*  1148*  1149*  1151*  1152*  1248*  1461*  1480*  1488  1491
                  1493*  1496  1507  1511*  1515  1526  1530  1531*  1533  1545  1549  1550*  1552
                  1564  1572  1587  1591  1592*  1605  1609  1611*  1614  1625  1629*  1669  1670*
                  1681*  1870  1871*  1876  1880  1882  1887*  1984  1997*  1998  2002  2026*  2125
                  2162*  2357  2362*  2370*  2372*  2374*  2377*  2380  2383*  2410  2420*  2422  2424
                  2434*  2435  2443*  2504  2527*  2568*  2570*  2572*  2579*  2580*  2581*  2623  2626*
                  2634*  2637*  2639*  2647  2668*  2670  2671  2672*  2674*  2675*  2676  2679*  2681
                  2685*  2686  2689  2691*  2700*  2701  2702  2704  2706*  2760*  2761*  2762  2789*
                  2790*  2791  2811*  2814*  2816*  2817  2822*  2828  2848  2879*  2885*  2890*  2896*
                  2907*  2915*  2917  2924*  2926  2935  2938  2941  2946  2950  2964  2965*  2973
                  2976  2979  2984  2988  3002  3003*  3201*  3202*  3203*  3206*  3207*  3208*  3209*
                  3211  3228*
R2      =%000002   106*   614*   615*   616*   617   618*   619*   688*   707*   725*   726*   744*   745*
                  764*   765*   817*   834*   891*   892   956*   957*   962   963   964   965   966
                  967  1012*  1013*  1015*  1057*  1093  1096*  1097*  1098  1108*  1135*  1136*  1147*
                  1496*  1515*  1533*  1534*  1552*  1553*  1572*  1573*  1614*  1631*  1671*  1672  1985
                  1996*  2000*  2003  2010*  2011*  2012  2017*  2025*  2358  2363*  2371*  2373*  2375*
                  2381  2382*  2411  2416*  2419*  2438  2442*  2505  2526*  2624  2628*  2629*  2630*
                  2631*  2632*  2633*  2634  2638*  2648  2658*  2659*  2666*  2686*  2713*  2719*  2722*
                  2723  2729*  2744  2748*  2749*  2756*  2757*  2765*  2766*  2771*  2775  2778*  2779*
                  2785*  2786*  2794*  2795*  2823*  2832  2952*  2958*  2964*  2966  2990*  2996*  3002*
                  3004  3124  3129*  3130*  3136*  3220*  3222*
R3      =%000003   107*   683*   687*   699*   706*   722*   741*   756*   760   777   779*   780*   783
                  785*   794*   796*   812*   813*   846   852*   853*   856   860*   884*   885   888
                  915  1491*  1495*  1507*  1514*  1530*  1549*  1564*  1568  1591*  1593*  1609*  1610*
                  1664*  1665  1668  1926  1935*  1941*  1942*  1945*  1950*  1951*  1952  1961*  1986
                  1994*  1995*  2009*  2012*  2021*  2022*  2024*  2282  2284*  2295  2298*  2299  2296*
```

```
                2297    2299    2307    2311    2313*   2319    2321    2323*   2326*   2506    2525*   2649    2657*
                2660*   2661    2665*   2702*   2733*   2736*   2739    2801*   2804*   2807    2824*   2837    2955*
                2958    2993*   2996    3036*   3037*   3038*   3039*   3040*   3041*   3042*   3125    3128*   3132*
                3135*   3218*   3221*
R4      =%000004  108#    686*    704*    705*    724*    743*    757*    758*    759*    760     778     781*    782*
                 783*    784*    797*    815*    816*    847     854*    855*    856*    859*    878*    879*    880*
                 881*    882*    883*    885     915     1174*   1176*   1494*   1512*   1513*   1532*   1551*   1565*
                1566*   1567*   1568    1594*   1612*   1613*   1657*   1658*   1659*   1660*   1661*   1662*   1663*
                1665    1927    1929*   1930*   1931*   1932    1933*   1947    1949*   1957*   1960*   2507    2524*
                2650    2734*   2736    2738*   2739    2740    2802*   2804    2806*   2807    2808    2825*   2842
                2957*   2960*   2961    2995*   2998*   2999    3126    3127*   3130    3131*   3134*   3217*   3223*
                3225*   3226*   3227*   3228    3466*   3467    3468*   3478*   3479    3480*
R5      =%000005  109#    598*    626*    633*    639*    643*    664*    690*    709*    728*    747*    767*    799*
                 819*    836*    871*    872*    873     874     875     876     896*    936*    949*    1060*   1159*
                1173*   1175*   1180*   1414*   1424    1431*   1432*   1437*   1444*   1445*   1446    1498*   1517*
                1536*   1555*   1575*   1596*   1616*   1633*   1675*   1928    1934*   1936*   1938*   1939*   1940*
                1941    1959*   1987    1989*   1991*   1998*   2002*   2017    2023*   2508    2523*   2576*   2577*
                2578    2625    2626    2641*   2730*   2762*   2772*   2791*   2826*   3062*   3063*   3064    3088*
                3089*   3113*   3114*   3165*   3166*   3204*   3205*   3206    3455*   3456*   3465*   3467*   3477*
                3479*   3489*   3497*   3506*   3507*
R6      =%000006  110#    496*    497*    498
R7      =%000007  111#
SAPS      021456  3549#
SBMSG1    027112  3361    4378#
SBMSG2    027151  3380    4385#
SELINE    014752   622    2594#
SELMSK    021526   584*    590     592*    982*    985     3576#
SENDP1    005720  1042    1157#   1197    1199
SENDP2    017650  1183    3201#
SETCL     017240   639    3099#
SETLPR    017154   633    3077#
SETPAR    017400   643    3147#
SINGLE    022356  1388*   1396*   1416*   1417    1421    1427    1431    1437    3507    3842#
SNMSG1    027373  1027    4416#
SNMSG2    027423  1158    4421#
SNMSG3    027503  1185    4430#
SOFT      004366   906     920#
SP      =%000036  112#    500*    517*    525*    529     670*    695     714     733     752     772     777*    778*
                 784     785     804     824     841     846*    847*    859     860     889*    902     1010    1029
                1161    1188    1225    1252    1385    1416    1477    1503    1504    1522    1523    1541    1542
                1560    1561    1580    1581    1601    1602    1621    1622    1638    1639    1669*   1681    1714*
                1752*   1755    1757    1758    1783    1784    1788*   1810    1831*   1834*   1846*   1851*   1870*
                1876*   1880*   1887    1888    1918*   1919    1920    1921*   1926*   1927*   1928*   1934    1959
                1960    1961    1962*   1963*   1983*   1984*   1985*   1986*   1987*   1988*   1989    1992*   2005
                2007*   2009    2019    2021    2023    2024    2025    2026    2027    2029*   2030*   2058*   2059
                2069*   2071    2072    2073*   2075    2077    2079    2085    2087*   2089*   2097*   2101    2105
                2106    2110    2124*   2125*   2132    2133*   2144    2145*   2146*   2156    2157*   2162    2163
                2187*   2188*   2189    2196*   2199*   2200*   2204*   2205*   2209    2212*   2216    2219    2220
                2221*   2228    2230    2232*   2233    2235*   2236*   2237*   2238*   2239*   2254*   2255*   2258*
                2259*   2260    2264*   2265*   2266    2269    2271    2273*   2282*   2283*   2288    2291    2295*
                2302    2306*   2325    2326    2327*   2328*   2329*   2354*   2355*   2356*   2357*   2358*   2360
                2364*   2366    2368    2376*   2377    2379    2380*   2382    2383    2384    2386    2407*   2408*
                2409*   2410*   2411*   2413    2415*   2426    2428*   2429*   2430*   2431*   2432*   2435*   2440*
                2441*   2442    2443    2444    2447    2461*   2462    2472*   2473*   2503*   2504*   2505*   2506*
                2507*   2508*   2509*   2510    2518*   2522    2523    2524    2525    2526    2527    2528    2533*
                2605*   2622*   2623*   2624*   2638    2639    2640    2656*   2663*   2694*   2709*   2712    2760
```

```
                    2763    2764*   2789    2792    2793*   2800    2828*   2832*   2837*   2842*   2859    2885    2896
                    2915    2924    2943*   2981*   3013    3014    3015    3016*   3023    3024    3025    3026*   3055*
                    3124*   3125*   3126*   3134    3135    3136    3178*   3179*   3180    3186    3188    3191    3193*
                    3196    3238    3240    3244    3252    3256    3267    3269    3273    3281    3285    3296    3331
                    3363    3392    3422    3442    3444    3447    3533*   3534*   3541*   3542*   3549*   3550*   3553*
                    3554*   3556    3557
SPACE   026067      2836    2841    4269#
SSR   = 000016      3572#
STACK = 001100       87#     500     670    3016    3026    3193
START1  002152       547#
START2  002336       574     580#    593    1726    2932
STDH1   002516       609#
STKLMT= 177774        98#
STMSG1  025535       939     941    4229#
STMSG2  025562       628     630    4233#
STMSG3  025607       951     953    4237#
STMSG4  025640       943    4242#
SUBUF1  017340      3112    3123#
SUCLMK  021354      1221    3514#
SUER1   015130       689     708     727     746     766     798     818    1497    1516    1535    1554    1574    1595
                    1615    2647#
SUER2   015124       835     895    1632    1674    2646#
SUNUM   015044       598     626     664     690     709     728     747     767     799     819     836     896     936
                     949    1060    1498    1517    1536    1555    1575    1596    1616    1633    1675    2621#
SUPATA  021136      1297    3453#
SUPATD  021214      1333    3475#
SUPATR  021246      1351    3487#
SUPATS  021326      1397    3504#
SUPATU  021164      1315    3463#
SWR     001140       221#    498     519*    521     527*    534*    541     556     561    1310    1328    1346    1364
                    1376    1407    1456    1747    1765    1772    1813    1825    1829    2183    2220*   2509    2522*
                    2818    2880    2891    2910    2919    2927    2953    2991    3080    3182
SWREG   000176       76#     527     541    2183    2196    3182
SW0   = 000001      151#
SW00  = 000001      141#     151
SW01  = 000002      140#     150
SW02  = 000004      139#     149
SW03  = 000010      138#     148
SW04  = 000020      137#     147
SW05  = 000040      136#     146
SW06  = 000100      135#     145
SW07  = 000200      134#     144
SW08  = 000400      133#     143
SW09  = 001000      132#     142
SW1   = 000002      150#
SW10  = 002000      131#
SW11  = 004000      130#
SW12  = 010000      129#
SW13  = 020000      128#
SW14  = 040000      127#
SW15  = 100000      126#
SW2   = 000004      149#
SW3   = 000010      148#
SW4   = 000020      147#
SW5   = 000040      146#
SW6   = 000100      145#
```

```
SW7    = 000200            144#
SW8    = 000400            143#
SW9    = 001000            142#
TBFPTR  030514             866*   878    891    908*   1655*  1657   1671   1683*  4508#
TBITVE= 000014             184#
TBUF    032776             759    866    893    1057   1093   1096   1099   1108   1159   1414   1424   1444   1567
                          1655   1673   2581   3129   3204   3207   3455   3465   3477   3489   3506   3525   4519*
TIMEA   022340             651*   1069*  1100*  1110*  1150*  1464*  3053*  3834#
TIMEB   022342             652*   1070*  1465*  3051*  3835#
TIMEIT  017114             655    1073   1468   3051#
TINT1   003044             618    680#
TINT2   005330            1054   1086#
TINT3   007644            1243   1488#
TITFLG  022336             493*   551    555*   3833#
TITLE   024640             554   4148#
TITLE2  024722             600    602   4157#
TKVEC = 000060             191#
TPVEC = 000064             192#
TRAPVE= 000034             190#   506*   507*
TRTVEC= 000014             185#
TST1    002506             607#
TYPDS = 104405            1715   1881   2488#
TYPE  = 104401             553    601    629    940    942    952   1001   1007   1026   1033   1063   1065   1157
                          1163   1185   1192   1216   1222   1230   1249   1257   1261   1292   1382   1390   1412
                          1418   1442   1713   1716   1816   1845   1862   1864   1867   1869   1884   1889   1953
                          2028   2080   2194   2195   2198   2211   2222   2241   2294   2300   2305   2309   2314
                          2315   2317   2320   2324   2388   2390   2449   2451   2484*  2531   2715   2755   2784
                          2820   2836   2841   2846   2850   2856   2882   2893   2912   2921   2929   2969   3007
                          3189   3233   3250   3262   3279   3293   3300   3327   3335   3351   3360   3367   3379
                          3388   3396   3409   3418   3426   3439
TYPOC = 104402            1853   1877   2197   2485#
TYPON = 104404            2487#
TYPOS = 104403            2486#  2829   2833   2838   2843
UPDER   004354             894    915#
VCFLG   021714             490*   559    566*   996*   1211*  2871*  2874*  3659#
VCPTR   022332             581*   588*   596    980*   3830#
VCWC    025440            2857   4218#
XSMSG1  026631            3234   4341#
XSMSG2  026657            3251   4346#
XSPTAB  022156            3235   3248   3772#
XSPTR   022154            3235*  3240   3242*  3243   3247*  3248   3770#
XTOTAL  030242             663    783*   963   4494#
$APTHD  000000              55     61#
$ASTAT= ****** U          2150   2165
$ATYC   012664            2121   2123#
$ATY1   012640            2119#
$ATY3   012646            2065   2120#
$ATY4   012656            1821   2122#
$AUTOB  001134             218#   545*  2191   2340   3184
$BASE   001306             299#
$BDADR  001122             213#
$BDDAT  001126             215#
$CDW1   001312             301#
$CDW2   001314             302#
$CHARC  012634            2082*  2092*  2099   2108*  2113#
$CKSWR  013106            2183#  2492
```

```
$CMTAG  001100           201*   495    496    504    510    511
$CM1   = 000010          233*   234*   235*   236*   237*   238*   239*   240*   241*
$CM2   = 000020          233*   234*   235*   236*   237*   238*   239*   240*   241*
$CM3   = 000010          231*   233
$CM4   = 000010          241*   242*   243*   244*   245*   246*   247*   248*   249*
$CNTLG  013761          2194   2335*  3189
$CNTLU  013754          2211   2309   2334*
$CPUOP  001260           273*
$CRLF   001227           252*   1816   1837   1845   1864   1869   1889   2081   2116   2222   2314   2334   2393
                        2453   2847   2850
$DBLK   012346          1994   2028   2036*
$DDW0   001316           303*
$DDW1   001320           304*
$DDW10  001342           313*
$DDW11  001344           314*
$DDW12  001346           315*
$DDW13  001350           316*
$DDW14  001352           317*
$DDW15  001354           318*
$DDW2   001322           305*
$DDW3   001324           306*
$DDW4   001326           307*
$DDW5   001330           308*
$DDW6   001332           309*
$DDW7   001334           310*
$DDW8   001336           311*
$DDW9   001340           312*
$DEVCT  001242           264*
$DEVM   001310           300*
$DHSEL  022144          2725*  2907   2957   2995   3758*
$DOAGN  011066          1709   1718   1724*
$DTBL   012336          1997   2032*
$ENDAD  011056          1720*
$ENDCT  011024          1711*
$ENDMG  011075          1713   1728*
$ENULL  011072          1716   1727*
$ENV    001252           269*   539    1818   2060   2128   2152
$ENVM   001253           270*   532    2062   2067   2130
$EOP    010770           977*   984    1701*
$EOPCT  011016          1708*  1712
$ERFLG  001103           204*  1736   1761   1763   1769*  1791   1806*  1837
$ERMAX  001115           210*   512*   1763   1786*  1791
$ERROR  011356           504    1804*
$ERRPC  001116           211*  1810*  1811*  1812   1837   1851   3879   3973   4000   4026   4054   4073   4138
$ERRTB  001356           337*  1859
$ERRTY  011530          1815   1844*
$ERTTL  001112           208*  1809*  1837
$ESCAP  001224           250*   511*  1785*  1832   1834   1837
$ETABL  001252           268*
$ETEND  001356            67    321*
$FATAL  001234           261*  2156*
$FFLG   013104          2119*  2122*  2150   2159*  2167*
$FILLC  001156           229*  2085   2116
$FILLS  001155           228*  2116
$GDADR  001120           212*
$GDDAT  001124           214*
```

```
$GET42  011046          1717#
$GTSWR  013156          2195#   2490    3190
$HD   = 000001            26      27
$HIBTS  000000            62#
$HIOCT  014146          2381*   2392#
$ICNT   001104           205#   1776*   1777    1779*   1790
$ILLUP  014556          2501    2517    2536#
$INTAG  001135           219#   2223    2340
$ITEMB  001114           209#   1812*   1820    1837    1848
$LF     001230           253#   1837    2116    2324    2334    2393    2453
$LFLG   013103          2160*   2166#
$LPADR  001106           206#    513*    603*   1757*   1783*   1788    1790
$LPERR  001110           207#    514*    667*    693*    712*    731*    750*    770*    802*    822*    839*    899*    968*
                        1079*   1305*   1323*   1341*   1359*   1405*   1454*   1474*   1501*   1520*   1539*   1558*   1578*
                        1599*   1619*   1636*   1678*   1767    1784*   1790    1831    3017*   3027*
$MADR1  001264           286#
$MADR2  001270           290#
$MADR3  001274           293#
$MADR4  001300           296#
$MAIL   001232            63      67     259#    531     539    1782    1818    2060
$MAMS1  001262           280#
$MAMS2  001266           288#
$MAMS3  001272           291#
$MAMS4  001276           294#
$MBADR  000002            63#
$MFLG   013102          2120*   2126    2161*   2165#
$MNEW   013777          2198    2338#
$MSGAD  001246           266#   2136*   2139
$MSGLG  001250           267#   2141*
$MSGTY  001232           260#   2134    2142*   2154    2158*
$MSWR   013766          2195    2336#
$MTYP1  001263           281#
$MTYP2  001267           289#
$MTYP3  001273           292#
$MTYP4  001277           295#
$MXCNT  011354          1780    1790#
$NULL   001154           227#   2087    2116
$NWTST= 000001           604#
$OCNT   012126          1925*   1954*   1967#
$OMODE  012130          1920*   1924*   1929    1932*   1943*   1969#
$OVER   011340          1748    1768    1778    1787#
$PASS   001240           263#    531*   1705*   1706*   1714    1727    1774    1791
$PASTM  000006            65#
$POWER  014564          2532    2539#
$PWRAD  014552          2534#
$PWRDN  014412           508    2501#   2529
$PWRMG  014546          2532#
$PWRUP  014464          2511    2517#
$QUES   001226           251#   1837    2116    2241    2317    2334    2390    2393    2451    2453
$RDCHR  013370          2254#   2493
$RDDEC  014150          2407#   2496
$RDLIN  013510          2282#   2494
$RDOCT  014010          2354#   2495
$RDSZ = 000010          2275#
$REGAD  001160           231#
$REGO   001162           233#   1303*   1321*   1339*   1357*   1403*   1452*   1472*   2646*   3973    4075    4138
```

N12


```
$REG1   001164          234#  1076*  1473*  2647*  3014*  3024*  3879   3973   4026   4054   4138
$REG2   001166          235#  2648*  3015*  3025*  3879   3973   4026
$REG3   001170          236#  2649*  3879   3973
$REG4   001172          237#  2650*  3879   3973
$REG5   001174          238#
$REG6   001176          239#  3013*  3023*  4026
$REG7   001200          240#
$RTNAD  011070         1726#
$R2A  = ******  U       2497
$SAVRE= ******  U       2497
$SAVR6  014562         2510*  2518   2519*  2520*  2538#
$SCOPE  011112          502   1744#  2799
$SETUP= 000117          494#   501    502    504    506    508    510    511    513    536   1703   1745   1805
                       1828   1836   2178   2340
$STUP = 177777          494#
$SVLAD  011304         1756   1781#
$SVPC = 000000           40#    45
$SWR  = 165000            8#    26     31     32     33     34     35     36    249    250    251    510    511
                        513    514    608   1698   1704   1719   1725   1727   1737   1738   1739   1740   1741
                       1747   1759   1761   1762   1763   1770   1771   1772   1784   1787   1790   1797   1798
                       1799   1800   1809   1813   1825   1829   1837   2535
$SWREG  001254          271#   534
$SWRMK= 000000         1741
$TESTN  001236          262#  1782*
$TIMES  001222          249#   510*   608*  1704*  1770*  1777   1780*  1790
$TKB    001146          224#  2176   2187   2204   2258   2264   3178
$TKS    001144          223#  2176   2185   2201   2225*  2256   2262
$TMPO   001202          241#   662*   828*   848    960*  1077*  1119*  1120*  1121   1130   1141   1146   1147
                       1625*  1642   3219   3220   3557*  4000   4005   4026   4054
$TMP1   001204          242#   663*   962*  4000   4005
$TMP2   001206          243#   963*  4005
$TMP3   001210          244#   964*  4005
$TMP4   001212          245#   965*  4005
$TMP5   001214          246#   966*  4005
$TMP6   001216          247#   967*  4005
$TMP7   001220          248#   888*   903    920*   923*   926*   948*   950   1668*
$TN   = 000002           26#   604    608#
$TPB    001152          226#  2105*  2116
$TPFLG  001157          230#  2054   2116
$TPS    001150          225#  2103   2116
$TRAP   014326          506   2461#
$TRAP2  014350         2472#  2483
$TRP  = 000014         2476#  2485*  2486#  2487#  2488#  2489#  2490   2491#  2492   2493#  2494#  2495#  2496#
                       2497#
$TRPAD  014362         2466   2483#
$TSTM   000004           64#
$TSTNM  001102          203#   976*  1703*  1736   1781*  1782   1787   1791   1808   1837
$TTYIN  013744         2284   2285   2297   2315   2329   2333#
$TYPBN= ******  U       2489
$TYPDS  012132         1982#  2488
$TYPE   012356         2054#  2147   2476   2484
$TYPEC  012570         2084   2091   2098   2103#  2104   2227
$TYPEX  012636         2109   2111   2114#
$TYPOC  011730         1923#  2485
$TYPON  011744         1922   1925#  2487
$TYPOS  011704         1918#  2486
```

```
$UNIT   001244            265*
$UNITM  000010             66*
$USWR   001256            272*
$VECT1  001302            297*
$VECT2  001304            298*
$XTSTR  011130           1750*
$$GET4= 000000           1719*
$DFILL  012127           1919*   1923*   1933    1968*
$40CAT= ****** U         1747    1815
.     = 034130            40     41*     43*     45*     51      52*     54*     56*     70*     74*    200*    254     499
                         513     514     603    1727    1731    1790    1791    1837    1892*   2036*   2116   2168*   2176
                        2333*   2334    2340    2393    2453    2513    2537    3849*   4053*   4074*   4137*   4490*   4493*
                        4494*   4495*   4496*   4497*   4498*   4513*   4519*
.$ASTA= ****** U         2120    2123
.$X   = 000000             51*     56
```

```
COMMEN      1#    194#
ENDCOM      1#    194#
ERROR      88#    668     694     713     732     751     771     803     823     840     900     969    1080    1306    1324
          1342   1360    1406    1455    1475    1502    1521    1540    1559    1579    1600    1620    1637    1679    3018
          3028
ESCAPE      1#    194#
GETPRI      1#    194#
GETSWR      1#     10#    194#    536
MHDRS0      1#
MHDR1       1#
MHDR10      1#
MHDR11      1#
MHDR12      1#
MHDR13      1#
MHDR14      1#
MHDR15      1#
MHDR16      1#
MHDR17      1#
MHDR2       1#
MHDR20      1#
MHDR21      1#
MHDR22      1#
MHDR23      1#
MHDR24      1#
MHDR25      1#
MHDR26      1#
MHDR27      1#
MHDR3       1#
MHDR30      1#
MHDR31      1#
MHDR32      1#
MHDR33      1#
MHDR34      1#
MHDR35      1#
MHDR36      1#
MHDR37      1#
MHDR4       1#
MHDR40      1#
MHDR41      1#
MHDR42      1#
MHDR43      1#
MHDR44      1#
MHDR45      1#
MHDR46      1#
MHDR47      1#
MHDR5       1#
MHDR50      1#
MHDR51      1#
MHDR52      1#
MHDR53      1#
MHDR54      1#
MHDR6       1#
MHDR60      1#
MHDR61      1#
MHDR62      1#
MHDR7       1#
```

```
MULT          1#    194#
NEWTST        1#    194#    604
POP           1#    194#    784     858    2023    2162    2163    2382    2442    2522    2523    2638    3134
PUSH          1#    194#    776     845    1982    2123    2125    2146    2356    2409    2503    2509    2621    3123
REPORT        1#    194#
SCOPE        89#    607     975    1702
SETPRI        1#    194#
SETTRA     2476#   2485    2486    2487    2488    2490    2492    2493    2494    2495    2496
SETUP         1#     10#    194#    494
SKIP          1#    194#
SLASH         1#    194#
SPACE       194#
STARS         1#     38      48      50      57    194#    196     254     257     604     606    1695    1733    1793    1839
           1895   1972    2039    2118    2175    2178    2246    2275    2342    2395    2455    2499    2515    2542    2544
           3562   3564    3857    3859    4144    4146
SWRSU         1#    194#    515#
TRMTRP     2476#
TYPBIN        1#    194#
TYPDEC        1#    194#   1714    1879
TYPNAM        1#    194#
TYPNUM        1#    194#
TYPOCS        1#    194#
TYPOCT        1#    194#   1851    1876    2196
TYPTXT        1#    194#
$$CMRE      194#    233     234     235     236     237     238     239     240
$$CMTM      194#    241     242     243     244     245     246     247     248
$$ESCA        1#    194#
$$NEWT        1#    194#    604
$$SET      2476#   2485    2486    2487    2488    2490    2492    2493    2494    2495    2496
$$SETM      531#
$$SKIP        1#    194#
.EQUAT        1#     10#     84
.HEADE        1#      9#     16
.K711         1#
.SETUP        1#      9#    494
.SWRHI        1#     10#     27
.SWRLO       10#     36#
.$ACTI        1#     13#     36
.$APTB        1#     13#    255#
.$APTH        1#     13#     46
.$APTY        1#     13#   2116
.$ASTA        1#
.$CATC        1#     10#     68
.$CMTA        1#      9#    194
.$DB2D        1#
.$DB2O        1#
.$DIV         1#
.$EOP         1#     10#   1693
.$ERRO        1#      9#   1791
.$ERRT        1#     11#   1837
.$MULT        1#
.$POWE        1#     11#   2497
.$RAND        1#
.$RDDE        1#     12#   2393
.$RDOC        1#     12#   2340
.$READ        1#     12#   2173
```

```
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#      11#   1731
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#      11#   2453
.$TYPB      1#
.$TYPD      1#      11#   1970
.$TYPE      1#      11#   2037
.$TYPO      1#      11#   1893
.$40CA      1#
.117C       1#
```

```
ADC     3493    3495
ADD      587     588     589     726     745     759     765     783     849     856     873     874     875     876     909
         979     980     981    1534    1553    1567    1573    1643    1646    1684    1687    1859    1921    1931    2002
        2073    2133    2145    2157    2212    2221    2377    2432    2435    2578    2605    2633    2636    2691    2706
        2943    2965    2981    3003    3038    3041    3055    3087    3088    3113    3165    3242    3247    3271    3276
        3492    3494
ASL      592     661     782     855     872     920     923     926     957     971     982    1017    1856    1857    1858
        2235    2236    2237    2370    2372    2374    2428    2430    2431    2465    2601    3111
ASLB    2007    3519
ASR     2140    2629    2630    2631    2718    2960    2998
BCC     2008
BEQ      533     540     560     568     571     574     593     654     681     719     738     761     791     886     904
         916     946     959     972     986    1016    1022    1031    1037    1039    1088    1090    1092    1165    1190
        1197    1199    1226    1254    1387    1422    1425    1489    1527    1546    1569    1588    1645    1666    1686
        1718    1762    1766    1775    1807    1830    1833    1861    1856    1872    1983    1948    2063    2076    2111
        2127    2131    2151    2153    2192    2219    2234    2303    2365    2421    2439    2602    2604    2677    2690
        2705    2745    2776    2813    2819    2860    2862    2864    2881    2886    2892    2897    2902    2905    2911
        2916    2920    2925    2928    2945    2949    2983    2987    3078    3083    3103    3105    3148    3150    3181
        3185    3239    3268    3298    3305    3307    3333    3339    3350    3365    3371    3378    3394    3400    3424
        3431    3443    3518
BGE     1778    2936    2974
BGT     1169    1709    1955    2016    2231    2272    2367    2423
BHI     1764
BIC      687     706     796     813    1011    1086    1118    1120    1495    1514    1593    1610    1663    1706    1945
        2188    2205    2232    2259    2265    2273    2376    2632    2635    2685    2757    2761    2766    2786    2795
        3109    3152    3236    3265    3329    3382    3390    3516
BICB    2608    3179    3340    3372    3448
BIS      647     659     669     696     715     734     753     773     786     805     825     842     857     883     958
        1078    1101    1111    1151    1461    1480    1649    1662    1950    1951    2010    2011    2239    2672    2674
        2675    2679    2748    2749    2756    2765    2778    2779    2785    2794    3110    3153    3243    3255    3272
        3284    3343    3347    3354    3375    3403    3407
BISB     705     816     881    1513    1613    1660    1848    2572
BIT      556     561     590     680     790     903     945     958     985    1488    1587    1747    1765    1772    1913
        1829    2426    2603    2676    2681    2818    2880    2891    2310    2919    2927    2941    2953    2979    2991
BITB     532    2062    2067    2099    2130
BLO     2298
BLOS    2286    2742    2809
BLT     1094    1229    1956    1999    2015    2090    2229    2270    2369    2425    2939    2977
BMI      700     809     829    1311    1329    1347    1365    1377    1408    1457    1508    1606    1626    2006    2909
        3081
BNE      499     522     538     542     552     557     562     576     591     851     911     983    1041    1072    1122
        1131    1142    1171    1177    1201    1264    1269    1273    1277    1281    1285    1289    1301    1308    1319
        1326    1337    1344    1355    1362    1395    1401    1428    1434    1450    1467    1648    1689    1748    1773
        1814    1819    1849    1875    1946    2004    2061    2068    2070    2078    2086    2100    2107    2129    2135
        2138    2155    2184    2190    2210    2217    2224    2261    2267    2290    2292    2308    2312    2322    2418
        2427    2521    2550    2552    2558    2560    2595    2662    2684    2714    2724    2754    2783    2849    2942
        2954    2962    2967    2980    2992    3000    3005    3052    3054    3065    3100    3133    3155    3159    3163
        3183    3187    3224    3241    3249    3270    3278    3309    3312    3342    3346    3374    3402    3406    3458
        3470    3482    3499    3509    3528
BPL      921     924     927    1826    1944    1990    2020    2055    2104    2186    2202    2257    2263    3212
BR       524     544     564     594     623     624     634     640     644     656     671     676     954     973    1018
        1035    1074    1103    1106    1128    1139    1144    1178    1181    1194    1232    1259    1294    1392    1435
        1438    1469    1750    1756    1759    1768    1771    1824    1854    1878    1885    1922    1937    1958    2001
        2018    2057    2083    2093    2102    2109    2121    2143    2213    2240    2242    2269    2301    2310    2316
        2318    2378    2391    2437    2452    2513    2537    2599    2606    2692    2707    2758    2787    2865    2898
        2899    2927    2940    2947    2951    2963    2968    2975    2978    2985    2989    3001    3006    3085    3157
```

```
            3161  3245  3253  3274  3282  3302  3310  3314  3337  3344  3348  3353  3369  3376  3281
            3398  3404  3408  3411  3428  3445  3521
BVS         2433  2436
CLR          489   490   491   492   493   497   510   511   531   566   611   612   613   638   641
             652   724   743   935   995   996  1045  1046  1070  1109  1210  1211  1246  1296  1304
            1314  1322  1332  1340  1350  1358  1381  1404  1411  1453  1465  1532  1551  1703  1704
            1746  1770  1785  1847  1935  1993  1996  2199  2200  2283  2306  2362  2363  2415  2416
            2519  2573  2607  2657  2659  2719  2747  2751  2777  2780  2826  2867  2868  2870  2875
            2876  3039  3042  3063  3086  3107  3127  3417  3420  3438  3466  3526  3549
CLRB         880   976  1025  1184  1247  1380  1659  1769  2022  2082  2108  2159  2160  2161  2313
            2323  2387  2448  2598  3292  3326  3359  3387  3416
CMP          498   521   541   575   653   695   714   733   752   760   772   804   824   841   850
             885   910  1087  1089  1091  1093  1121  1130  1141  1165  1168  1170  1228  1268  1272
            1276  1280  1284  1288  1300  1307  1318  1325  1336  1343  1354  1361  1400  1421  1424
            1427  1449  1503  1522  1541  1560  1568  1580  1601  1621  1638  1644  1647  1665  1685
            1688  1757  1777  2014  2183  2189  2209  2216  2228  2230  2260  2266  2269  2271  2285
            2297  2661  2689  2704  2739  2740  2763  2792  2807  2808  2861  2863  2935  2938  2966
            2973  2976  3004  3064  3077  3082  3102  3147  3158  3162  3182  3240  3248  3269  3277
            3527
CMPB         539   915  1030  1038  1040  1189  1198  1200  1253  1386  1763  1818  2060  2075  2077
            2085  2106  2110  2128  2191  2223  2289  2307  2311  2321  2366  2368  2417  2422  2424
            3180  3184  3186  3297  3306  3308  3332  3341  3345  3349  3364  3373  3377  3393  3401
            3405  3423
COM          555
COMB        2596
DEC         1015  1176  1433  1707  1855  2296  2722  3053  3223  3520
DECB        1943  1954  2089  2092  3480
EMT           88
HALT          74  1827  2056  2512  2536  2716  2931
INC          586   908   917   922   925   928   970   978  1013  1299  1317  1335  1353  1399  1448
            1683  1705  1776  1809  1949  1957  2000  2158  2238  2520  2660  2721  2753  2782  3051
            3101  3106  3131  3132  3226  3457  3469  3481  3498  3508
INCB        1781  1806  2112  2600  3468
ICT           89
JMP           78    80    81    82    83   569   572   577   578   625   697   716   735   754   774
             806   826   943   984   987   997  1004  1042  1081  1202  1212  1218  1266  1270  1274
            1278  1282  1286  1290  1312  1330  1348  1366  1378  1409  1458  1479  1505  1524  1543
            1562  1582  1603  1623  1640  1725  2553  2554  2555  2561  2562  2563  2872  2877  2903
            2906  2932  3021  3031  3190  3195
JSR          558   565   598   609   610   622   626   633   639   643   646   648   655   664   675
             684   689   690   702   708   709   721   727   728   740   746   747   763   766   767
             793   798   799   811   818   819   831   835   836   894   895   896   906   936   949
            1019  1044  1060  1067  1073  1143  1183  1220  1221  1297  1298  1315  1316  1333  1334
            1351  1352  1369  1371  1373  1375  1397  1398  1447  1460  1462  1468  1481  1492  1497
            1498  1510  1516  1517  1529  1535  1536  1548  1554  1555  1571  1574  1575  1590  1595
            1596  1608  1615  1616  1628  1632  1633  1674  1675  1720  1815  1821  2065  2084  2091
            2098  2147  2227  2569  2887  2898  3020  3030  3112  3194  3315  3316  3317  3319  3319
            3320  3321  3453  3463  3475  3487  3504
MOV          496   500   502   503   504   505   506   507   508   509   513   514   517   518   519
             520   525   527   528   529   534   547   548   549   550   563   580   581   592   583
             584   595   596   597   603   608   614   615   618   620   631   635   636   651   662
             663   667   670   674   683   685   688   693   699   703   704   707   712   722   723
             725   731   741   742   744   750   756   757   764   770   777   778   779   784   785
             794   795   802   812   814   815   817   822   828   832   834   839   846   847   848
             852   859   860   865   866   867   868   869   870   884   888   889   890   891   892
             899   902   934   948   956   960   962   963   964   965   966   967   968   977   994
```

```
   999  1000  1010  1012  1014  1029  1047  1048  1049  1050  1051  1054  1056  1057  1059
  1069  1076  1077  1079  1095  1098  1099  1100  1102  1105  1108  1110  1119  1126  1127
  1137  1138  1148  1149  1150  1152  1159  1161  1188  1209  1214  1215  1225  1234  1236
  1238  1239  1240  1243  1248  1252  1255  1265  1303  1305  1321  1323  1339  1341  1357
  1359  1368  1370  1372  1374  1385  1388  1396  1403  1405  1414  1416  1430  1446  1452
  1454  1464  1471  1472  1473  1474  1491  1493  1496  1501  1507  1511  1512  1515  1520
  1530  1531  1533  1539  1549  1550  1552  1558  1564  1565  1572  1578  1591  1592  1599
  1609  1611  1612  1614  1619  1625  1629  1631  1636  1642  1654  1655  1664  1668  1669
  1670  1671  1672  1678  1681  1710  1714  1717  1752  1753  1755  1758  1767  1779  1780
  1783  1784  1787  1788  1808  1810  1831  1834  1846  1851  1860  1865  1870  1871  1873
  1876  1880  1887  1888  1918  1926  1927  1928  1934  1941  1959  1960  1961  1962  1963
  1983  1984  1985  1986  1987  1988  1989  1994  1997  2017  2023  2024  2025  2026  2027
  2029  2030  2058  2059  2064  2072  2087  2124  2125  2132  2136  2141  2142  2144  2146
  2156  2162  2163  2196  2220  2225  2254  2255  2282  2284  2295  2326  2327  2328  2329
  2354  2355  2356  2357  2358  2360  2361  2380  2381  2382  2383  2384  2407  2408  2409
  2410  2411  2413  2414  2429  2441  2442  2443  2444  2461  2462  2466  2472  2473  2501
  2502  2503  2504  2505  2506  2507  2508  2509  2510  2511  2517  2518  2522  2523  2524
  2525  2526  2527  2528  2529  2530  2533  2568  2570  2574  2575  2576  2579  2580  2581
  2597  2622  2623  2624  2625  2626  2628  2638  2639  2640  2646  2647  2648  2649  2650
  2656  2658  2663  2664  2665  2666  2668  2686  2694  2699  2700  2702  2709  2712  2725
  2729  2730  2731  2732  2733  2734  2736  2738  2760  2762  2764  2771  2772  2773  2789
  2791  2793  2799  2800  2801  2802  2804  2806  2811  2817  2822  2823  2824  2825  2928
  2832  2837  2842  2859  2866  2871  2873  2879  2885  2890  2896  2907  2915  2917  2918
  2924  2926  2946  2950  2952  2955  2957  2958  2964  2984  2988  2990  2993  2995  2996
  3002  3013  3014  3015  3016  3017  3023  3024  3025  3026  3027  3036  3037  3040  3062
  3079  3084  3108  3124  3125  3126  3128  3129  3134  3135  3136  3151  3156  3160  3164
  3170  3171  3172  3193  3201  3202  3203  3206  3207  3208  3209  3218  3228  3235  3264
  3296  3313  3331  3363  3392  3422  3454  3455  3464  3465  3476  3477  3478  3488  3489
  3490  3505  3506  3514  3515  3525  3533  3534  3541  3542  3550  3551  3553  3554  3556
  3557
MOVB   512   545   616   619   660   686   781   797   854   871   878  1032  1052  1055  1124
  1125  1133  1134  1135  1136  1146  1147  1162  1173  1174  1175  1180  1191  1241  1244
  1256  1389  1417  1431  1432  1437  1440  1441  1494  1594  1657  1782  1786  1812  1920
  1919  1920  1923  1924  1925  1929  1932  1933  1952  1992  1995  2009  2012  2021  2069
  2097  2105  2119  2120  2122  2187  2204  2258  2264  2288  2293  2299  2304  2319  2364
  2419  2420  2464  2627  2634  2637  3130  3178  3217  3219  3220  3221  3222  3225  3299
  3334  3366  3395  3425  3432  3433  3444  3446  3449  3456  3467  3479  3497  3507
NEG    758   780   853  1097  1235  1445  1566  1930  1991  2440  2577  2816  3205  3227
NOP   1721  1722  1723
RESET 1719  3019  3029  3192
ROL   1936  1938  1939  1940  1942  2371  2373  2375  2720
RTI    526   787   861  1113  1153  1584  1650  1789  1836  1964  2031  2074  2226  2274  2330
  2385  2445  2474  2535  2695  2710  2767  2797  3197  3535  3543  3555
RTS    913   918   929  1023  1309  1327  1345  1363  1482  1691  1890  2114  2164  2467  2582
  2609  2641  2651  2851  2869  2971  3009  3043  3056  3066  3089  3114  3137  3166  3173
  3213  3229  3258  3287  3322  3459  3383  3412  3434  3450  3459  3471  3483  3500  3510
  3522  3529  3536  3544  3558
SUB    833   893  1096  1444  1630  1673  1811  1998  2139  2434  2713  2790  2814  3204
SWAB   879   882  1658  1661
TRAP  2476  2485  2486  2487  2488  2490  2492  2493  2494  2495  2496  3552
TST    537   551   559   567   570   573   718   737  1021  1053  1071  1242  1466  1477  1504
  1523  1526  1542  1545  1561  1581  1602  1622  1639  1754  1774  1825  1832  1982  1947
  2003  2013  2071  2079  2101  2134  2152  2154  2218  2233  2291  2302  2325  2379  2386
  2438  2447  2463  2549  2551  2557  2559  2670  2671  2701  2723  2744  2775  2812  2848
  2901  2904  2908  2944  2948  2961  2982  2986  2999  3099  3104  3149  3154  3188  3191
  3196  3211  3238  3244  3252  3256  3267  3273  3281  3285  3311  3442  3447  3517
```

```
             2492  2493  2494  2495  2496
.IRP          494   604   777   784   846   859  1746  1983  2023  2124  2125  2146  2162  2163  2356
              2382  2409  2442  2503  2509  2522  2523  2622  2638  3124  3134
.LIST            1     7    74   194   231   233   234   235   236   237   238   239   240   241   242
               243   244   245   246   247   248   249   255   258   494   515   604   608  1703  1719
              1741  1836  2275  2476  2484  2485  2486  2487  2488  2489  2490  2491  2492  2493  2494
              2495  2496  2497
.MACRO           1    36   194   531  2476
.MCALL           9    10    11    12    13   194   255   515
.NLIST           1     6    74   194   231   233   234   235   236   237   238   239   240   241   242
               243   244   245   246   247   248   249   255   258   494   515   604   608  1703  1719
              1741  1836  2275  2476  2484  2485  2486  2487  2488  2489  2490  2491  2492  2493  2494
              2495  2496  2497
.PAGE           84   194   323   547   603   677   788   862   975   988  1044  1084  1116  1157  1203
              1245  1296  1411  1460  1485  1585  1651  1693  2542  3856  4143  4437
.REPT           74   233   241  3686  3705  3723  3742
.SBTTL          27    36    46    68    77    84   194   255   323   494   536   604   988  1203  1693
              1731  1791  1837  1893  1970  2037  2116  2173  2340  2393  2453  2476  2497  3561  3856
              4143
.TITLE          16
.WORD           44    62    63    64    65    66    67    74    75    76   202   205   206   207   208
               211   212   213   214   215   216   217   220   221   222   231   233   234   235   236
               237   238   239   240   241   242   243   244   245   246   247   248   260   261   262
               263   264   265   266   267   271   272   273   286   290   293   296   297   298   299
               300   301   302   303   304   305   306   307   308   309   310   311   312   313   314
               315   316   317   318  1708  1711  1726  1863  1868  1969  2066  2113  2148  2389  2392
              2450  2483  2532  2534  3686  3687  3688  3689  3690  3691  3692  3693  3694  3695  3696
              3697  3698  3699  3700  3705  3706  3707  3708  3709  3710  3711  3712  3713  3714  3715
              3716  3717  3718  3723  3724  3725  3726  3727  3728  3729  3730  3731  3732  3733  3734
              3735  3736  3737  3742  3743  3744  3745  3746  3747  3748  3749  3750  3751  3752  3753
              3754  3755  3879  3973  4000  4005  4026  4054  4075  4138
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*,DZDHNB=LIB:SYSMAC.SML,DSK:DHMMAC,DZDHNB
RUN-TIME: 68 64 8 SECONDS
RUN-TIME RATIO: 238/141=1.6
CORE USED: 50K (99 PAGES)

L13

[illegible] SYS #514/546 **END**
10:15:10 LPDAT [LPTSJS Starting Job DZDHNB, Seq #7172, request created at 21-Feb-77 10:08:57]
10:15:11 LPMSG [LPTSTF Starting File DSKZ:DZDHNB.SEQ<055>[400,3722]]
10:15:47 LPMSG [LPTFPF Finished Printing File DSKZ:DZDHNB.SEQ<055>[400,3722]]
10:15:47 LPSUM Spooler runtime 17 Seconds, 156 KCS, 619 disk reads, 1 disk writes, 165 pages printed
[illegible]

M13

10:15:10 LPDAT [LPTSJS Starting Job DZDHNB, Seq #7172, request created at 21-Feb-77 10:08:57]
10:15:11 LPMSG [LPTSTF Starting File DSKZ:DZDHNB.SEQ<055>[400,3722]]
10:15:47 LPMSG [LPTFPF Finished Printing File DSKZ:DZDHNB.SEQ<055>[400,3722]]
10:15:47 LPSUM Spooler runtime 17 Seconds, 156 KCS, 619 disk reads, 1 disk writes, 165 pages printed