

DH11

RELIABILITY DIAGNOSTIC
MD-11-DZDHN-A

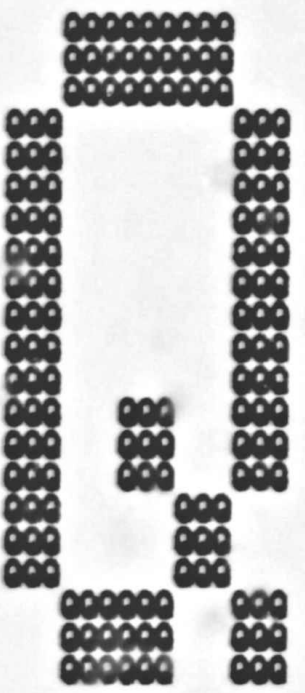
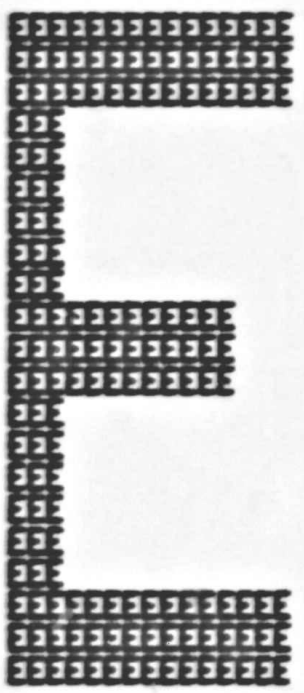
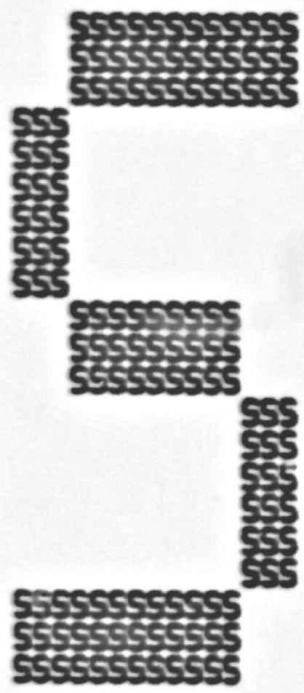
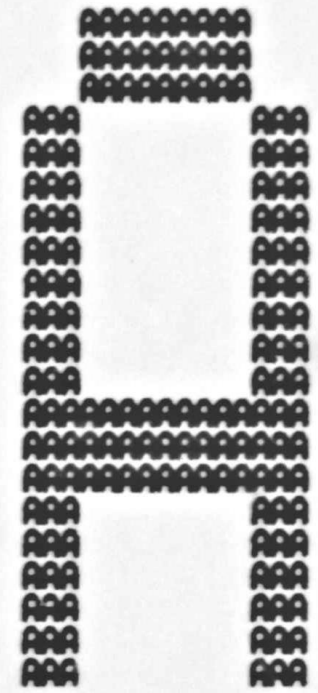
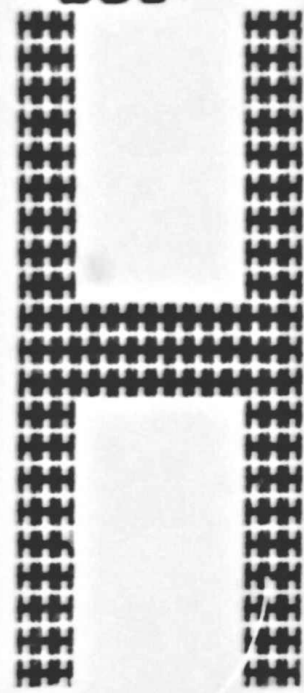
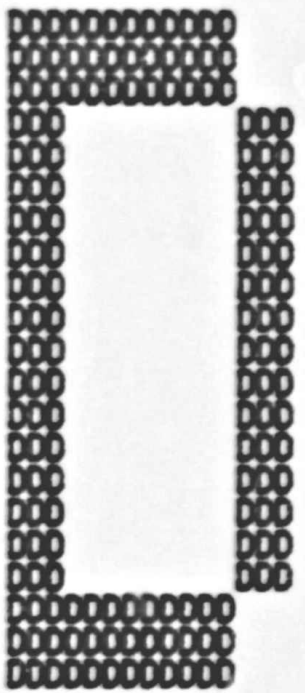
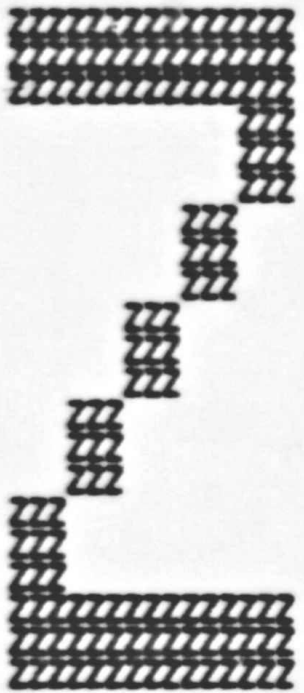
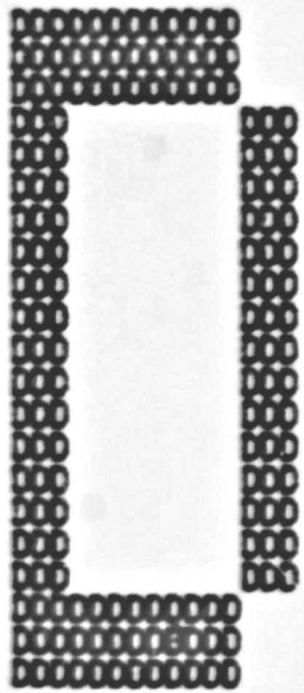
EP-DZDHN-A-DL-A
COPYRIGHT © 1976

FEB 1976
digital
FICHE 1 OF 1
MADE IN USA

DZDHN A SEQ

The microfiche card displays a grid of 14 columns and 20 rows of frames. Each frame contains a small, high-contrast image of a technical diagram or data table. The diagrams include various symbols, lines, and text, typical of a diagnostic manual. The top-left frame is labeled "DZDHN A SEQ". The overall layout is a dense grid of technical information.

B01



LPTSP Version 101(2107) Running on HTA140
START User WELINGHAM,A[1,2] Job DZDHNA Seq. 5761 Date 15-Jan-76 11:38:53 Monitor RX22SD SYS #514/546 *START*
Request created: 15-Jan-76 11:21:39
File: MHTG:DZDHNA.SEO<055>[404,3722] Created: 09-Jan-76 16:08:38 Printed: 15-Jan-76 11:38:55
QUEUE Switches: /FILE:ASCII /COPIES:1 /SPACING:1 /LIMIT:441 /FORMS:NORMAL

C01

SEQ 0001

PRODUCT CODE: MAINDEC-11-DZDHN-A
PRODUCT NAME: DH11 DATA RELIABILITY TESTS / SINGLE LINE
ECHO AND PATTERNS/CABLE TESTS
DATE: 21-JANUARY-1976
AUTHOR: E. CROWLEY
MAINTAINED BY: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1976

DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

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

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

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

TABLE OF CONTENTS

- 1.0 GENERAL PROGRAM DESCRIPTION
- 1.1 PROGRAM PURPOSE
 - 1.1.1 SUBPROGRAM 1 DH11 DATA RELIABILITY TESTS
 - 1.1.2 SUBPROGRAM 2 DH11 SINGLE LINE ECHO TESTS
 - 1.1.3 SUBPROGRAM 3 DH11 SINGLE LINE DATA PATTERNS/CABLE TESTS
 - 1.1.4 CORE MEMORY MAP
- 1.2 SYSTEM REQUIREMENTS
 - 1.2.1 HARDWARE REQUIREMENTS
 - 1.2.2 SOFTWARE REQUIREMENTS
- 1.3 RELATED DOCUMENTS AND STANDARDS
- 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
- 1.5 FAILURE ASSUMPTIONS

- 2.0 OPERATING INSTRUCTIONS
 - 2.1 LOADING AND STARTING PROCEDURES
 - 2.1.1 LOADING PROCEDURES
 - 2.1.2 STARTING PROCEDURES
 - 2.1.2.1 SUBPROGRAM 1 DATA RELIABILITY TESTS
 - 2.1.2.2 SUBPROGRAM 2 SINGLE LINE ECHO TESTS
 - 2.1.2.3 SUBPROGRAM 3 SINGLE LINE DATA PATTERNS/CABLE TESTS
 - 2.1.3 RESTART PROCEDURES
 - 2.2 SPECIAL ENVIRONMENTS
 - 2.2.1 ACT11/APT11
 - 2.2.2 "XDP" SYSTEMS
 - 2.3 PROGRAM OPTIONS
 - 2.3.1 CONSOLE SWITCH REGISTER
 - 2.3.2 CORE MEMORY LOCATIONS
 - 2.4 EXECUTION TIMES
- 3.0 ERROR INFORMATION
 - 3.1 ERROR REPORTING PROCEDURES
 - 3.1.1 STANDARD SYSMAC SML ERROR REPORTING CONVENTIONS
 - 3.1.2 ERROR MESSAGE TABLE
 - 3.1.3 DATA HEADER MNEMONIC DEFINITIONS
 - 3.2 POWER FAIL PRINTOUT
 - 3.3 ERROR HALTS

- 4.0 PERFORMANCE AND PROGRESS REPORTS
 - 4.1 PERFORMANCE REPORTS
 - 4.2 PROGRESS REPORTS
- 5.0 DH11 DEVICE INFORMATION
 - 5.1 ADDRESS AND VECTOR ASSIGNMENTS
 - 5.2 REGISTER DEFINITIONS
 - 5.2.1 SYSTEM CONTROL REGISTER
 - 5.2.2 NEXT RECEIVED CHARACTER REGISTER
 - 5.2.3 LINE PARAMETER REGISTER
 - 5.2.4 CURRENT ADDRESS REGISTER
 - 5.2.5 BYTE COUNT REGISTER
 - 5.2.6 BUFFER ACTIVE REGISTER
 - 5.2.7 BREAK CONTROL REGISTER
 - 5.2.8 SILO STATUS REGISTER
 - 5.3 DH11 MODULE ALLOCATION CHART
- 6.0 MAINTENANCE PROCEDURES
 - 6.1 MAINTENANCE CONNECTORS
 - 6.2 DATA RELIABILITY TESTING
 - 6.3 DATA PATTERNS TESTING
 - 6.4 ECHO TESTING
- 7.0 FLOW CHARTS

1.0 GENERAL PROGRAM DESCRIPTION
-----**1.1 PROGRAM PURPOSE**

"MD-11-DZDHN-A" 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 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.

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. IT HAS TWO MODES OF OPERATION; SEND MODE OR ECHO MODE:

SEND MODE: THE USER TYPES AN ASCII 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.

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.

```
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 *  
*          *  
*****  
SEOP:  *          *  
*          *  
*   STANDARD SYSMAC *  
*   UTILITY ROUTINES *  
*          *  
*****  
CKRST1: *          *  
*          *  
*   COMMON DH11 UTILITIES *  
*          *  
*****  
DHADR: *          *  
*          *  
*   DH11 PROGRAM CONSTANTS *  
*   AND VARIABLES         *  
*          *  
*****  
*          *  
*****  
* CONT. *          * CONT. *  
*****
```

* CONT. *

* CONT. *

EMI:

* 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)

1. M315 TEST CONNECTOR
2. M8611 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, HOWEVER THE PROGRAM CAN NOT BE RUN AS PART OF A QUICK VERIFY "CHAIN" SINCE IT REQUIRES OPERATOR INTERVENTION TO SPECIFY PARAMETERS.
- 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 WHERE THE USER WILL BE ASKED TO TYPE IN THE DH11 PARAMETERS FOR HIS SYSTEM. THE PROGRAM CAN NOT BE RUN AS PART OF AN "XXDP" CHAIN UNLESS UPDATED AS DESCRIBED IN SECTION 2.2.2

1.3 RELATED DOCUMENTS AND STANDARDS

- A. DH11-0 ENGINEERING DRAWINGS
- B. DH11 MANUAL EK-DH11-MM-002
- C. PDP11 PERIPHERALS HANDBOOK
- D. PDP11 PROCESSOR HANDBOOK
- E. MD-11-DZQAC-B1 SYSMAC.SPL
- 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-A 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-DZDHN-A DH11 BASIC DIAGNOSTIC

1.5 FAILURE ASSUMPTIONS

MD-11-DZDHN-A ASSUMES THAT THE DH11 HARDWARE VERIFIED BY MD-11-DZDHN-A, THE BASIC DH11 DIAGNOSTIC, IS FUNCTIONING ERROR FREE.

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

2.1.2 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 2.1.3)

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-A.
2. BOOT THE SYSTEM TO LOAD THE MONITOR
3. ONCE LOADED THE "XXDP" MONITOR PRINTS AN INTRODUCTORY MESSAGE AND RESPONDS WITH A " "
4. TYPE: "DZDHN" 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 AND ASK FOR THE DH11 PARAMETERS.

2.1.3 STARTING PROCEDURES

2.1.3.1 SUB-PROGRAM 1 DH11 DATA RELIABILITY TESTS

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:

A. TO SET UP DH11 SYSTEM PARAMETERS (START AT LOCATION 000200(8))

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

NOTE: 1. SYSTEMS WHERE THE DH11-BB VECTORS ARE INTERLEAVED WITH THE DH11 VECTORS HAVE EIGHT WORDS BETWEEN VECTORS. (THIS IS THE CASE FOR THE 2040 FRONT END)

2. STANDARD SYSTEMS HAVE THE DH11 VECTORS CONTIGUOUS WITH A FOUR WORD 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=1 TEST DH11 #02 ETC.

EXAMPLES:

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

000005<CR> TEST DH11 #00 AND 02

SEQ 0011

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=1 TEST LINE #02 ETC.

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 SR15=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 SR15=1 THE PROGRAM WILL HALT AND TYPE THE FOLLOWING MESSAGE:

"DEPRESS CONTINUE TO START TESTING"

WHEN CONTINUE IS DEPRESSED, THE PROGRAM WILL BEGIN TESTING 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.

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

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

SET THE SR=000200 (QUICK PASS)

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

** THE DEFAULT PARAMETERS ASSUME ONE DH11 WITH THE FOLLOWING ADDRESS ASSIGNMENTS

DH11 #0 DEVADR=760020, VECTOR=330, BR5

8. PROGRAM EXECUTION BEGINS. REFER TO SECTIONS 2.4, 3.0, AND 4.0 FOR EXECUTION TIMES, ERROR REPORTS, AND PROGRESS REPORTS.

C. CHANGE DEVICE AND LINE SELECT PARAMETERS (START AT LOC 000210(8))

1. INSTALL THE REQUIRED TEST CONNECTORS FOR THE SPECIFIC TEST APPLICATION.
2. SET THE HALT/ENABLE SWITCH TO THE HALT POSITION
3. SET THE SR=000210
4. DEPRESS LOAD ADDRESS
5. SET THE SR=000000 (WORST CASE TESTING)
SET THE SR=000200 (QUICK PASS)
SET THE SR=100000 (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 (A-10) ABOVE.
9. PROGRAM WILL ASK FOR LINE SELECTION PARAMETERS.
PROCEED AS IN (A-11) ABOVE.
10. PROGRAM WILL BEGIN EXECUTION AS DESCRIBED IN
PARA 2.1.3.1 (A,12) ABOVE

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=000000
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 WORDS BETWEEN VECTORS. TYPE EITHER A "4" OR "10" AS DESCRIBED IN PARA 2.1.3.1 (A7) 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. 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).

002

SEQ 0015

1. TERMINATE THE LINE TO BE TESTED WITH AN M315 TEST CONNECTOR.
2. SET THE HALT/ENABLE SWITCH TO HALT
3. SET THE SR=000220(8)
4. DEPRESS LOAD ADDRESS
5. SET THE SR=000000
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 WORDS BETWEEN VECTORS. TYPE EITHER A "4" OR "10" AS DESCRIBED IN PARA 2.1.3.1 (A 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 000.
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.

SEQ 0018

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 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 DN11'S, LINE PARAMETERS ETC. YOU MUST RESTART THE TESTS AT LOC. 000220(8).

2.1.4 RESTART PROCEDURES

SAME AS THE STARTING PROCEDURES

SPECIAL ENVIRONMENTS

- 2.2.1 ACT11/
APT11 THE PROGRAM MAY BE LOADED BY THE ACT11/APT11 SYSTEMS, BUT MAY NOT BE RUN AS PART OF A QUICK VERIFY CHAIN SINCE THE PROGRAM REQUIRES OPERATOR INTERVENTION.
- 2.2.2 XXDP ALTHOUGH THE PROGRAM IS NOT DESIGNED TO RUN UNDER CHAIN MODE IN AN "XXDP" ENVIRONMENT, THE USER MAY MODIFY AND UPDATE THE PROGRAM TO CREATE A ".BIC" FILE ON THE "XXDP" MEDIA THAT MAY BE CHAINED USING THE FOLLOWING PROCEDURE:
- 1) LOAD THE UPDATE PROGRAM (UPD1 OR UPD2)
 - 2) USE UPDATE TO LOAD "DZDHNA.BIN"
 - 3) SET SR15=1 AND START THE PROGRAM AT LOC 000200(8)
 - 4) TYPE IN THE DH11 PARAMETERS TO MATCH THE SYSTEM TO BE TESTED.
 - 5) AFTER THE PROGRAM HALTS, PATCH LOCATION 202(8) AS FOLLOWS:

EXAMINE LOCATION 206(8) AND NOTE ITS CONTENTS
DEPOSIT WHAT IS IN LOCATION 206(8) INTO LOCATION 202(8)
- 6) NOW RESTART THE UPDATE PROGRAM AND DUMP THE PROGRAM BACK ON TO THE "XXDP" MEDIUM USING THE NAME "DZDHNA.BIC"
 - 7) THIS ".BIC" FILE MAY NOW BE USED TO RUN THE DATA RELIABILITY PROGRAM AS PART OF A CHAIN.

2.3 PROGRAM OPTIONS

2.3.1 CONSOLE SWITCH REGISTER

 IMPORTANT NOTE

FOR ANY CPU THAT HAS NO PHYSICAL SWITCH REGISTER THERE IS A CORE LOCATION TAGGED "SMR:" THAT SERVES AS A SOFTWARE SWITCH REGISTER. THE BITS IN THIS LOCATION SERVE THE SAME FUNCTION AS THE SWITCH REGISTER BITS DESCRIBED BELOW.

- 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 |
| SR7=1 | QUICK VERIFY - DO COMPLETE TESTING ON EACH LINE AT 9600. BAUD ONLY |
- 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

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 "BRVL:" 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 800). 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 (A 10) FOR A DESCRIPTION OF ITS ENCODING.

5) LINE SELECTION PARAMETER (PARA 2.1.3.1 (A11))

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.

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 TIMESA. SUB-PROGRAM 1 DH11 DATA RELIABILITY TESTS

1. SR07=1 QUICK TEST

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

2. SR7=0 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 ENT) 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 ERROR MESSAGES WITHIN MD-11-DZDHN-A 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.

;ERROR TABLE ITEM FOR ERROR 1

```
EM1      : "NON EX MEMORY ERROR - DROPPED LINE # "
DH1      : " (PC)  CURLPR  DEVAOR  REGADR  WAS      S/B"
DT1      : "SERRPC,CURLPR,SREG1,SREG2,SREG3,SREG4
DF2      : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 2

```
EM2      : "TRANSMITTER FALSE INTERRUPT - DROPPED LINE # "
DH1      : " (PC)  CURLPR  DEVAOR  REGADR  WAS      S/B"
DT1      : "SERRPC,CURLPR,SREG1,SREG2,SREG3,SREG4
DF2      : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 3

```
EM3      : "BUFFER ACTIVE REGISTER ERROR - DROPPED LINE # "
DH1      : " (PC)  CURLPR  DEVAOR  REGADR  WAS      S/B"
DT1      : "SERRPC,CURLPR,SREG1,SREG2,SREG3,SREG4
DF2      : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 4

```
EM4      : "BYTE COUNT REGISTER ERROR - DROPPED LINE # "
DH1      : " (PC)  CURLPR  DEVAOR  REGADR  WAS      S/B"
DT1      : "SERRPC,CURLPR,SREG1,SREG2,SREG3,SREG4
DF2      : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 5

```
EM5      : "CURRENT ADDRESS REGISTER ERROR - DROPPED LINE # "
DH1      : " (PC)  CURLPR  DEVAOR  REGADR  WAS      S/B"
DT1      : "SERRPC,CURLPR,SREG1,SREG2,SREG3,SREG4
DF2      : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 6

```
EM6      : "SILO OVERFLOW ERROR - DROPPED LINE # "
DH1      : " (PC)  CURLPR  DEVAOR  REGADR  WAS      S/B"
DT1      : "SERRPC,CURLPR,SREG1,SREG2,SREG3,SREG4
DF2      : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 7

```
EM7      : "RECEIVER FALSE INTERRUPT - LINE # "
DH1      : " (PC)  CURLPR  DEVAOR  REGADR  WAS      S/B"
DT1      : "SERRPC,CURLPR,SREG1,SREG2,SREG3,SREG4
DF2      : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 10

```
EM10     : "INVALID DATA IN SILO - DROPPED LINE # "
DH2      : " (PC)  CURLPR  CHAR #  WASADR  SHADR  WAS      S/B"
DT2      : "SERRPC,CURLPR,SREG0,SREG1,SREG2,SREG3,SREG4
DF2      : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 11

```

EM11      : "DATA ERROR - LINE # "
DH2       : (PC)  CURLPR CHAR # WASADR SHBADR WAS  S/B"
DT2       : SERRPC,CURLPR,SREG0,SREG1,SREG2,SREG3,SREG4
DF2       : PRINT ALL OCTAL

```

;ERROR TABLE ITEM FOR ERROR 12

```

EM12      : "TEST TIMEOUT - DROPPED LINE # "
DH3       : (PC)  CURLPR RTOTAL XTOTAL RDONE"
DT3       : SERRPC,CURLPR,STMP0,STMP1RDONE"
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       : STMP0,STMP1,STMP2,STMP3,STMP4,STMP5,STMP6
DF1       : PRINT ALL DECIMAL

```

;ERROR TABLE ITEM FOR ERROR 14

```

EM14      : "BUS ERROR TRAP TO 04"
DH4       : (PC)  (PS)  (SP)  TRAPPC TRAPPS"
DT5       : SERRPC,STMP0,SREG6,SREG1,SREG2"
DF2       : PRINT ALL OCTAL

```

;ERROR TABLE ITEM FOR ERROR 15

```

EM15      : "RSVD INSTR TRAP TO 10"
DH4       : (PC)  (PS)  (SP)  TRAPPC TRAPPS"
DT5       : SERRPC,STMP0,SREG6,SREG1,SREG2"
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       : SERRPC,SREG1,LINE,STMP0,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 I/O PATTERN TEST DONE"
DH6       : " (PC)  DEVAOR  LINE  CURLPR  ICOUNT"
DT7       : SERRPC,DHAOR,LINE,CURLPR,SREGO
DF2       : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 20

```
EM20      : "BINARY UP COUNT PATTERN TEST DONE"
DH6       : " (PC)  DEVAOR  LINE  CURLPR  ICOUNT"
DT7       : SERRPC,DHAOR,LINE,CURLPR,SREGO
DF2       : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 21

```
EM21      : "BINARY DOWN COUNT PATTERN TEST DONE"
DH6       : " (PC)  DEVAOR  LINE  CURLPR  ICOUNT"
DT7       : SERRPC,DHAOR,LINE,CURLPR,SREGO
DF2       : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 22

```
EM22      : "RANDOM DATA PATTERN TEST DONE"
DH6       : " (PC)  DEVAOR  LINE  CURLPR  ICOUNT"
DT7       : SERRPC,DHAOR,LINE,CURLPR,SREGO
DF2       : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 23

```
EM23      : "SINGLE CHAR PATTERN TEST DONE"
DH6       : " (PC)  DEVAOR  LINE  CURLPR  ICOUNT"
DT7       : SERRPC,DHAOR,LINE,CURLPR,SREGO
DF2       : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 24

```
EM24      : "TYPED BUFFER PATTERN TEST DONE"
DH6       : " (PC)  DEVAOR  LINE  CURLPR  ICOUNT"
DT7       : SERRPC,DHAOR,LINE,CURLPR,SREGO
DF2       : PRINT ALL OCTAL
```

;ERROR TABLE ITEM FOR ERROR 25

```
EM25      : "DATA PATTERNS TEST TIMEOUT"
DH7       : " (PC)  DEVAOR  LINE  CURLPR  ICOUNT  PATCDE"
DT10      : SERRPC,DHAOR,LINE,CURLPR,SREGO,SREGI
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
 DEVRDR 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 RSVI INSTR TRAP.
 TRPPS CONTENTS OF THE PSW AT THE TIME OF A BUS ERROR
 OR RSVI 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 POP11 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
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000
LLLLL	RRRRR	XXXXX	DDDDD	PPPPP	FFFFF	00000

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 0NN 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 SRI4=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)

5.0 DH11 DEVICE INFORMATION5.1 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/OL11-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 CHART PRESENTS 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. ALL BITS IN THE ACCOMPANYING DIAGRAMS ARE SHOWN IN THE STATE THEY ASSUME ON POWER CLEAR OR INIT.

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.

J03

SEQ 0034

13 TRANSMITTER AND NON-EX-MEM INTERRUPT ENABLE

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.)

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.

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 SETTING 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:

	BIT				
TRANSMITTER	13	12	11	10	
RECEIVER	9	8	7	6	
	--	--	--	--	
	0	0	0	0	ZERO BAUD
	0	0	0	0	50 BAUDS
	0	0	0	0	75 BAUDS
	0	0	0	0	110 BAUDS
	0	0	0	0	134.5 BAUDS
	0	0	0	0	150 BAUDS
	0	0	0	0	200 BAUDS
	0	0	0	0	300 BAUDS
	0	0	0	0	600 BAUDS
	0	0	0	0	1200 BAUDS
	0	0	0	0	1800 BAUDS
	0	0	0	0	2400 BAUDS
	0	0	0	0	4800 BAUDS
	0	0	0	0	9600 BAUDS
	1	1	1	1	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 DMI FROM BITS 00-15 OF THIS REGISTER. ADDRESS BITS 16-17 ARE TRANSFERRED INTO SEMICONDUCTOR MEMORIES IN THE DMI 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.

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 B BYTE CNT	CURRENT ADDR B ADDR	SYSTEM CNTL B RCV SCAN	INTR CNTL	PRIORITY SELECTOR (NOTE #9)	LINE PARAMETER CNTL	UNIBUS CONNECTOR (NOTES #1) & #2)
			M796				M405	M971		
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

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. ED2 MUST BE G727 GRANT CONTINUITY IF MODEM CONTROL MODULE SET IS NOT INSTALLED. * DENOTES DH11-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.

6.0 MAINTENANCE PROCEDURES

THIS SECTION OUTLINES SOME GENERAL TECHNIQUES FOR USING MD-11DZDN-A 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-DZDN-A, PRIOR TO USING THIS PROGRAM TO INSURE COMPLETE CHECKOUT AND VERIFICATION OF THE DH11 HARDWARE.

6.1 MAINTENANCE CONNECTORS

BOTH THE DATA RELIABILITY AND PATTERNS/CABLE SUB-PROGRAMS 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 ALL LINES WITHOUT DATA CABLES

- 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

- 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

- A. INSTALL AN M315 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 A87 OF THE DH11 BACKPLANE).
- B. INSTALL TWO M8611 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 M315 TEST CONNECTOR IN ITS PLACE ON THE DH11-AD 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-DZDM-A AND START IT AT LOC 000200(8).
- 3) TYPE IN THE DESIRED DH11 PARAMETERS - SET THE SR=000000 AND LET THE PROGRAM RUN.

A COMPLETE TEST RUN FOR 16. LINES ON EACH DH11 WILL TAKE APPROX 4 HOURS (THO 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=000200(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-DZDM-A, AND THE DATA RELIABILITY TESTS USE ONLY A BINARY UP COUNT PATTERN FOR DATA TESTING WITH A MAXIMUM BUFFER SIZE OF 256. BYTES. TO PROVIDE 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-DZDM-A AND START IT AT LOC 000220(8) TO RUN THE DATA PATTERNS TESTS.
- 3) REFER TO PARA 2.1.3.3 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 VERIFY THE PARTICULAR LINE AS IT MIGHT BE USED ON-LINE. REFER TO PARA 2.1.3.2 FOR THE OPERATING INSTRUCTIONS FOR THE DH11 ECHO TEST.

MD-11-DZDHN-A DH11 DATA RELIABILITY TESTS

DECFLO VER 00.10 15-DEC-75 09:42 PAGE A

FLOW CHART

MD-11-DZDHN-A DH11 DATA RELIABILITY TESTS

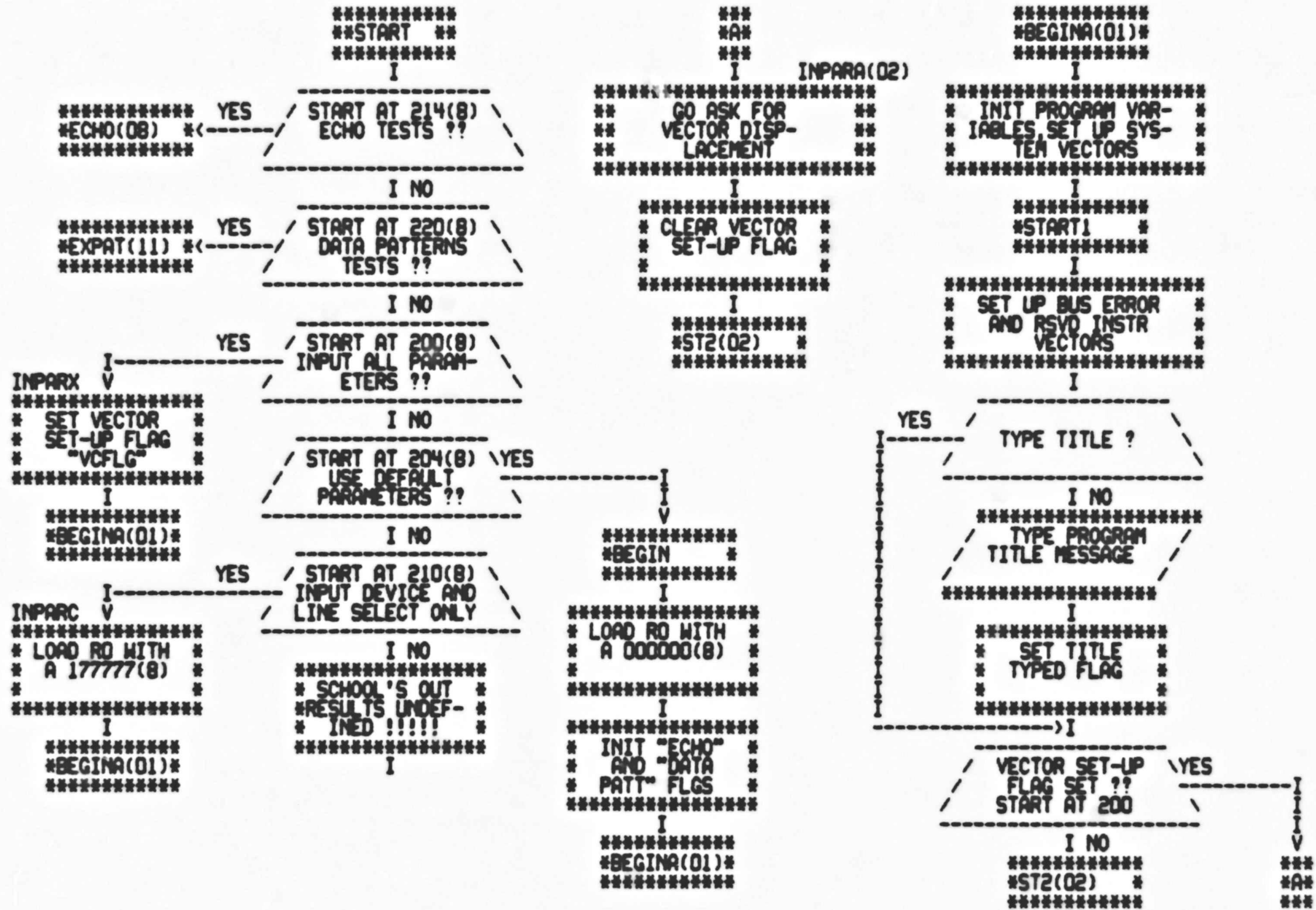
COPYRIGHT 1975
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

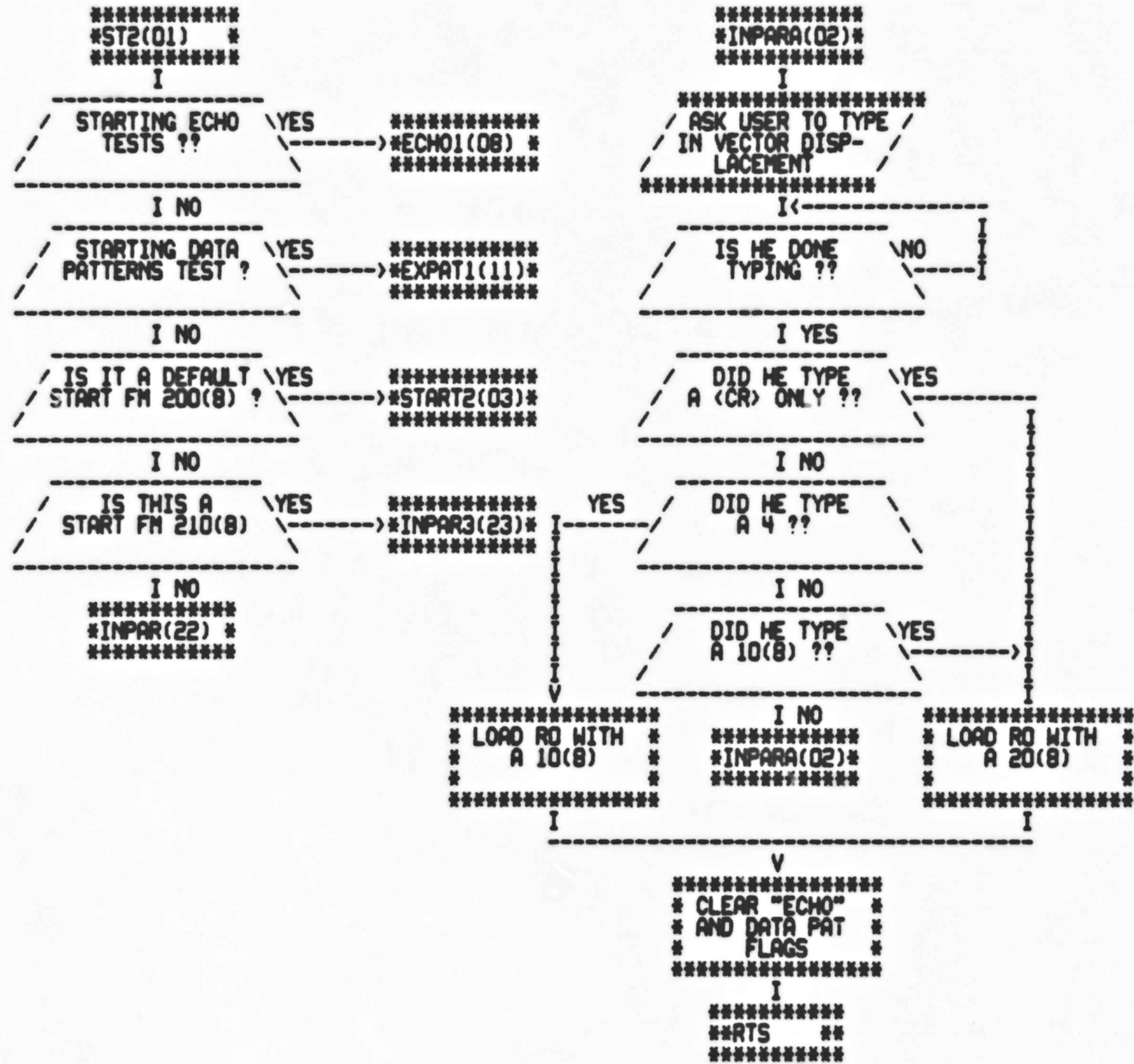
TABLE OF CONTENTS

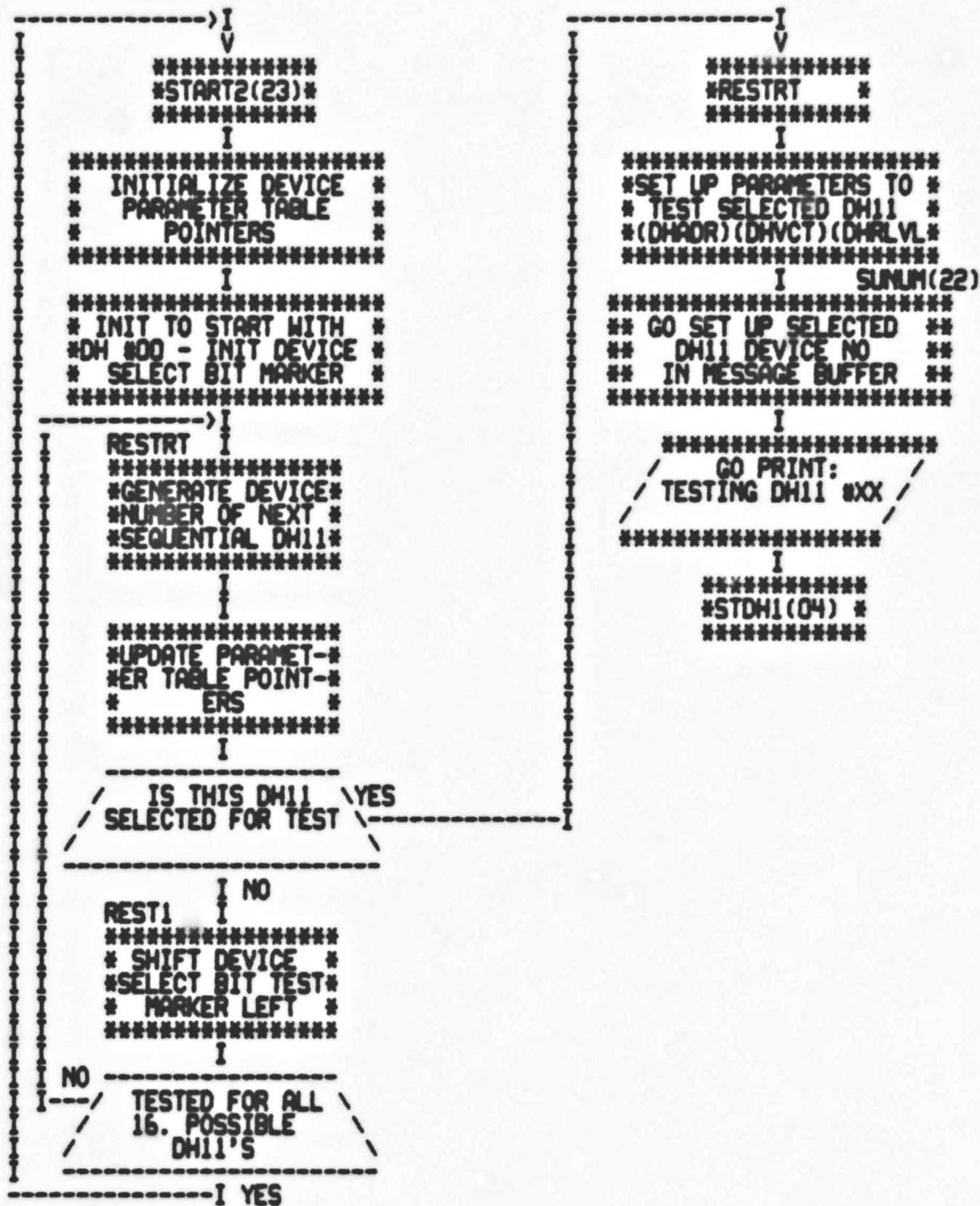
PAGE 01	INITIAL PROGRAM START-UP (PAGE 1)
PAGE 02	INITIAL PROGRAM START-UP (PAGE 2)
PAGE 03	INITIALIZATION
PAGE 04	SELECTED DEVICE INITIALIZATION
PAGE 05	SELECTED DEVICE ACTIVATION
PAGE 06	DATA RELIABILITY -TRANSMITTER INTERRUPT SERVICE
PAGE 07	DATA RELIABILITY - RECEIVER INTERRUPT SERVICE
PAGE 08	ECHO CABLE TESTS - PAGE 1
PAGE 09	ECHO CABLE TESTS - PAGE 2
PAGE 10	ECHO CABLE TESTS - XMITTR INTR SERVICE
PAGE 11	DATA PATTERNS TESTS - PAGE 1
PAGE 12	DATA PATTERNS TESTS - PAGE 2
PAGE 13	DATA PATTERNS TESTS - PAGE 3
PAGE 14	DATA PATTERNS TESTS - PAGE 4
PAGE 15	DATA PATTERNS TESTS - PAGE 5
PAGE 16	DATA PATTERNS TEST - PAGE 6
PAGE 17	DATA PATTERNS TESTS - XMITTR INTERRUPT SERVICE
PAGE 18	DATA PATTERNS TESTS - RCVR INTERRUPT SERVICE
PAGE 19	COMMON SUBROUTINES 1
PAGE 20	COMMON SUBROUTINES 2
PAGE 21	COMMON SUBROUTINES 3
PAGE 22	COMMON SUBROUTINES 4
PAGE 23	COMMON SUBROUTINES 5
PAGE 24	COMMON SUBROUTINES 6
PAGE 25	COMMON SUBROUTINES 7

TABLE OF CONTENTS

PAGE 26	COMMON SUBROUTINES 8
PAGE 27	COMMON SUBROUTINES 9
PAGE 28	COMMON SUBROUTINES 10
PAGE 29	COMMON SUBROUTINES 11
PAGE 30	COMMON SUBROUTINES 12
PAGE 31	COMMON SUBROUTINES 13
PAGE 32	COMMON SUBROUTINES 14
PAGE 33	COMMON SUBROUTINES 15
PAGE 34	COMMON SUBROUTINES 16







```

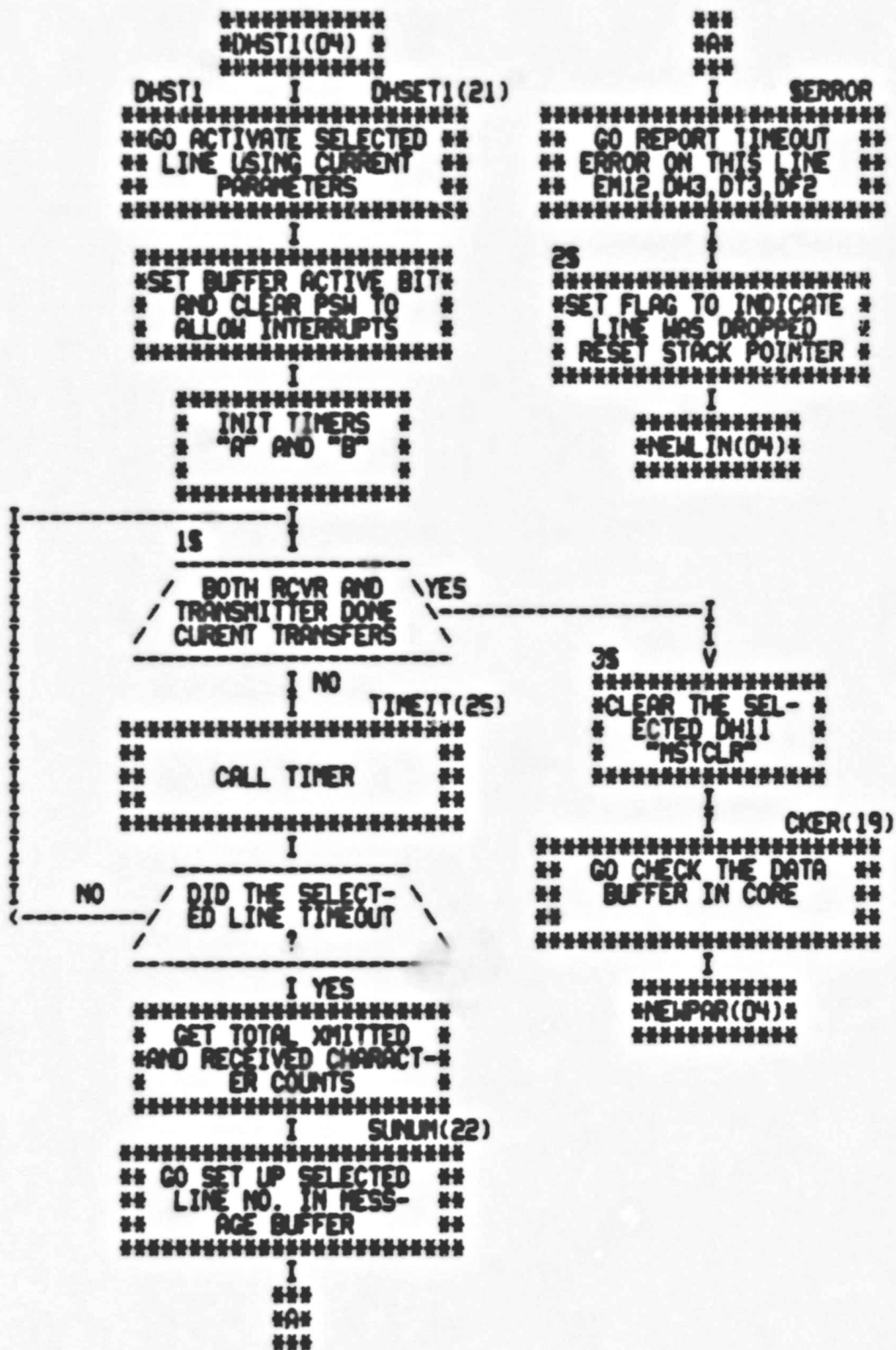
*****
*STDH1(03)*
*****
      I      CLSTAT(26)
*****
** GO CLEAR ALL THE **
** STATISTICS TABLES **
** **
*****
      I      KYBD1(27)
*****
** GO SET UP TO ALLOW **
** KEYBOARD INTERRUPTS **
** **
*****
      I
*****
*INIT SOME FLAGS*
* AND VARIABLES *
*****
      I
*****
* SET UP TRANSMITTER *
* AND RECEIVER VECTORS *
* LOAD RI WITH DMVADR *
*****
***
#A*----->I
*** NEWLIN      I      SELINE(21)
*****
** GO SELECT A LINE **
** NUMBER TO TEST **
** **
*****
      I
*****
      I YES /-----\
      / TESTED ALL \
      \ SELECTED LINES /
      \-----/
      I
*****
#PRSTAT(20)*
*****
      I NO
      I      SUNUM(22)
*****
** GO SET UP LINE **
** NUMBER IN MESSAGE **
** BUFFER **
*****
      I
*****
      / GO PRINT: \
      \ TESTING LINE #XX /
      \-----/
      I
*****
    
```

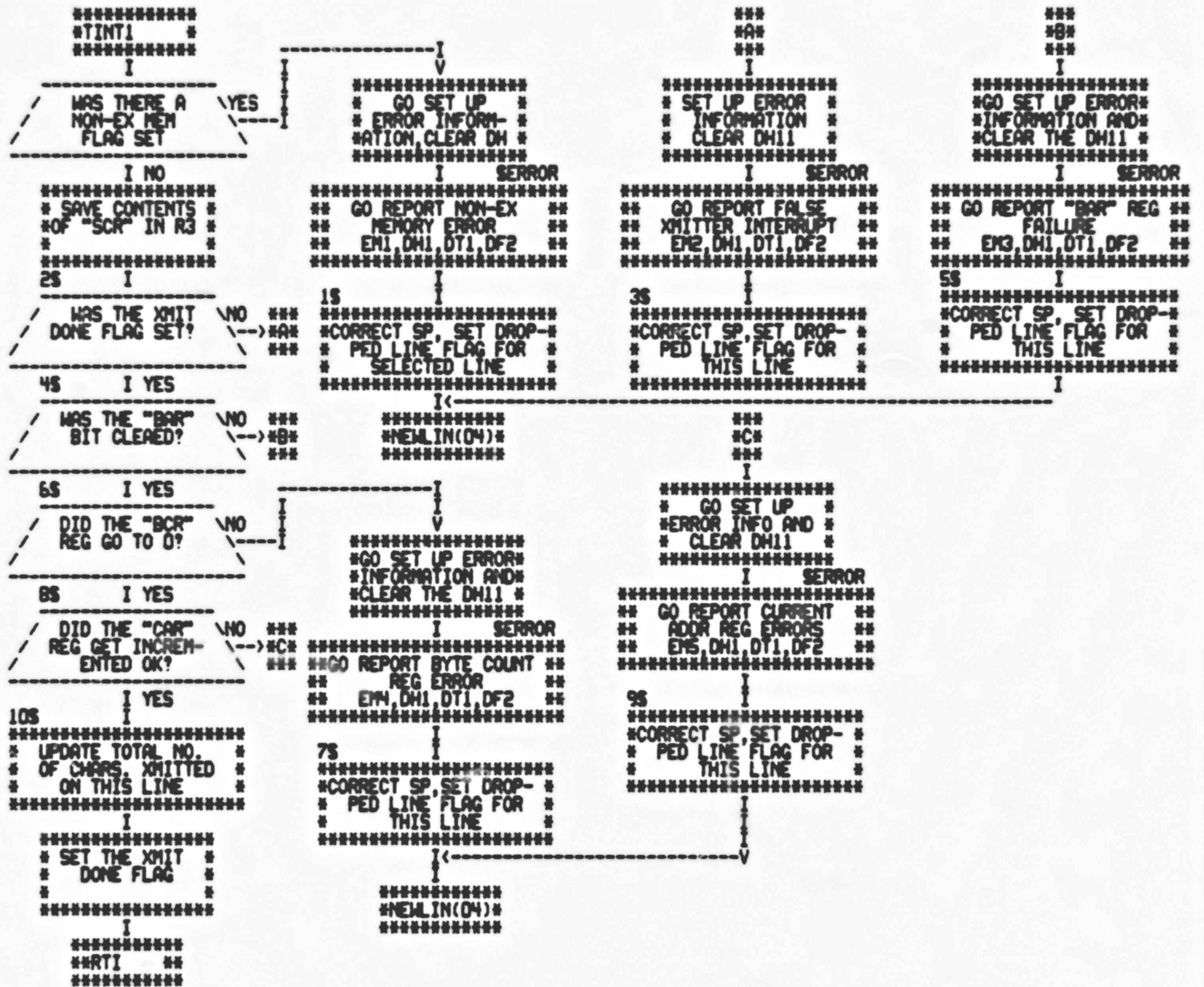
```

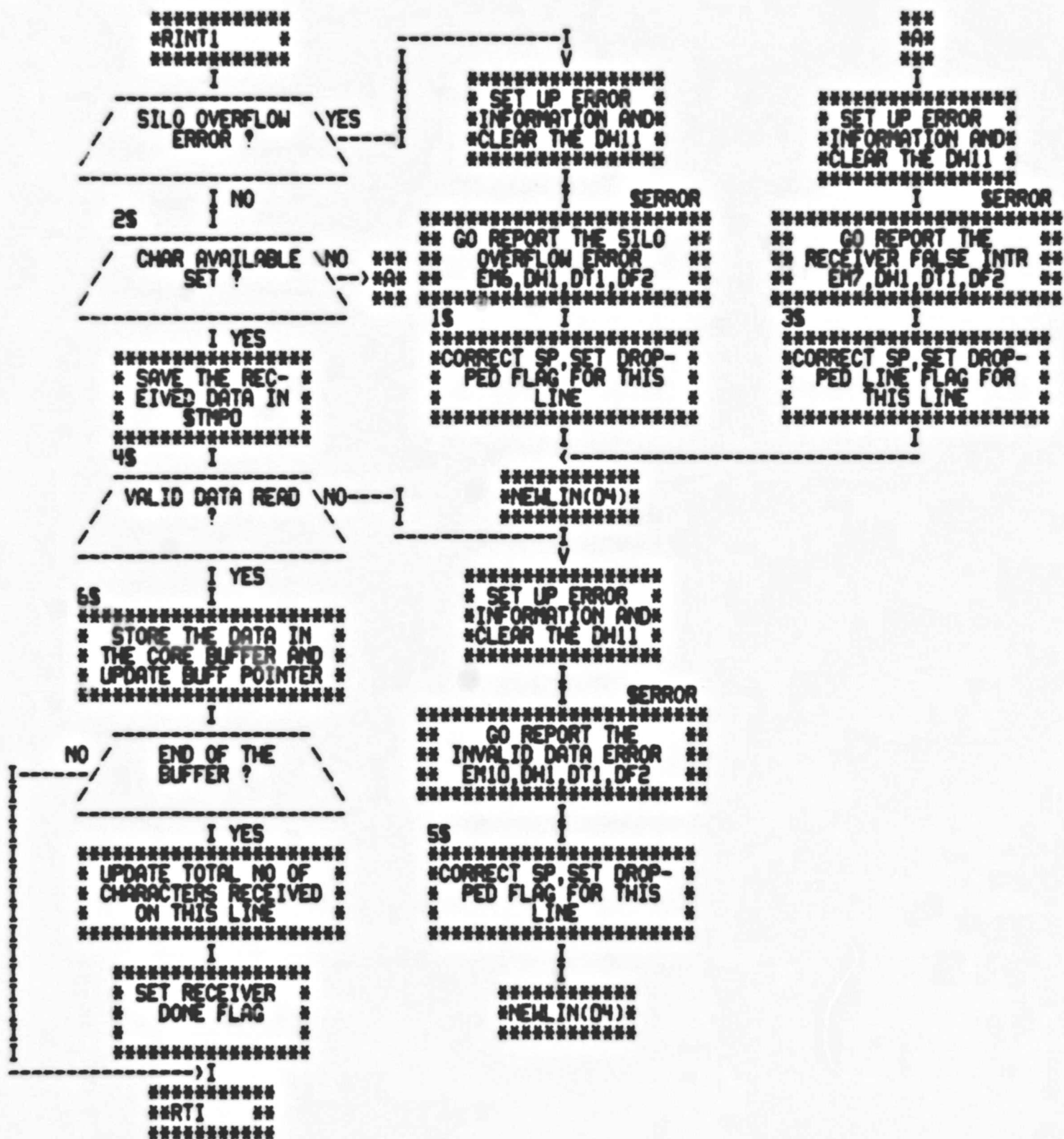
      I
      I V
*****
* INIT "LPR" *
* TABLE POINTER *
*****
      I
*****
      I <-----\
      / NEWLPR      I      SETLPR(26)
      \-----/
*****
** GO RETRIEVE NEW **
** "LPR" CONSTANT FROM **
** TABLE **
*****
      I
*****
*** YES / HAVE WE DONE \
*** #A* <-----/ ALL BAUD RATES ON \
*** /-----\ SELECTED LINE
      \-----/
      I NO
*****
* INIT CHAR COUNT *
* AND CHAR LENGTH *
* SELECT PARAMETERS *
*****
      I
*****
* INIT QUICK *
* TEST FLAG TO *
* ZERO *
*****
***
#B*----->I
*** NEWCL      I      SETCL(26)
*****
** GO SET UP CHARACTER **
** LENGTH SELECT BITS **
** IN "LPR" CONSTANT **
*****
      I
*****
      / HAVE WE TESTED \ YES
      \ ALL FOUR CHARACTER \
      \ LENGTHS /
      \-----/
      I NO
*****
* INIT PARITY *
* SELECT BITS *
* CONSTANT *
*****
      I
*****
    
```

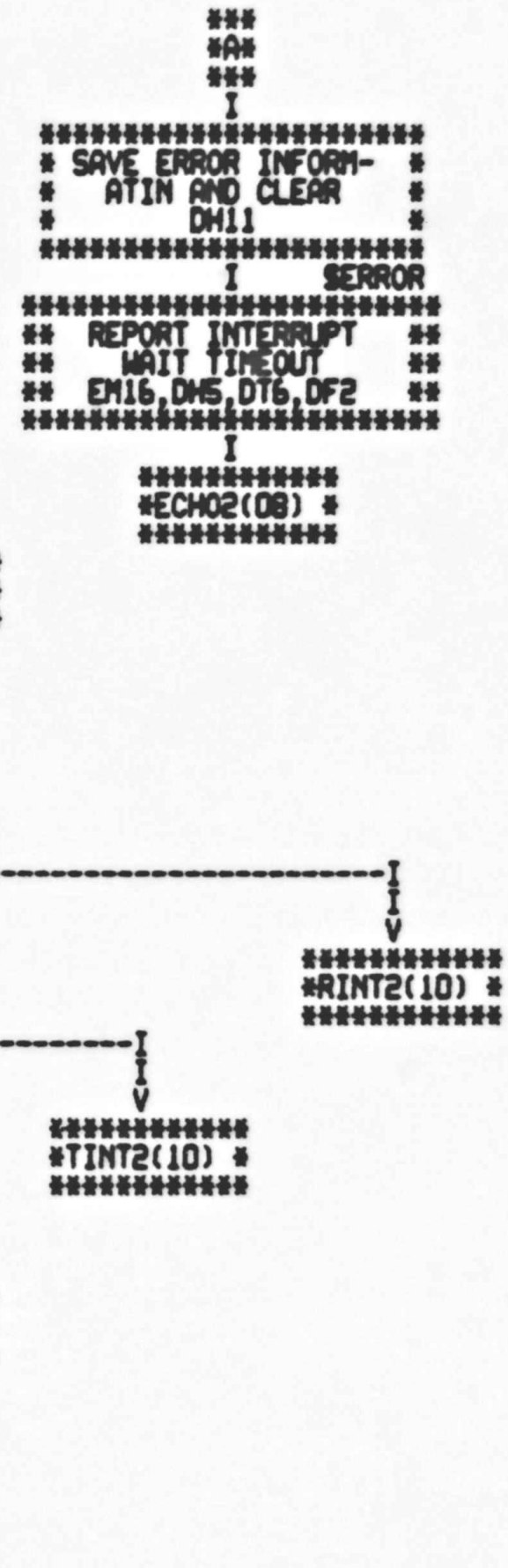
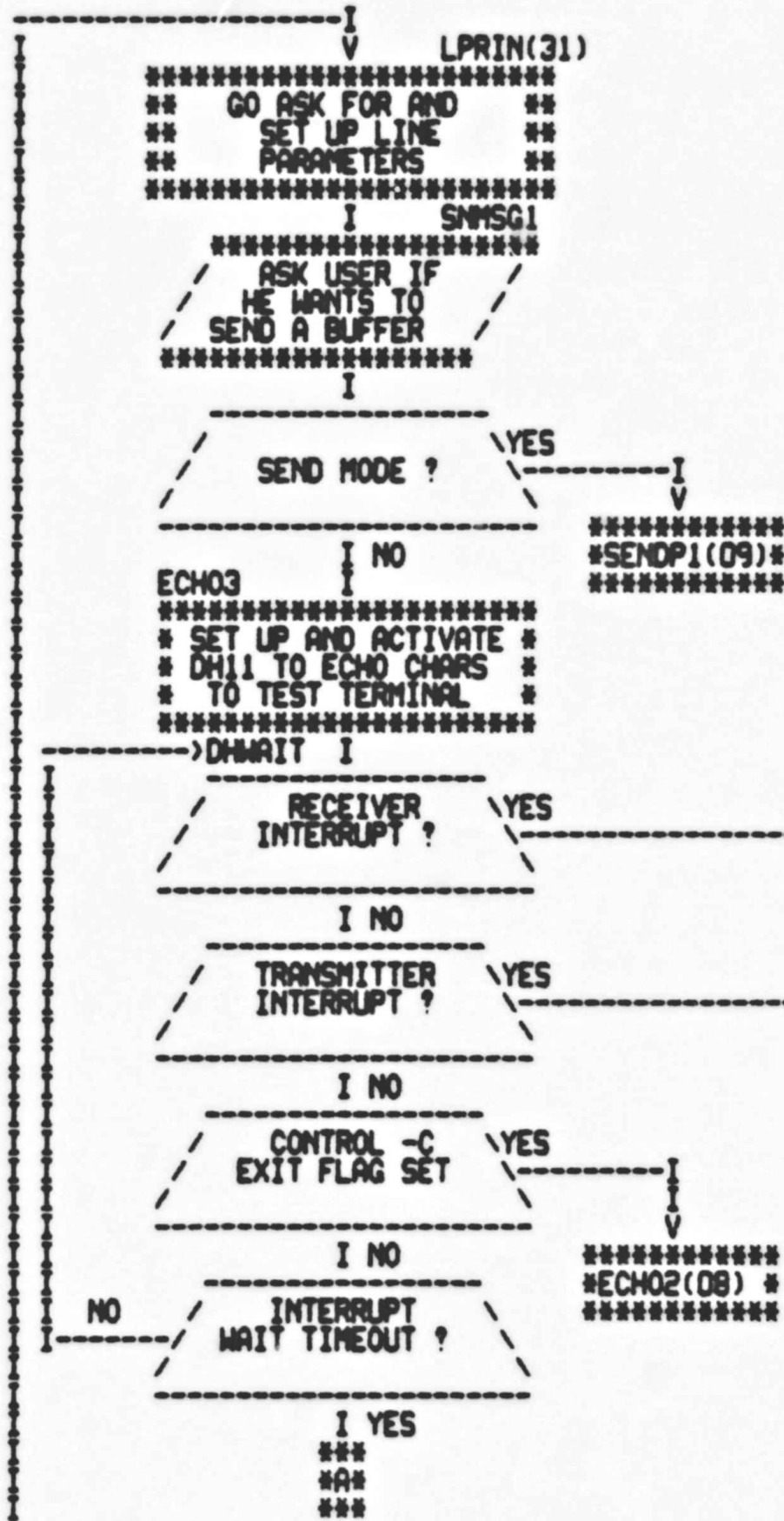
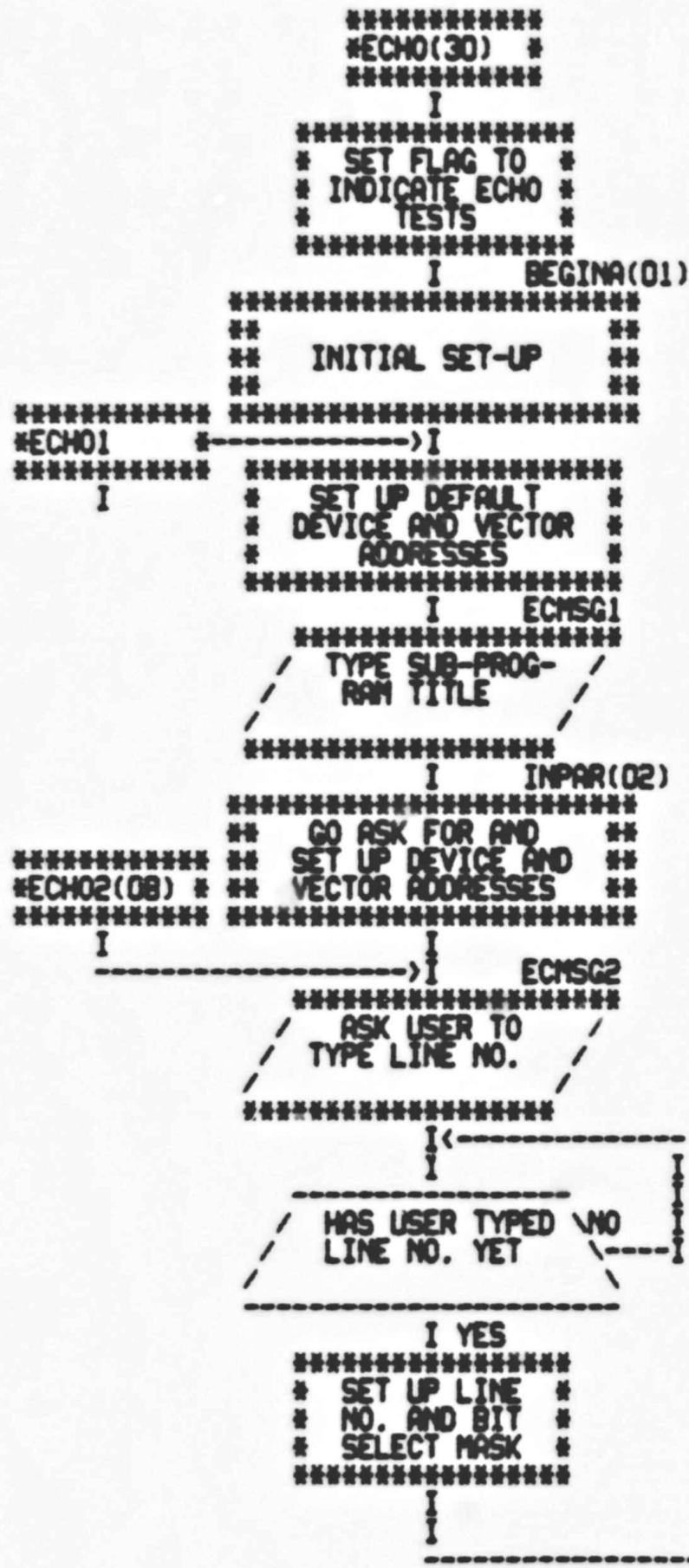
```

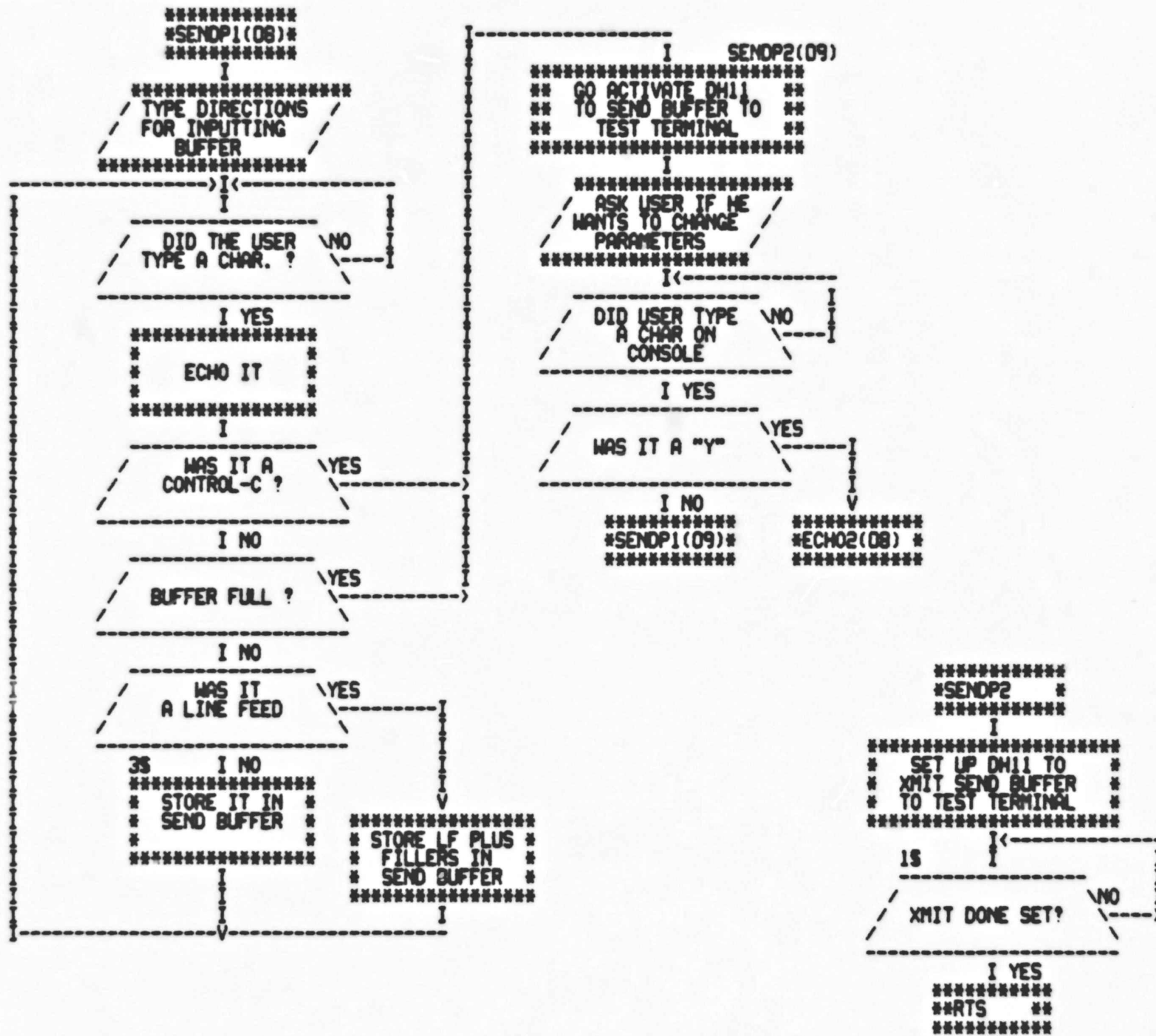
      I
      I V
*****
      I      NEWPAR      I      SETPAR(27)
*****
*** ** GO SET UP NEW **
*** #C* > PARITY SELECT BITS **
*** ** IN "LPR" CONSTANT **
*****
      I
*****
*** YES/ DONE ALL PARITY \
*** #B* <----- SELECT BIT COMB- \
*** /----- INATIONS (3)
      \-----/
      I NO
*****
*DHST1(05)*
*****
*****
#NEWLIN(05)*
*****
      I
      I
***
#A*
***
*****
#NEWPAR(05)*
*****
      I
      I
***
#C*
***
    
```







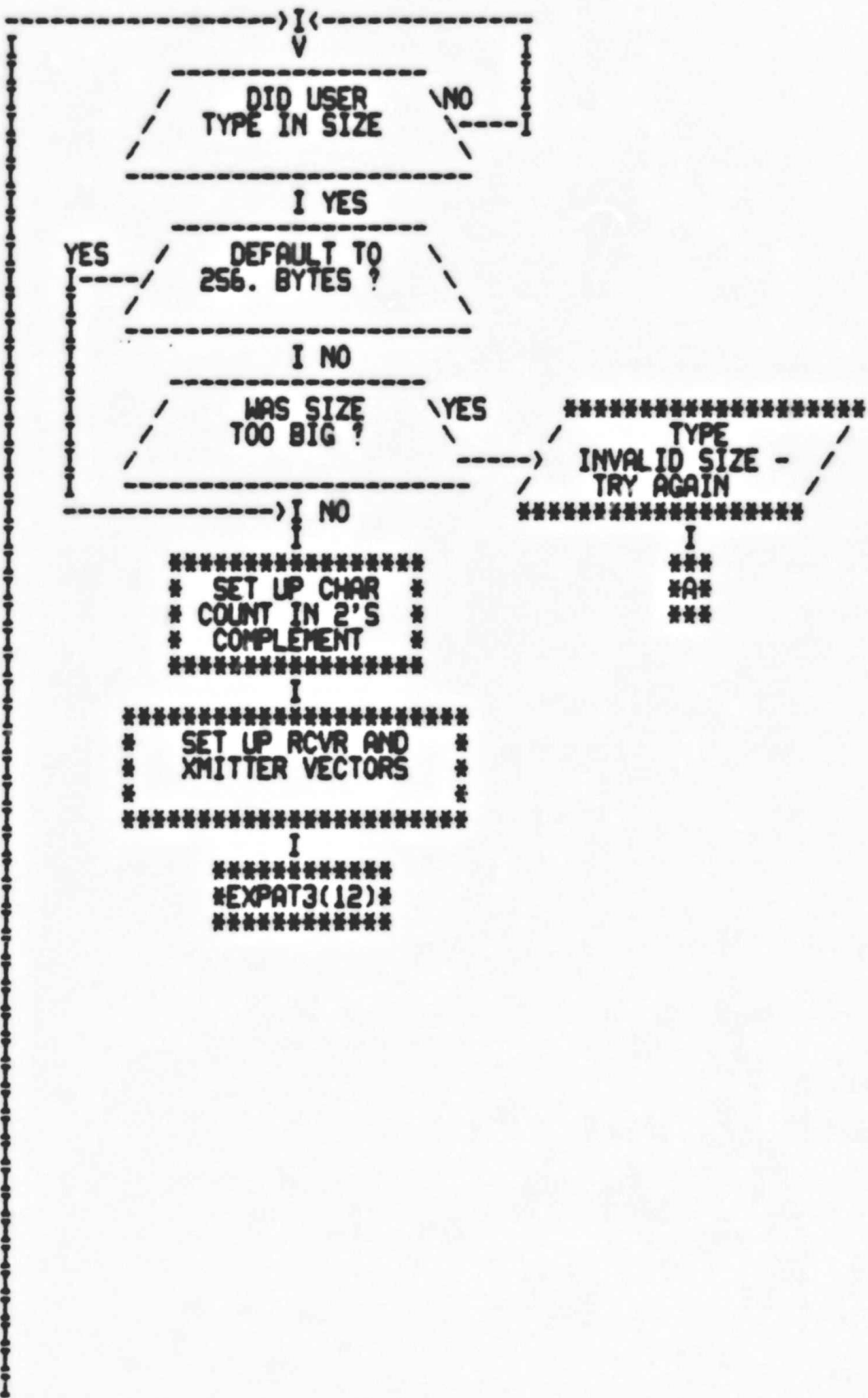


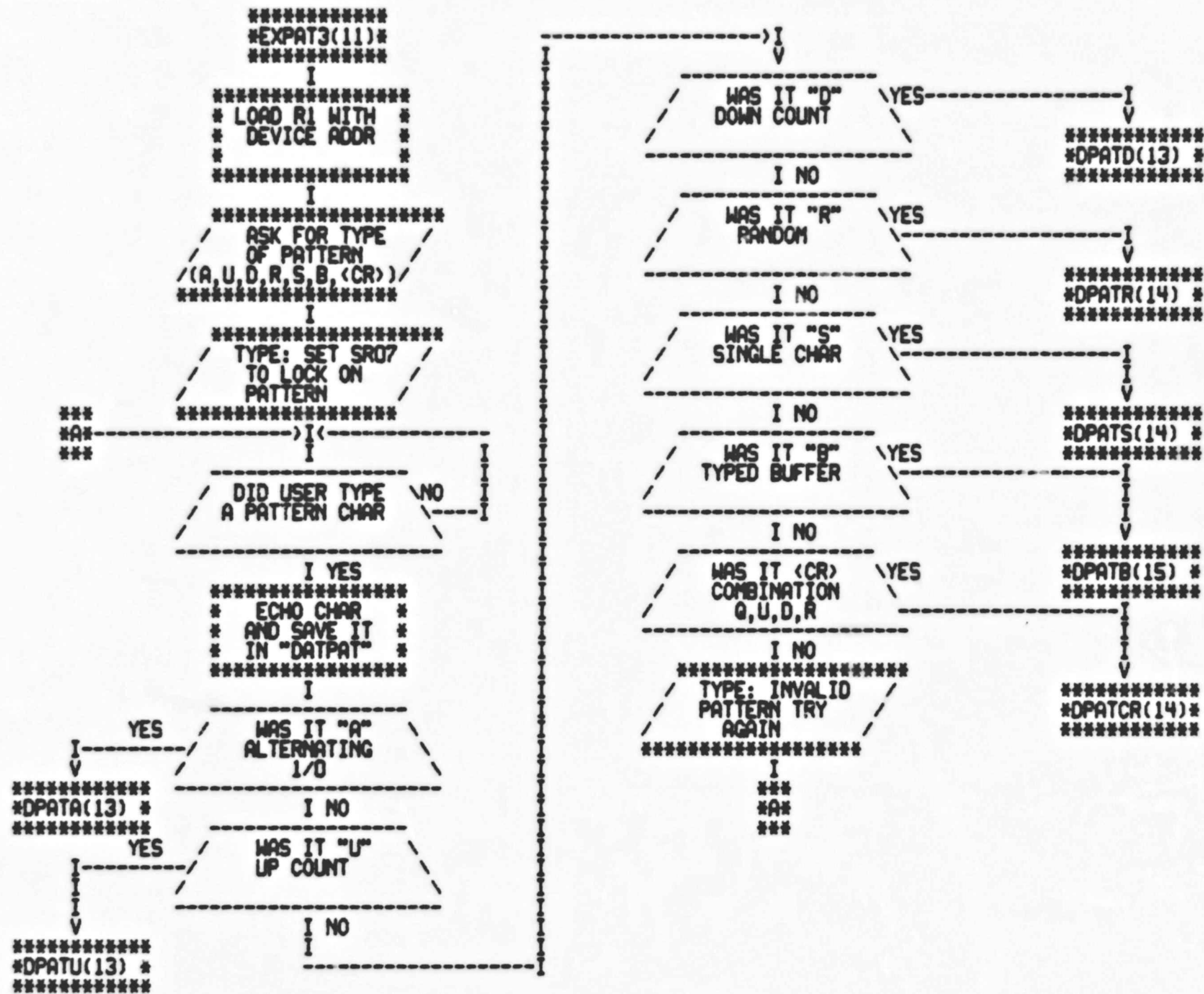


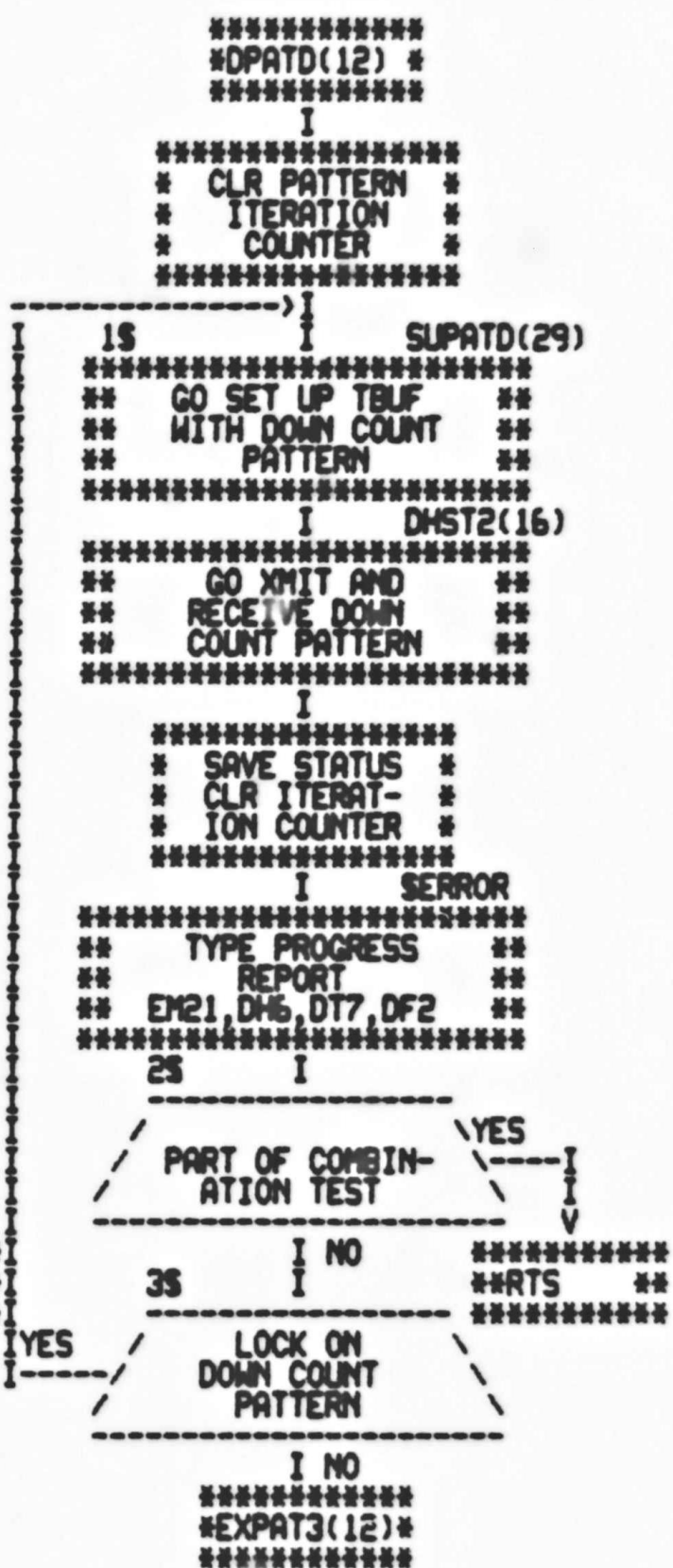
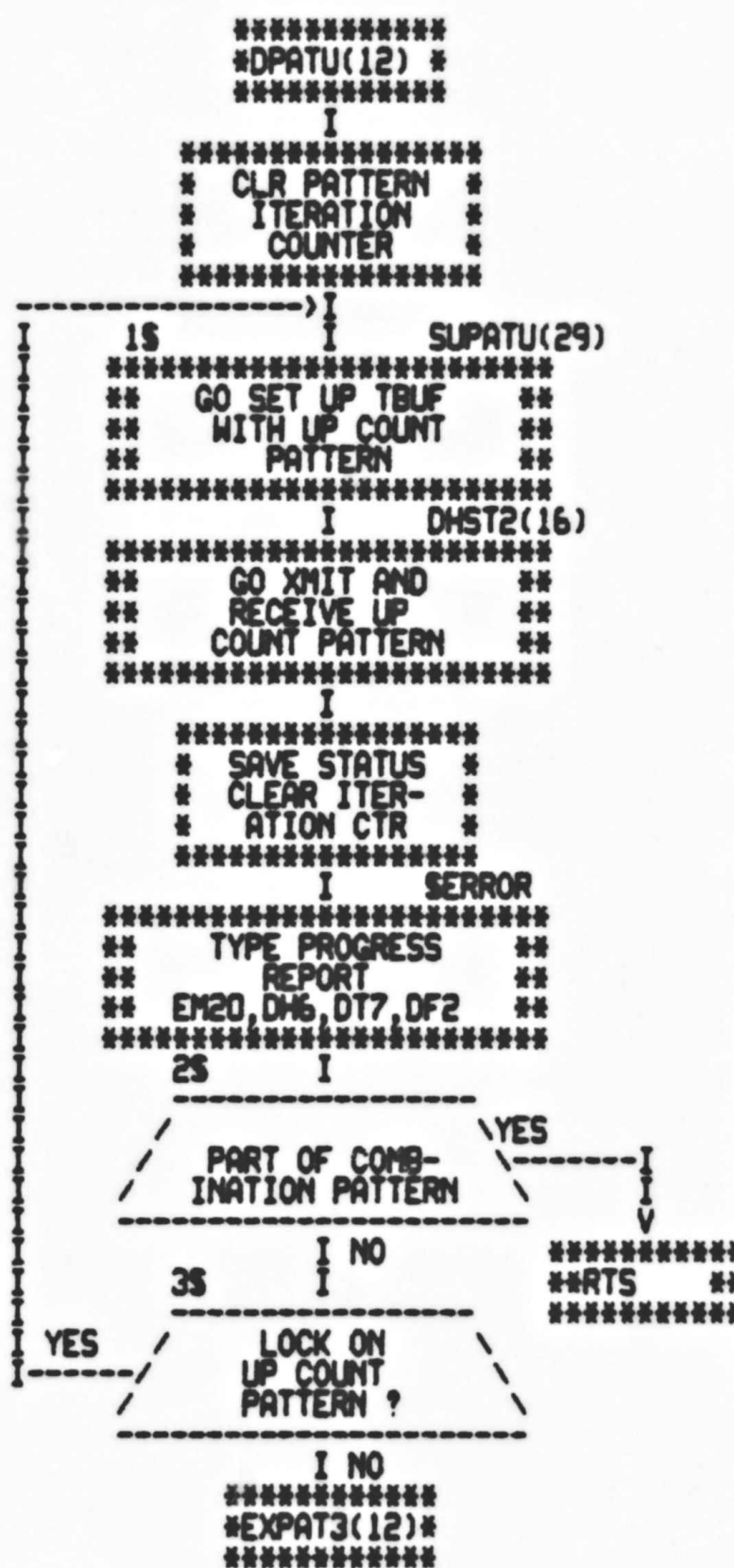
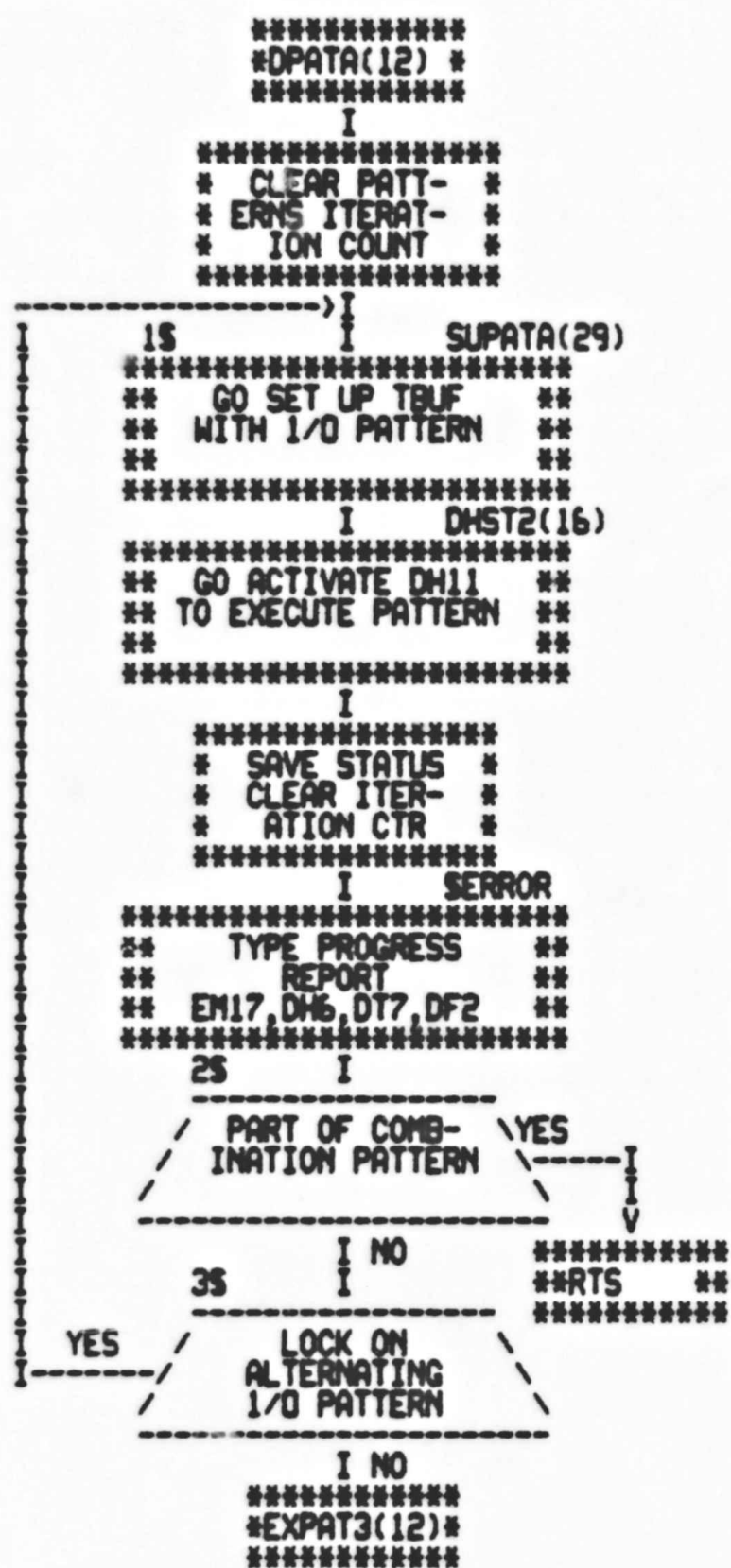

```

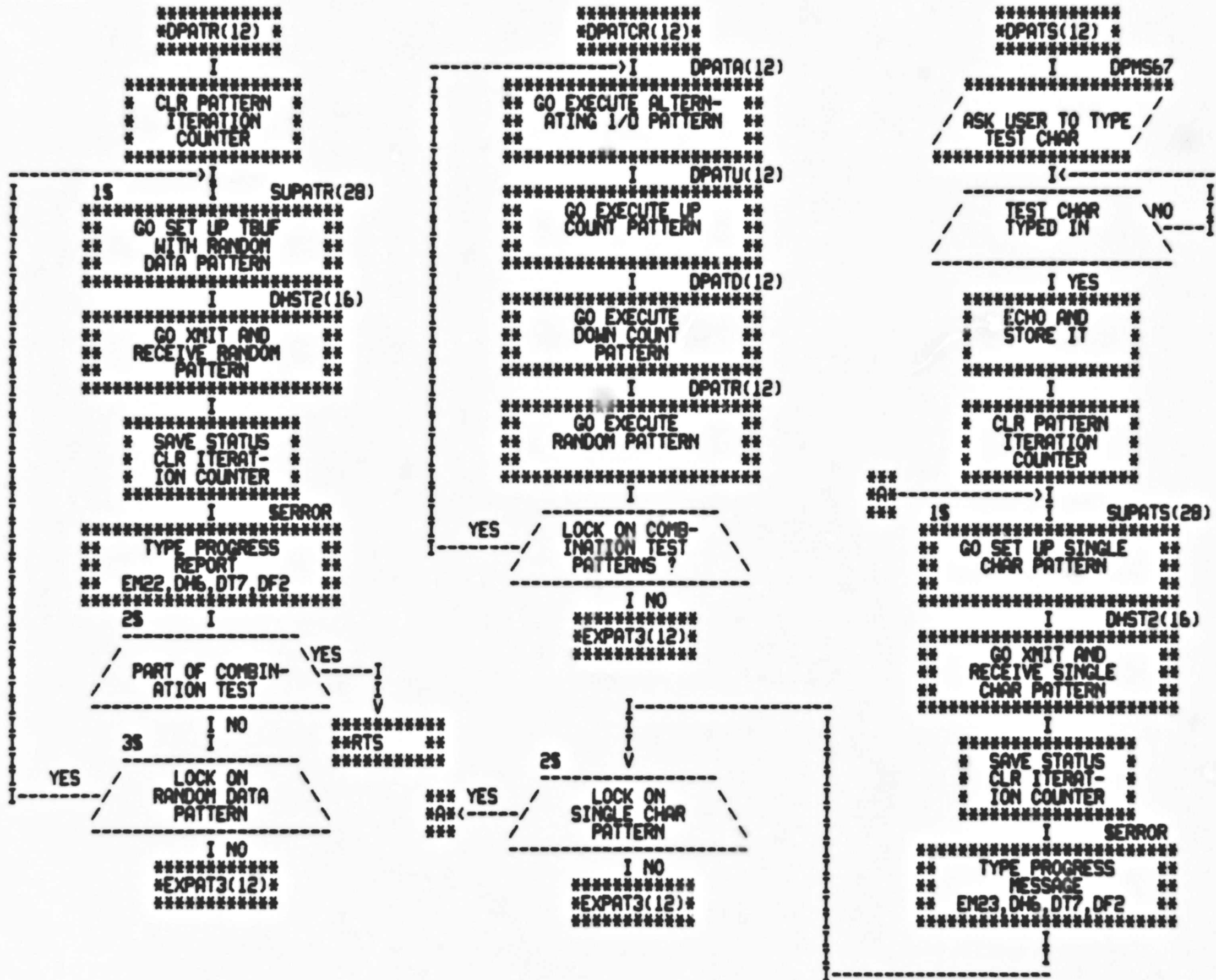
*****
#EXPAT(30) #
*****
I
*****
# SET PATTERNS #
# TEST FLAG CLR #
# ECHO FLAG #
*****
I BEGINA(01)
*****
** INITIAL SETUP **
**
*****
#EXPAT1 #
*****
I
*****
# SET UP DEVICE AND #
# VECTOR DEFAULT #
# ADDRESSES #
*****
I DPMSG1
*****
/ TYPE PROGRAM \
/ SUB-TITLE \
*****
I INPAR(02)
*****
** GO ASK FOR AND **
** SET UP DEVICE AND **
** VECTOR ADDRESSES **
*****
EXPAT2 I ECHO2(08)
*****
** GO ASK FOR AND **
** SET UP LINE NO. **
** AND "LPR" **
*****
I SUCLMK(28)
*****
** GO SET UP **
** CHAR LENGTH **
** MASK **
*****
***
#A#
***
I DPMSG2
*****
/ ASK USER FOR \
/ BUFFER SIZE \
/ (1-512) \
*****
I

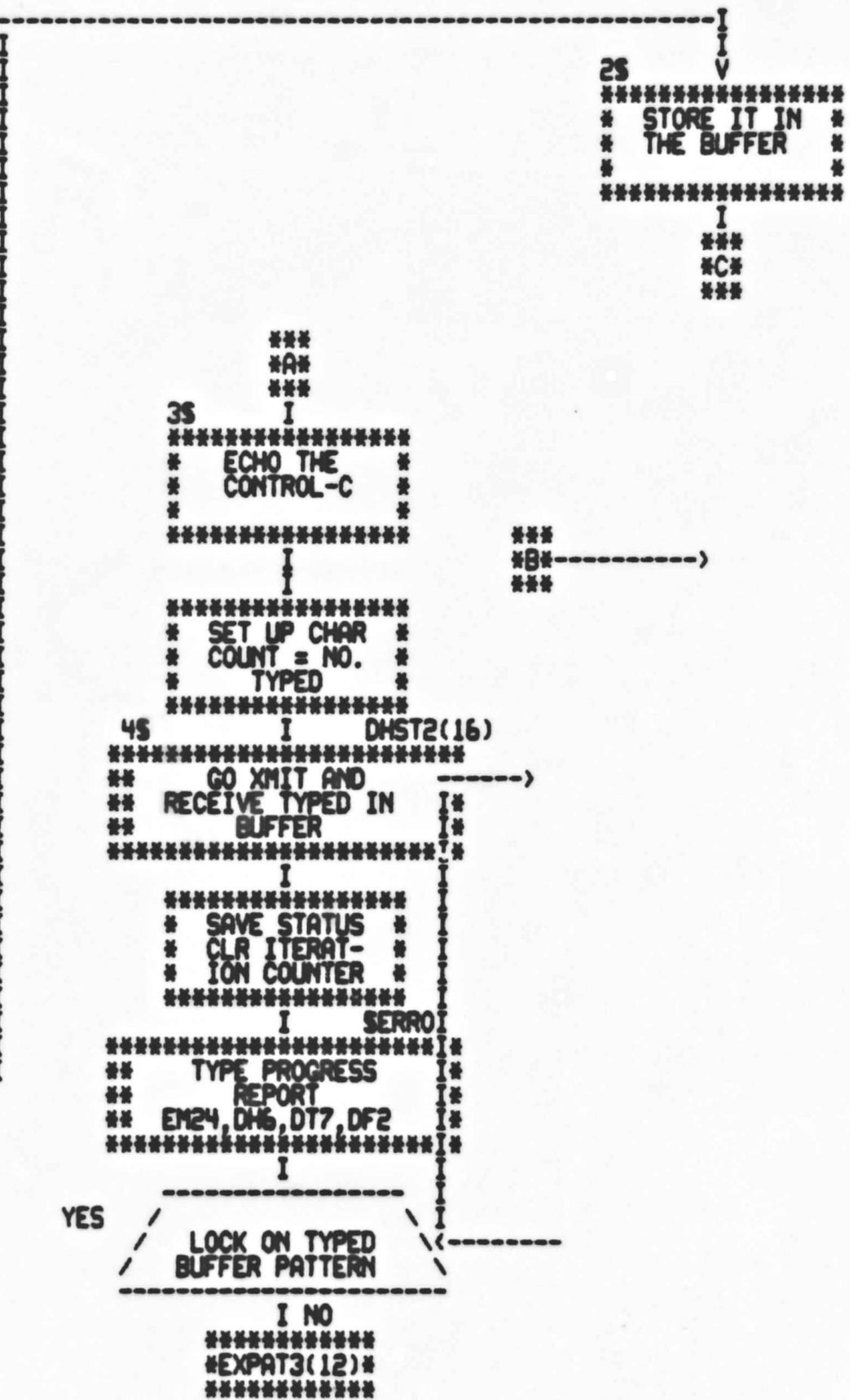
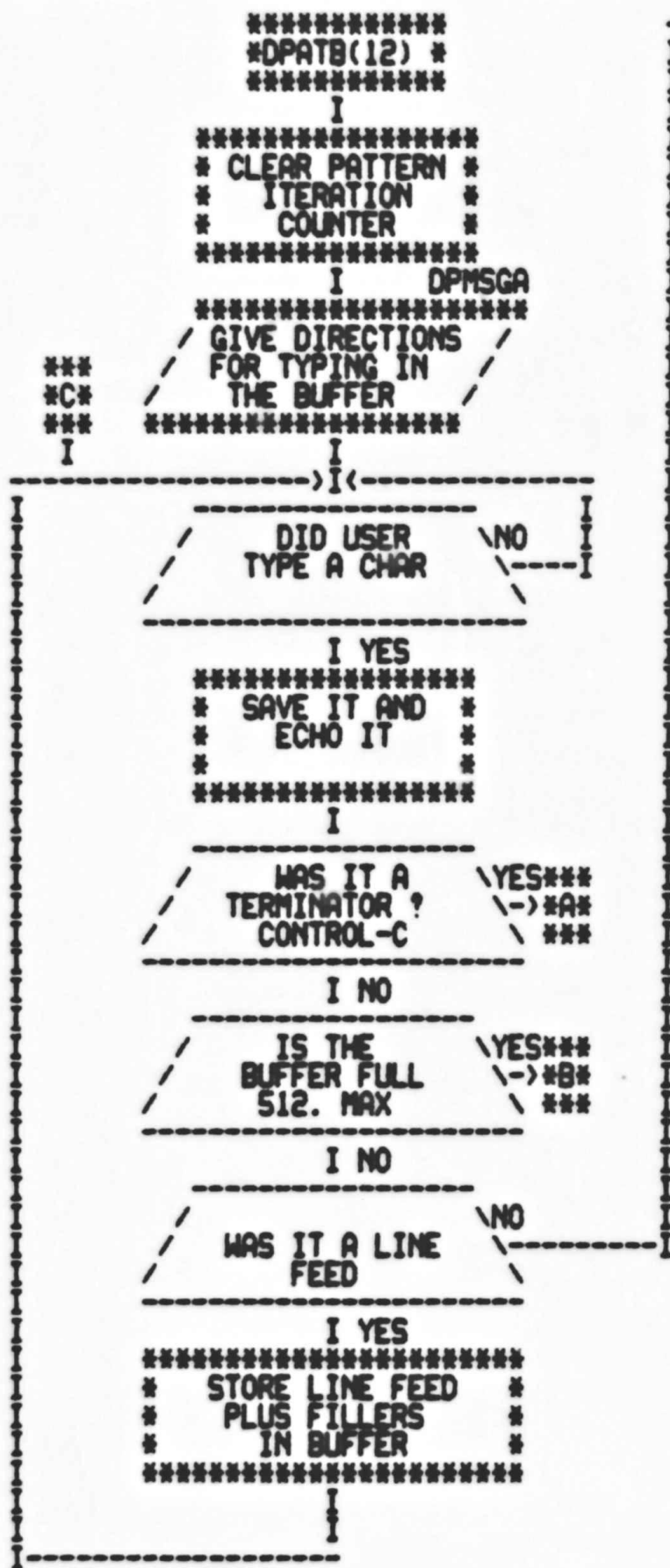
```











 #B*

YES

```

*****
#DHST2 #
*****
I      DHSET1(21)
*****
** GO SET UP TO **
** XMIT AND RECEIVE **
** PATTERN. **
*****
I
** ACTIVATE SEL- **
** ECTED LINE **
** CLEAR PSW **
*****

```

```

***
#B#
***
I
*****
** RETURN TO APPROP- **
** RIATE DRIVE **
** ROUTINE VIA RTS **
*****

```

NOTE: APPROPRIATE ROUTINES

DPATA
 DPATU
 DPATD
 DPATR
 DPATB
 DPATS

```

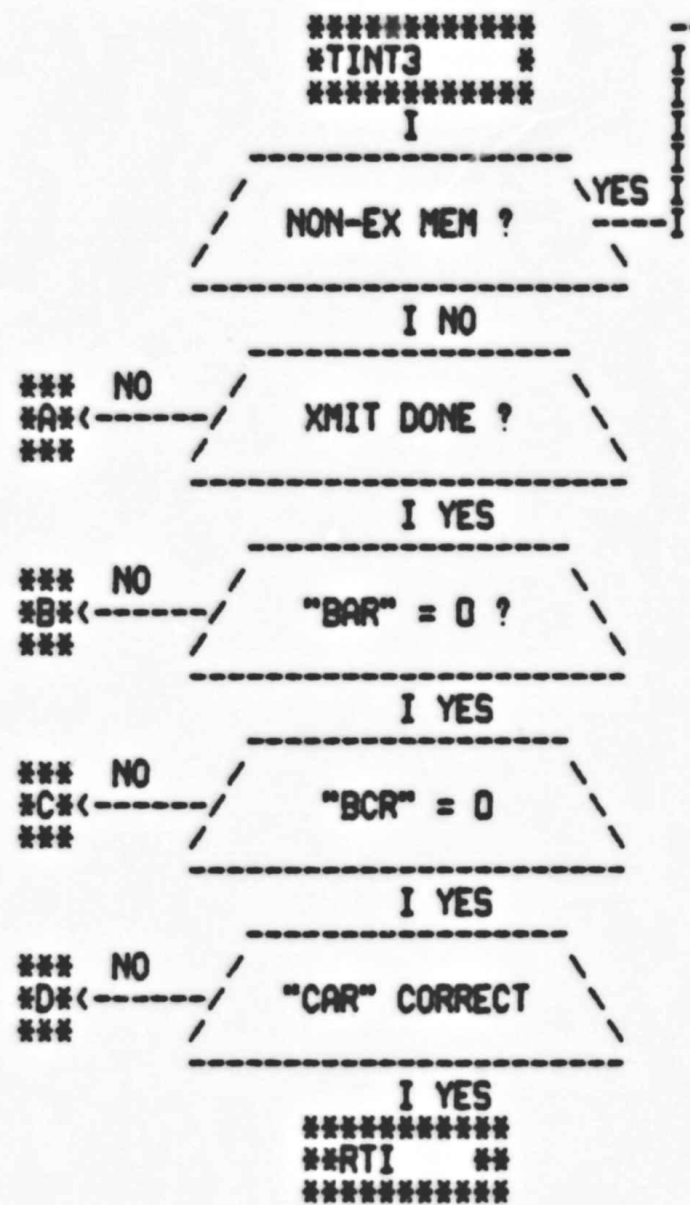
I
V
*****
# INIT TIMERS #
# "A" AND "B" #
*****
I
RECEIVER DONE
I NO
TIMEIT(25)
*****
** CALL TIMER **
*****
I
TIMEOUT ?
I YES
*****
** CLEAR DH11 **
** SAVE ERROR **
** INFO **
*****
I ERROR
*****
** REPORT INTR **
** WAIT TIMEOUT **
** EM25, DH7, DT10, DF2 **
*****
I
*****
# FIX SP #
*****
I
*****
#EXPAT3(12)#
*****

```

```

I
V
CKERP
*****
#SET UP POINTERS#
# TO CHECK #
# BUFFERS #
*****
I
*****
# CHECK ONE #
# WORD #
# [TBUF]=[RBUF] #
*****
I
DATA COMPARE ERROR ?
I NO
CHECKED ALL CHARS ?
I NO
*****
# UPDATE #
# POINTERS #
*****
I
***
#A#
***
I
*****
# SET UP ERROR #
# INFORMATION #
*****
I ERROR
*****
** REPORT DATA **
** COMPARE ERROR **
** EM11, DH2, DT2, DF2 **
*****

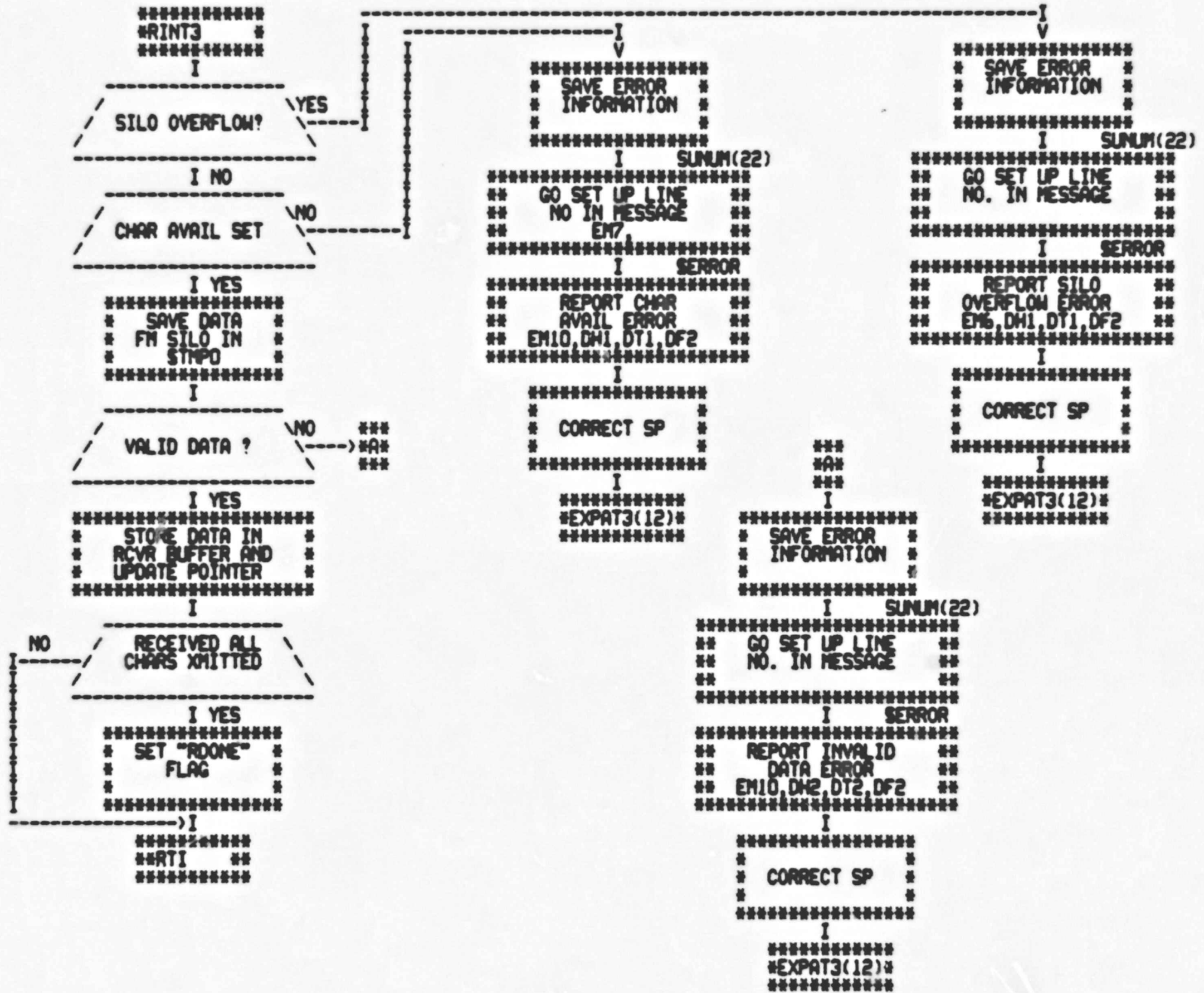
```

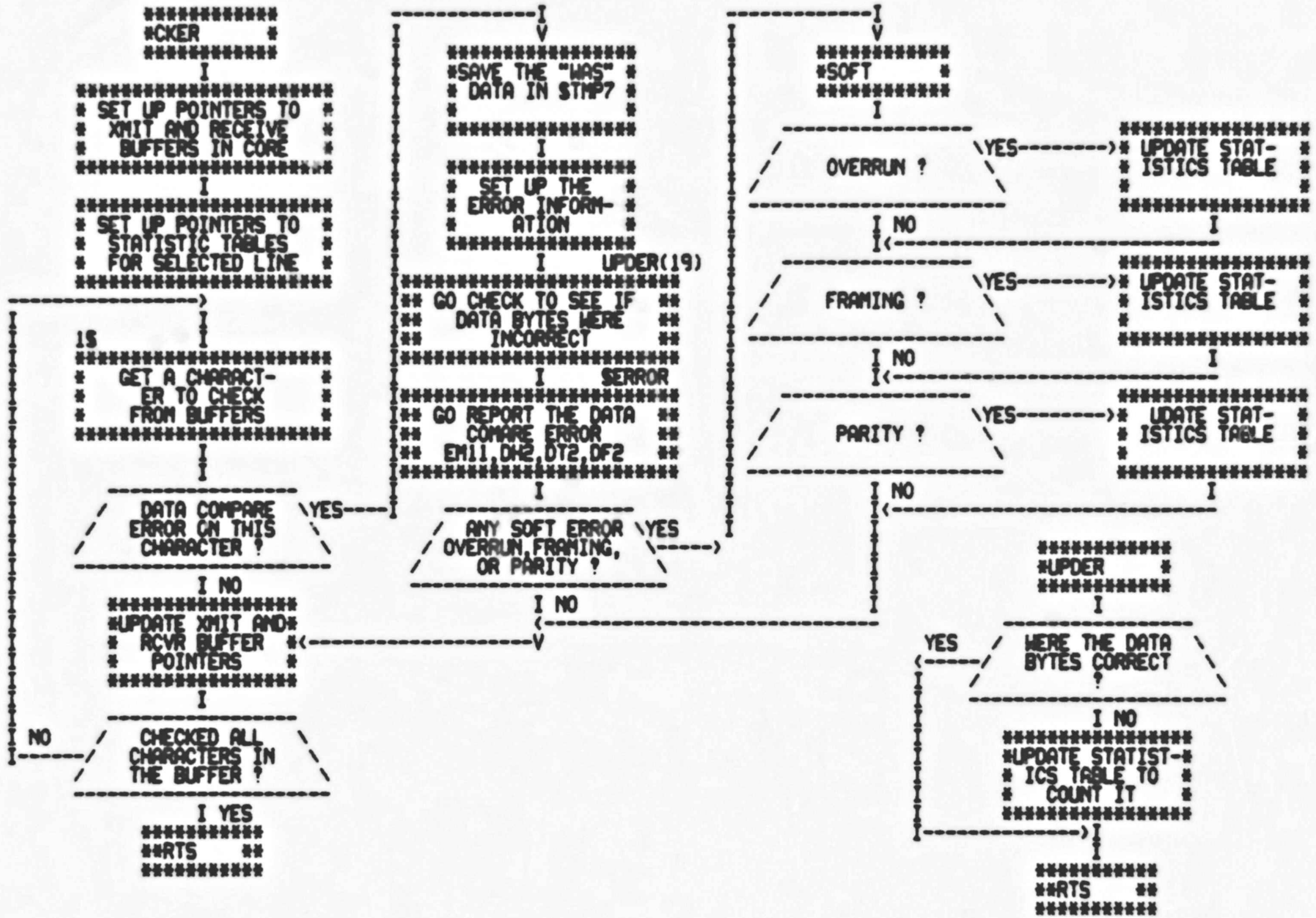


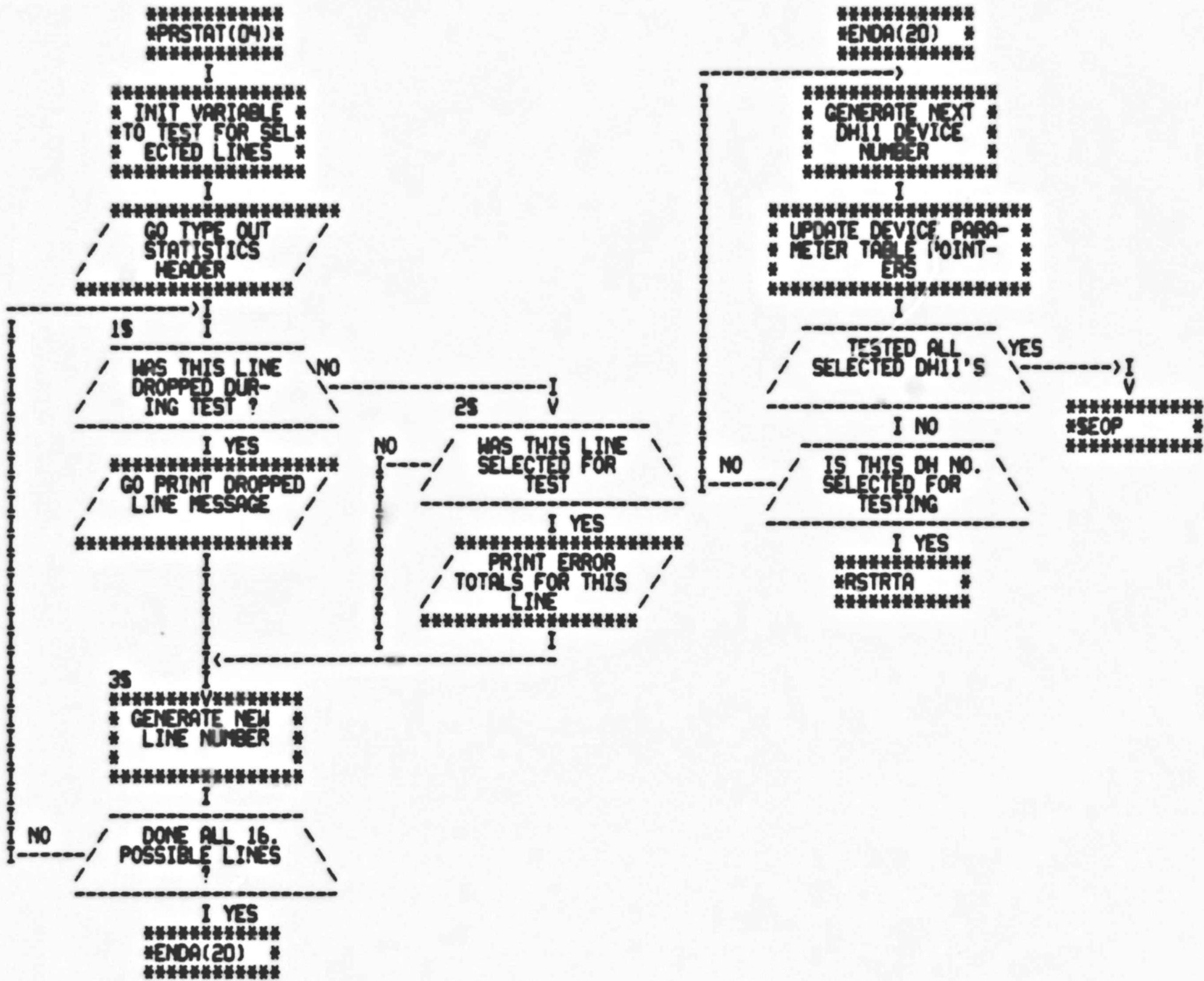
```
-----I  
V  
*****  
# SAVE ERROR #  
# INFORMATION #  
*****  
I SUNUM(22)  
*****  
## GO SET UP LINE ##  
## NO. IN MESSAGE ##  
*****  
I SERROR  
*****  
## REPORT NON-EX- ##  
## MEMORY ERROR ##  
## EM1,DH1,DT1,DF2 ##  
*****  
I  
*****  
# FIX SP #  
*****  
I  
***  
##C#  
##E#  
***  
I  
*****  
# SAVE ERROR #  
# INFORMATION #  
*****  
I SUNUM(22)  
*****  
## GO SET UP LINE ##  
## NO. IN MESSAGE ##  
*****  
I SERROR  
*****  
## REPORT "BCR" ##  
## REG PROBLEM ##  
## EM1,DH1,DT1,DF2 ##  
*****  
I  
*****  
# FIX SP #  
*****  
I  
***----->I  
##E# *****  
## EXPAT3(12) #  
*** *****
```

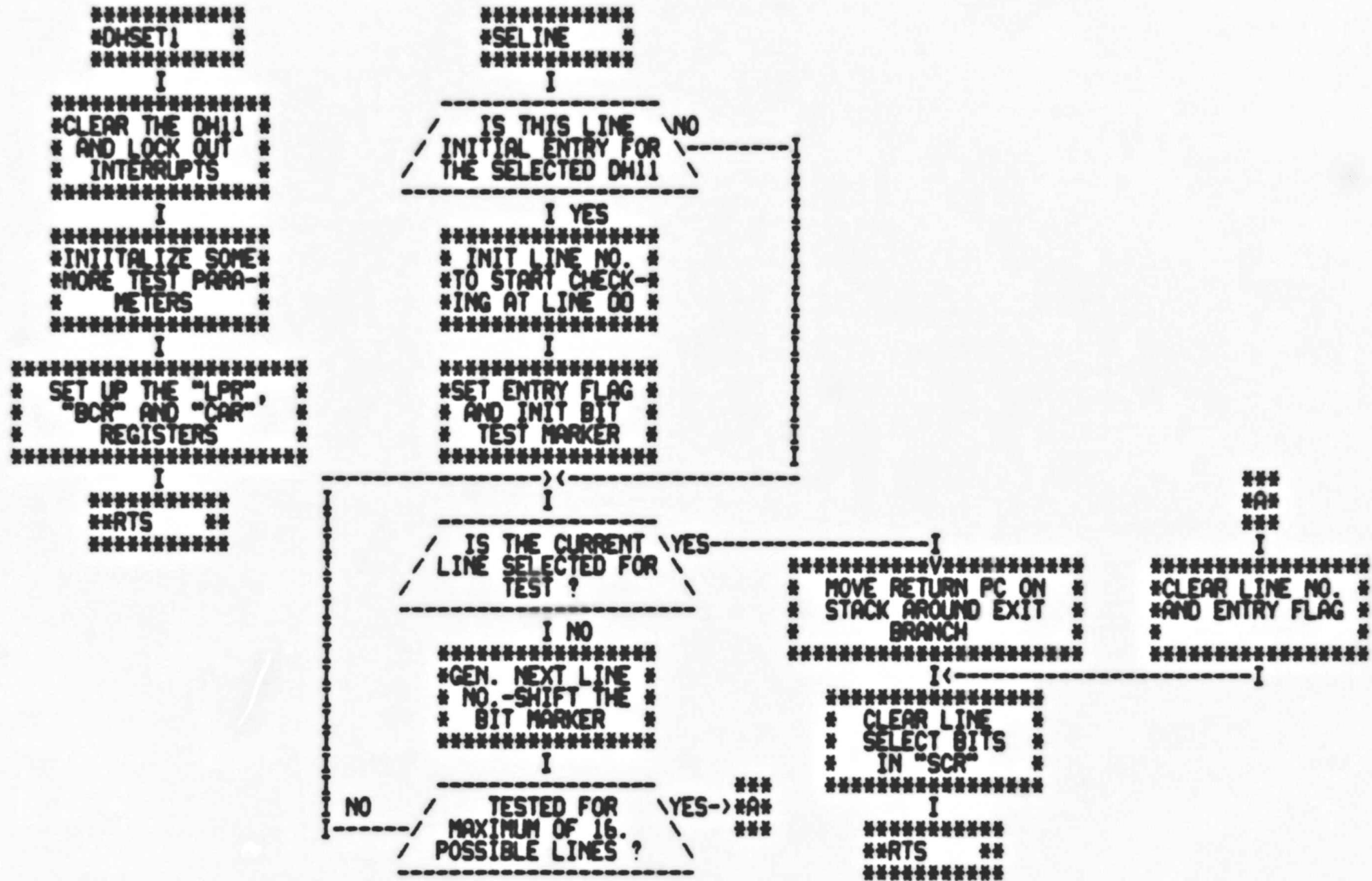
```
***  
##A#  
***  
I  
*****  
# SAVE ERROR #  
# INFORMATION #  
*****  
I SUNUM(22)  
*****  
## GO SET UP LINE ##  
## NO. IN MESSAGE ##  
*****  
I SERROR  
*****  
## REPORT XMITTR ##  
## FALSE INTR ##  
## EM1,DH1,DT1,DF2 ##  
*****  
I  
*****  
# FIX SP #  
*****  
I  
***  
##D#  
***  
I  
*****  
#EXPAT3(12)# *****  
*****  
# SAVE ERROR # *****  
# INFORMATION # *****  
*****  
I SUNUM(22)  
*****  
## GO SET UP LINE ##  
## NO. IN MESSAGE ##  
*****  
I SERROR  
*****  
## REPORT "CAR" ##  
## REG PROBLEM ##  
## EM1,DH1,DT1,DF2 ##  
*****  
I  
*****  
# FIX SP #  
*****  
I  
***  
*****  
#EXPAT3(12)# *****  
*****
```

```
***  
##B#  
***  
I  
*****  
# SAVE ERROR #  
# INFORMATION #  
*****  
I SUNUM(22)  
*****  
## GO SET UP LINE ##  
## NO. IN MESSAGE ##  
*****  
I SERROR  
*****  
## REPORT "BAR" ##  
## REG ERROR ##  
## EM1,DH1,DT1,DF2 ##  
*****  
I  
*****  
# FIX SP #  
*****  
I  
***  
*****  
#EXPAT3(12)# *****  
*****  
# SAVE ERROR # *****  
# INFORMATION # *****  
*****  
I SUNUM(22)  
*****  
## GO SET UP LINE ##  
## NO. IN MESSAGE ##  
*****  
I SERROR  
*****  
## REPORT "CAR" ##  
## REG PROBLEM ##  
## EM1,DH1,DT1,DF2 ##  
*****  
I  
*****  
# FIX SP #  
*****  
I  
***  
*****  
#EXPAT3(12)# *****  
*****
```









```

*****
#SUNUM *
*****
I
*****
* SAVE R0,R1 *
* AND R2 ON THE *
* STACK *
*****
I
*****
* RETRIEVE ADDRESS OF *
* NO. TO BE LOADED FM *
* CALLING ROUTINE *
*****
I
*****
* RETRIEVE ADDRESS OF *
* ASCII BUFFER FROM *
* CALLING ROUTINE *
*****
I
*****
* RETRIEVE ADDRESS OF *
* ASCII BUFFER FROM *
* CALLING ROUTINE *
*****
I
*****
* CONVERT AND LOAD MSD *
* OF NO. INTO MESSAGE *
* BUFFER *
*****
I
*****
* CONVERT AND LOAD LSD *
* OF NO. INTO MESSAGE *
* BUFER *
*****
I
*****
* RESTORE R2,R1 *
* AND R0 FROM *
* STACK *
*****
I
*****
**RTS **
*****

```

```

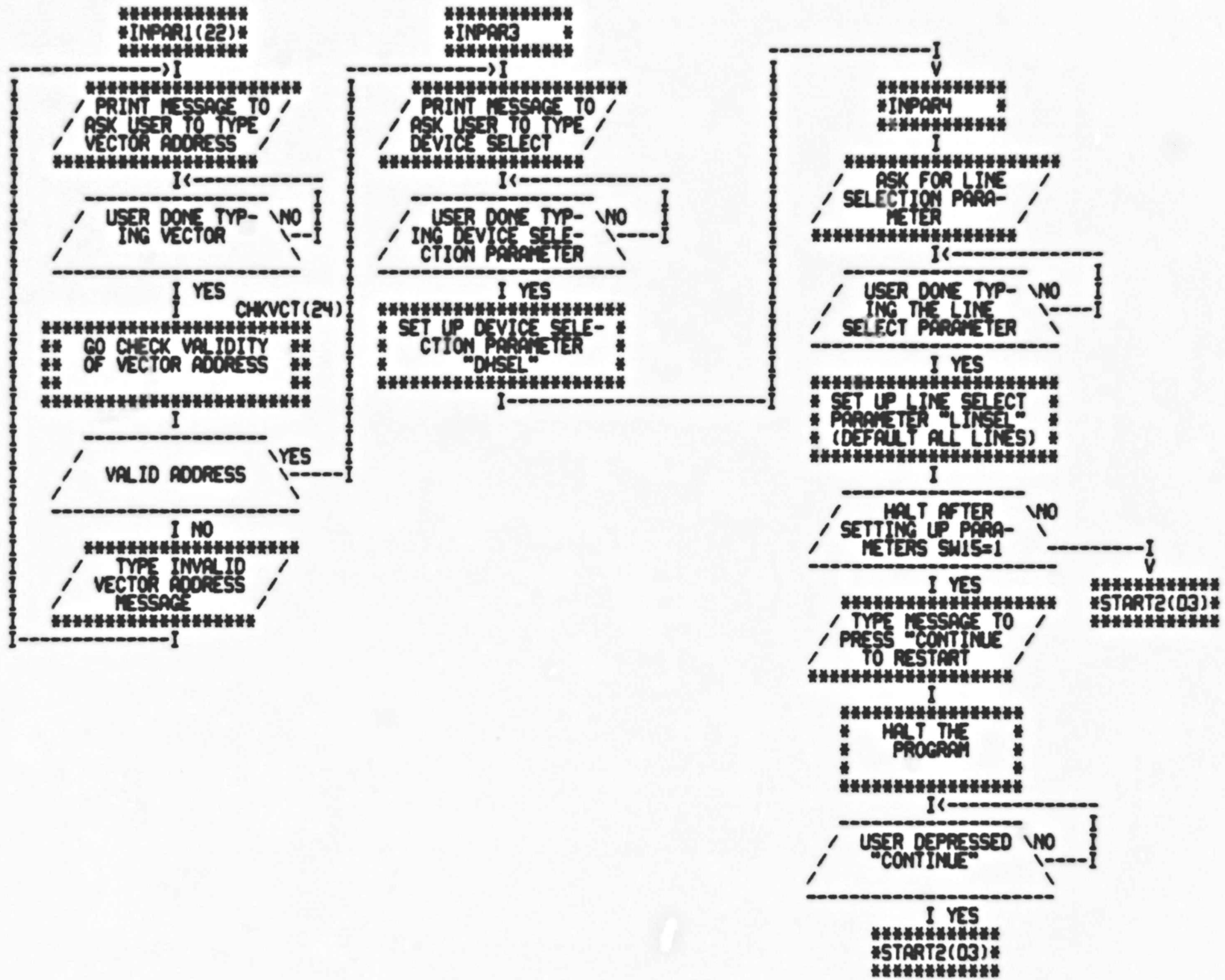
*****
#SUER2 *
*****
I
*****
* SAVE R0 IN *
* SREG0 *
*****
I<-----
*****
* SAVE R1 THRU R4 IN *
* SREG1 THRU SREG4 *
*****
I
*****
**RTS **
*****

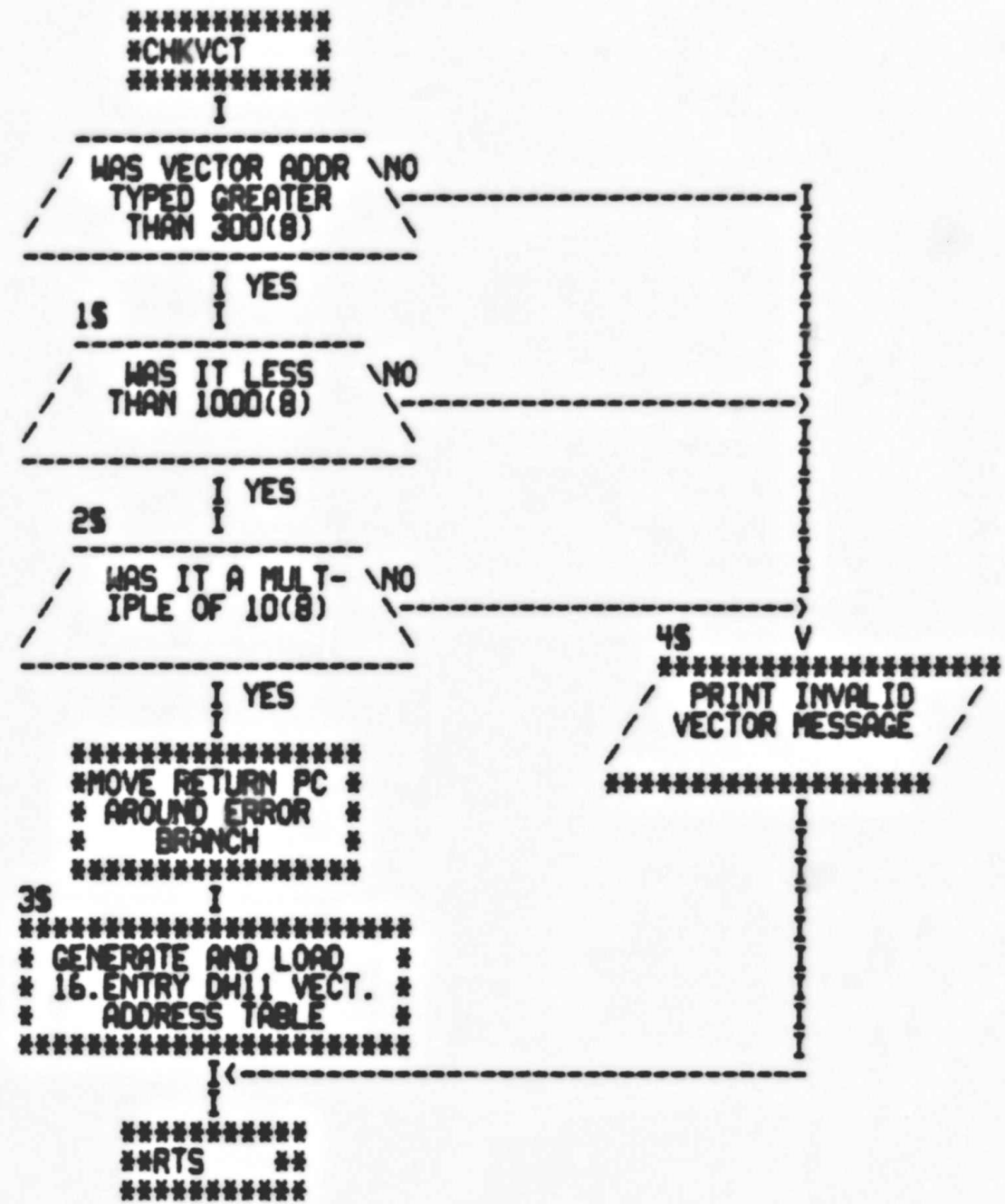
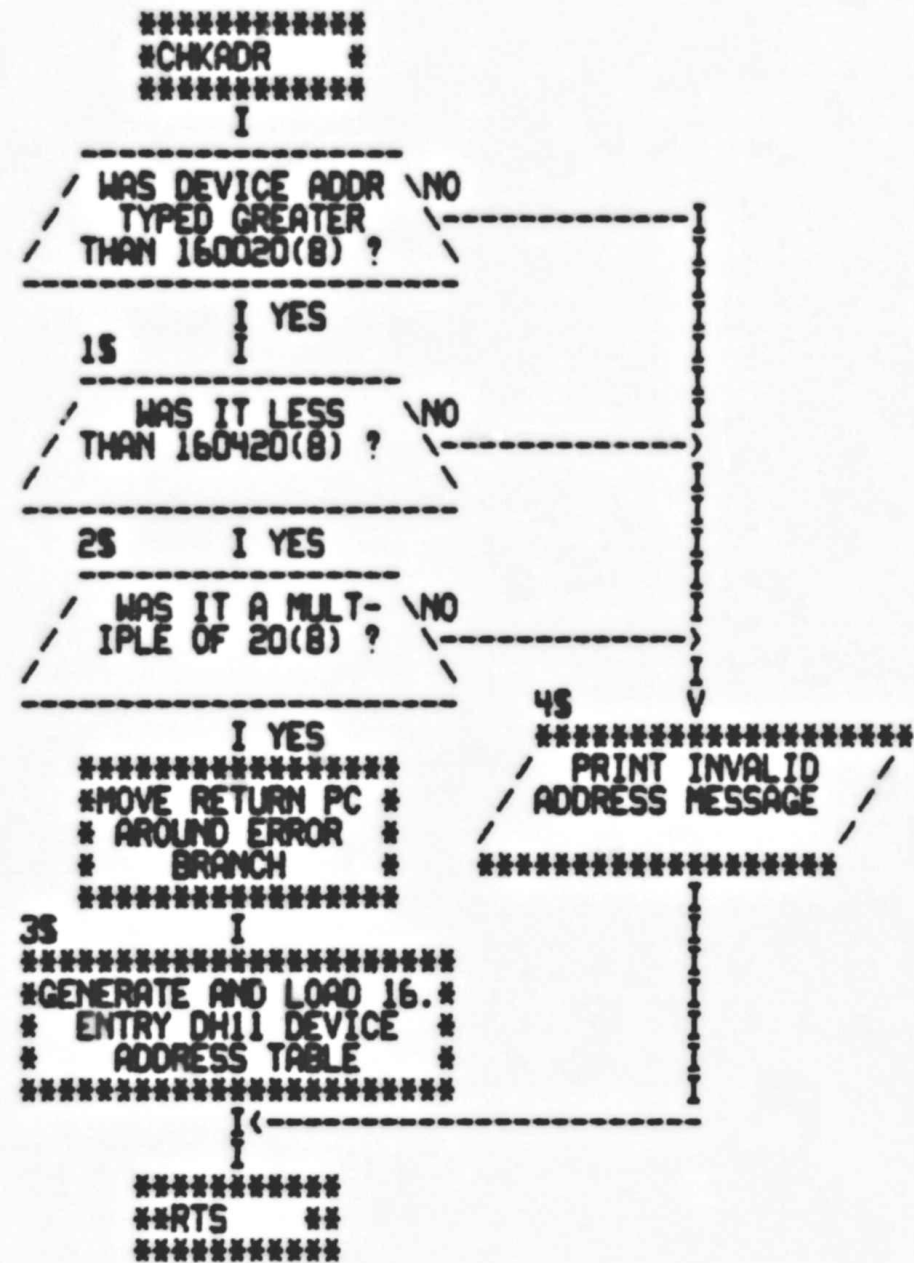
```

```

*****
#SUER1 *
*****
I
*****
#INPAR(02) *
*****
>I
*****
/ PRINT MESSAGE TO /
/ ASK USER TO TYPE /
/ DEVICE ADDRESS /
*****
I<-----
-----I
/ USER DONE TYP- \NO \
/ ING ADDRESS /-----I
-----I
I YES CHKADR(24)
*****
** GO CHECK VALIDITY **
** OF ADDRESS TYPED **
** **
*****
I
-----I
/ VALID ADDRESS \YES-----I
-----I
I NO
*****
/ PRINT INVALID /
/ DEVICE ADDRESS /
/ MESSAGE /
*****
I
-----I
*****
#INPAR(23)*
*****

```





```

*****
#BUSER      *
*****
I
*****
* SAVE THE PSW *
* AND STACK *
* POINTER *
*****
I
*****
* RETRIEVE AND SAVE *
* "TRAPPC" AND "TRAPPS" *
* FROM STACK *
*****
I
*****
* INITIALIZE THE *
* STACK POINTER *
*****
I ERROR
*****
** REPORT THE BUS **
** ERROR **
** EM14, DM4, DT5, DF2 **
*****
I
*****
* RESET THE *
* WORLD AND *
* CLEAR PSW *
*****
I
*****
*REST! *
*****

```

```

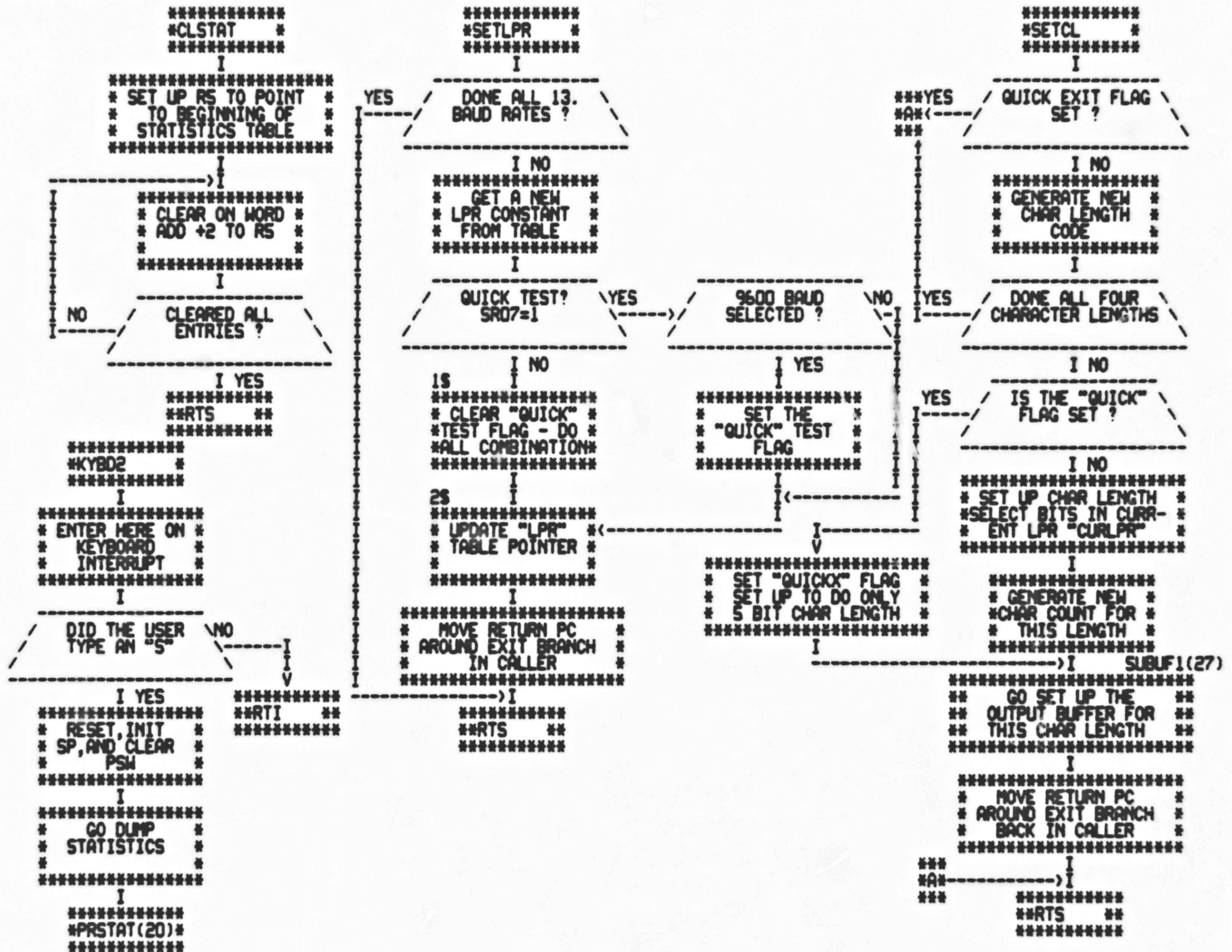
*****
#RESERR     *
*****
I
*****
* SAVE THE PSW *
* AND STACK *
* POINTER *
*****
I
*****
* RETRIEVE AND SAVE *
* "TRAPPC" AND "TRAPPS" *
* FROM STACK *
*****
I
*****
* INITIALIZE *
* STACK POINTER *
*****
I ERROR
*****
** REPORT THE RSVD **
** INSTR ERROR **
** EM15, DM4, DT5, DF2 **
*****
I
*****
* RESET THE *
* WORLD AND *
* CLEAR PSW *
*****
I
*****
*REST! *
*****

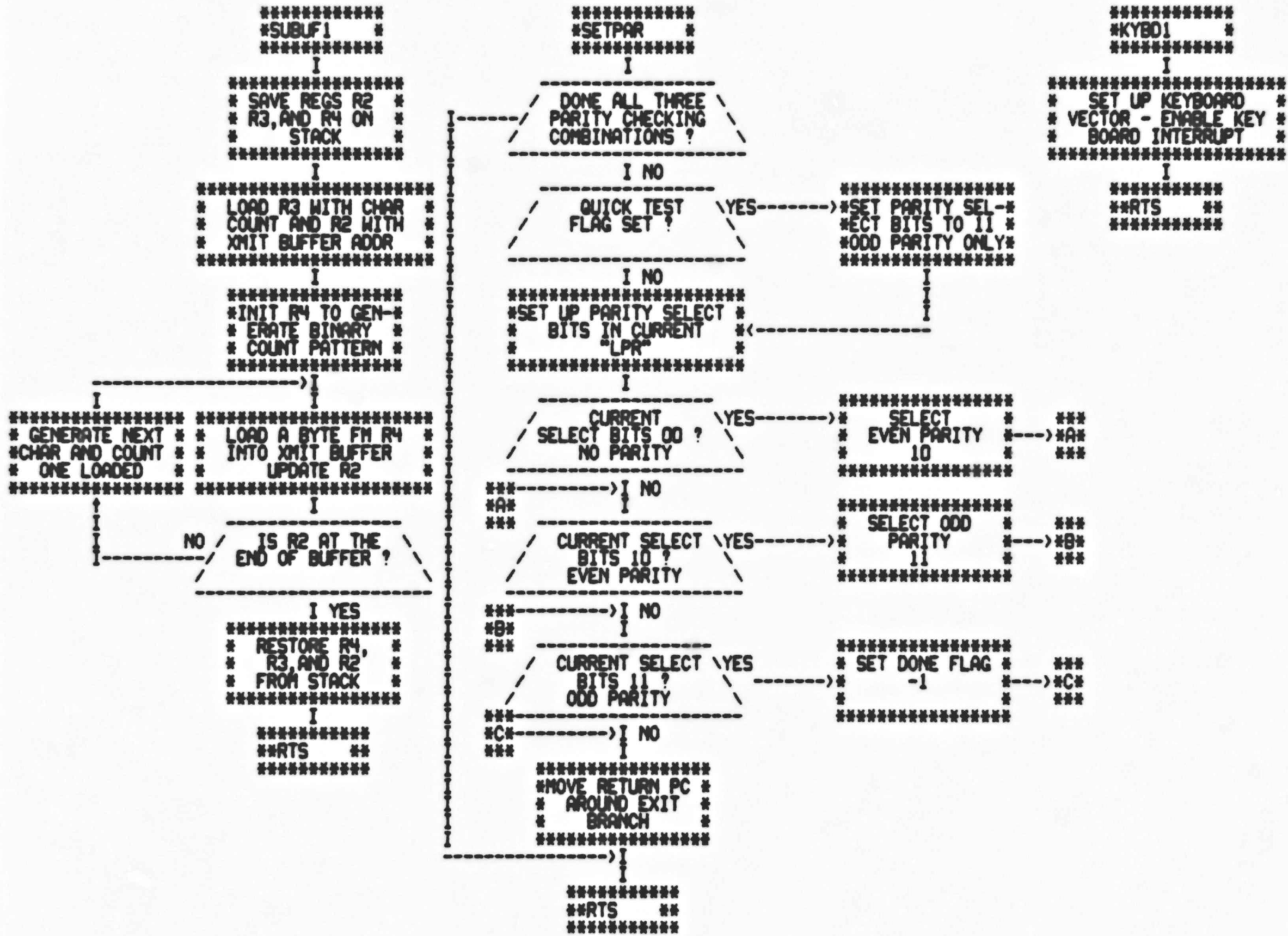
```

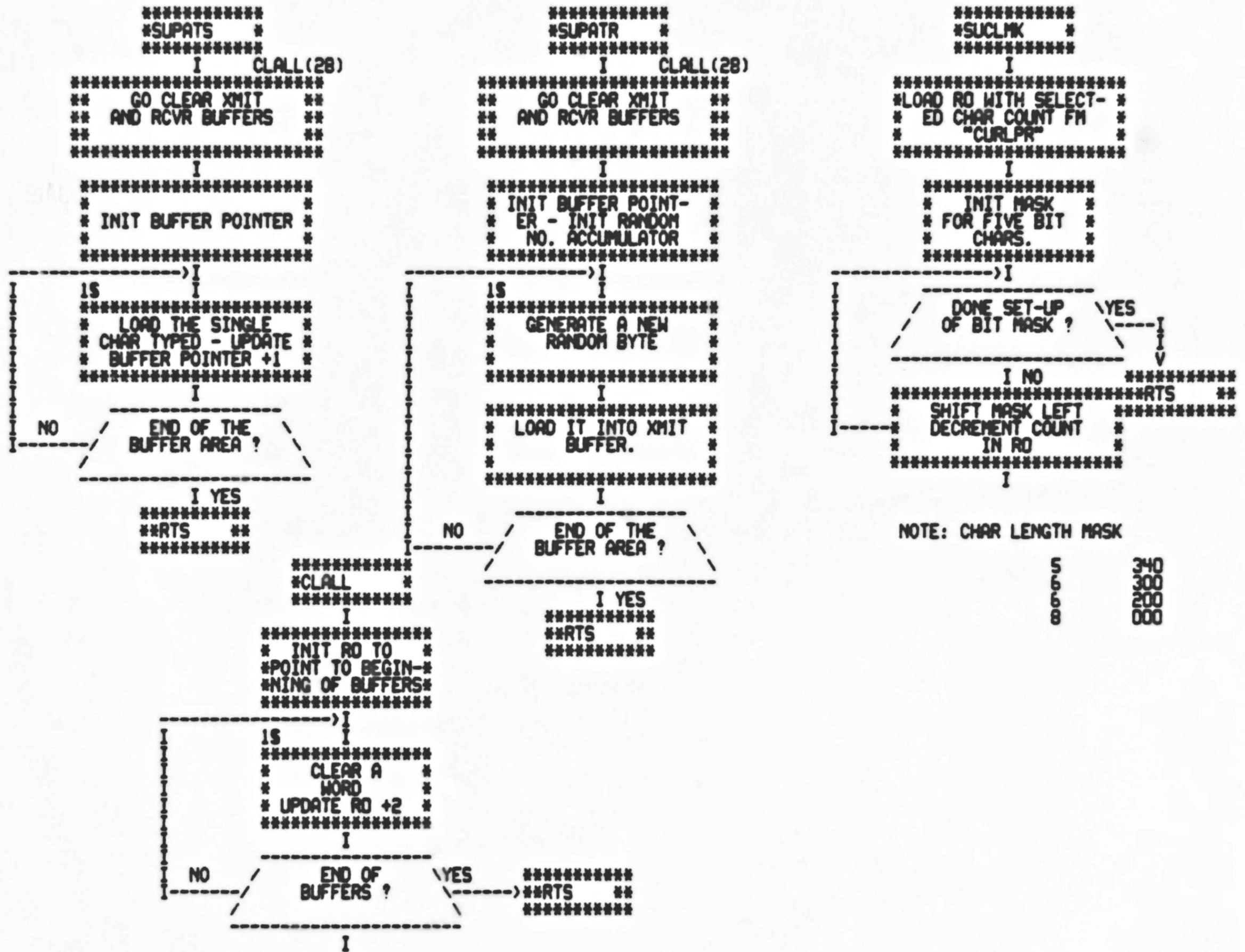
```

*****
#RESTRP     *
*****
I
*****
* GET CURRENT *
* VECTOR ADDRESS *
*****
I
*****
* LOAD VECTOR (RCVR *
* AND XMIT) WITH *
* TRAP CATCHER *
*****
I
*****
**RTS **
*****
*****
*TIMEIT *
*****
I
*****
* INCREMENT *
* TIMER "B" *
*****
I
-----
/ \ NO
 \ /  TIMER "B" = 0
 / \
-----
I YES
*****
* DECREMENT *
* TIMER "A" *
*****
I
-----
/ \ NO
 \ /  TIMER "A" = 0
 / \
-----
I YES
*****
* MOVE RETURN PC *
* AROUND LOOP BRANCH *
* TO CAUSE ERROR *
*****
----->I<-----
*****
**RTS **
*****

```



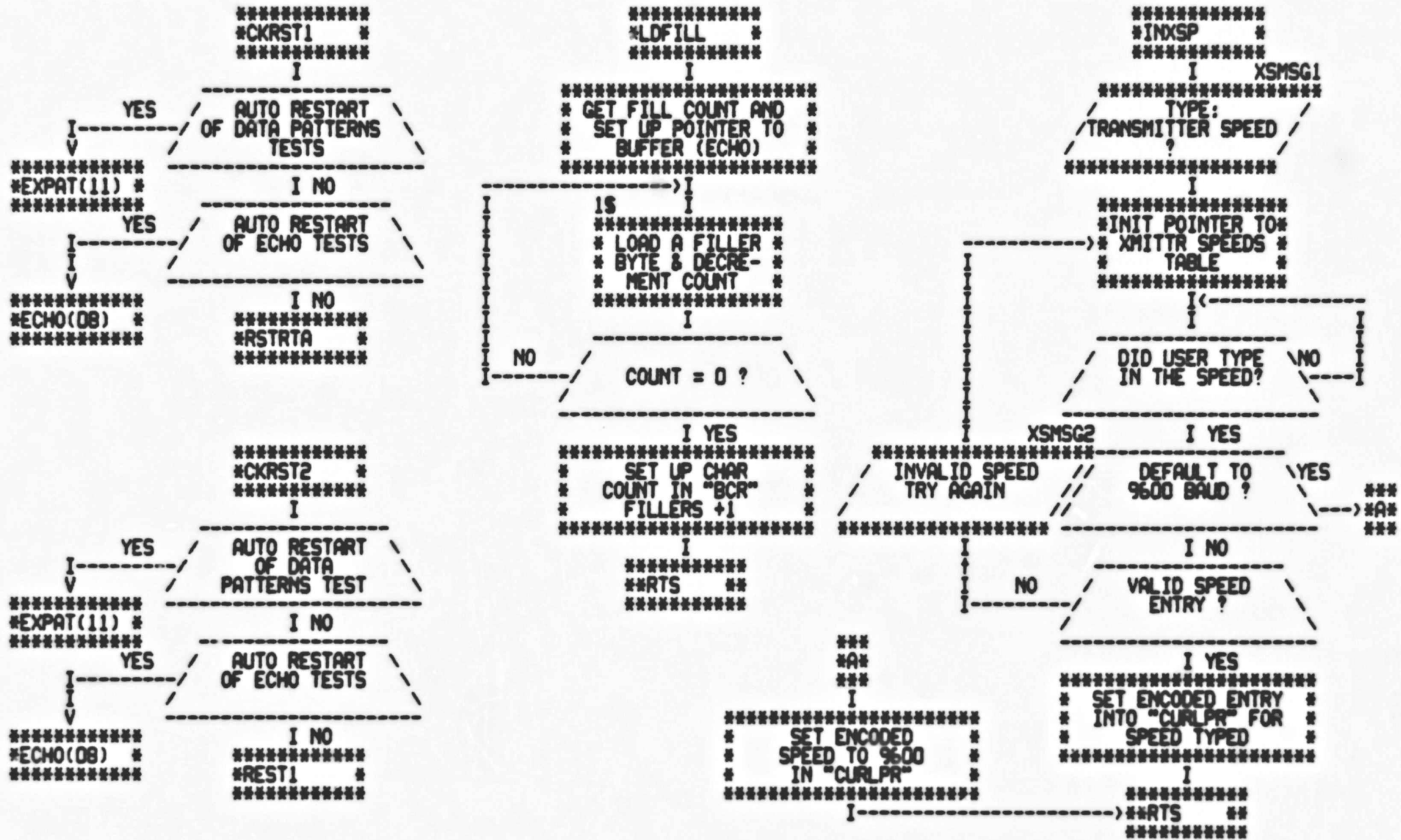




```
*****  
#SUPATA #  
*****  
I CLALL(28)  
*****  
** GO CLEAR XMIT **  
** AND RCVR BUFFERS **  
**  
*****  
I  
*****  
* INIT POINTER TO *  
* BEGINNING OF *  
* BUFFERS *  
*****  
I  
IS  
*****  
* MOVE A 252(8) BYTE *  
* INTO BUFFER-UPDATE *  
* POINTER +1 *  
*****  
I  
-----  
NO / END OF THE /  
-----  
I YES  
*****  
**RTS **  
*****
```

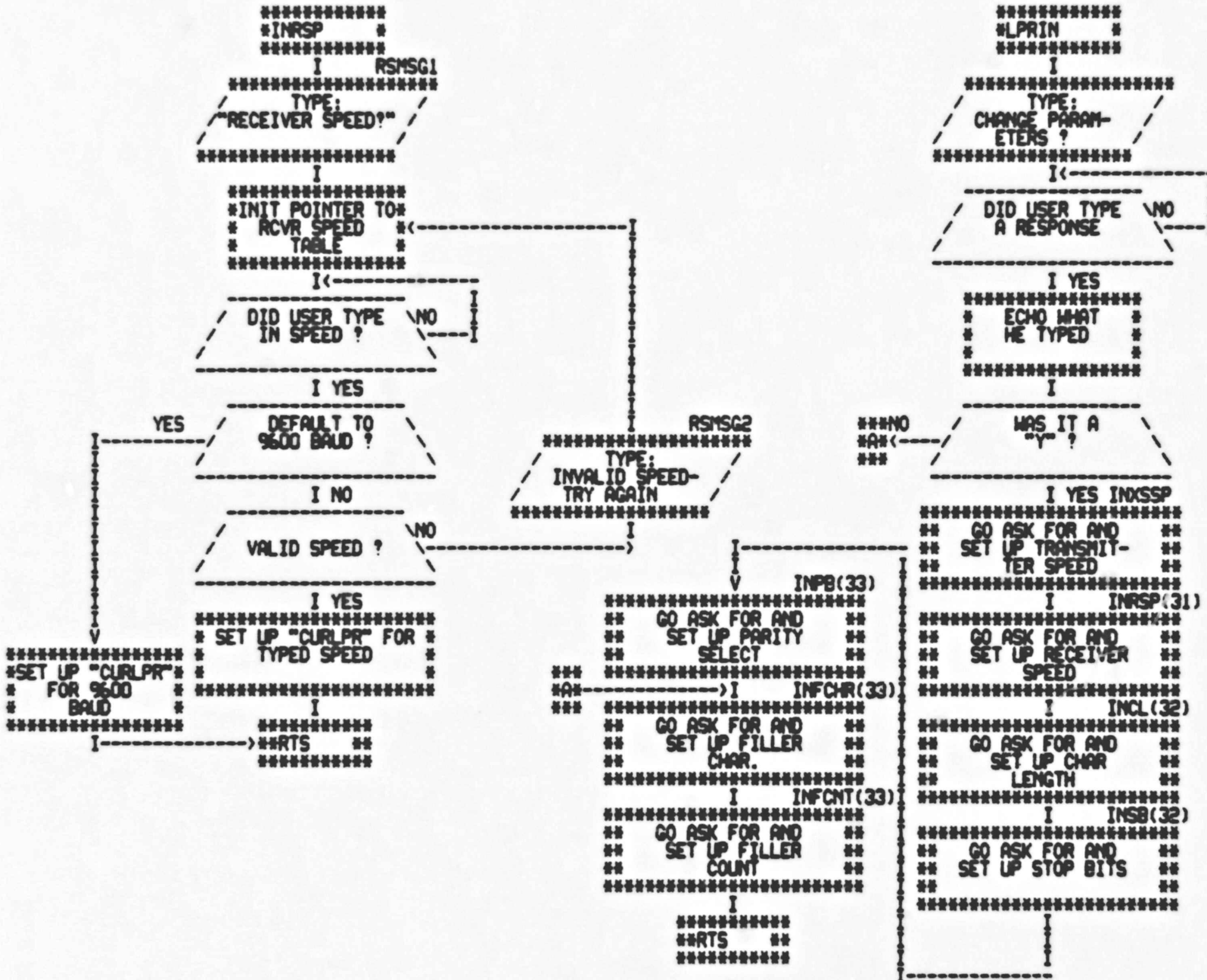
```
*****  
#SUPATU #  
*****  
I CLALL(28)  
*****  
** GO CLEAR XMIT **  
** AND RCVR BUFFERS **  
**  
*****  
I  
*****  
* INIT POINTER TO *  
* BUFFER AND CHAR *  
* GENERATOR *  
*****  
I  
IS  
*****  
* MOV A BYTE INTO *  
* THE BUFFER-UPDATE *  
* POINTER +1 *  
*****  
I  
*****  
* GENERATE *  
* NEXT CHAR *  
*****  
I  
-----  
NO / END OF THE /  
-----  
I YES  
*****  
**RTS **  
*****
```

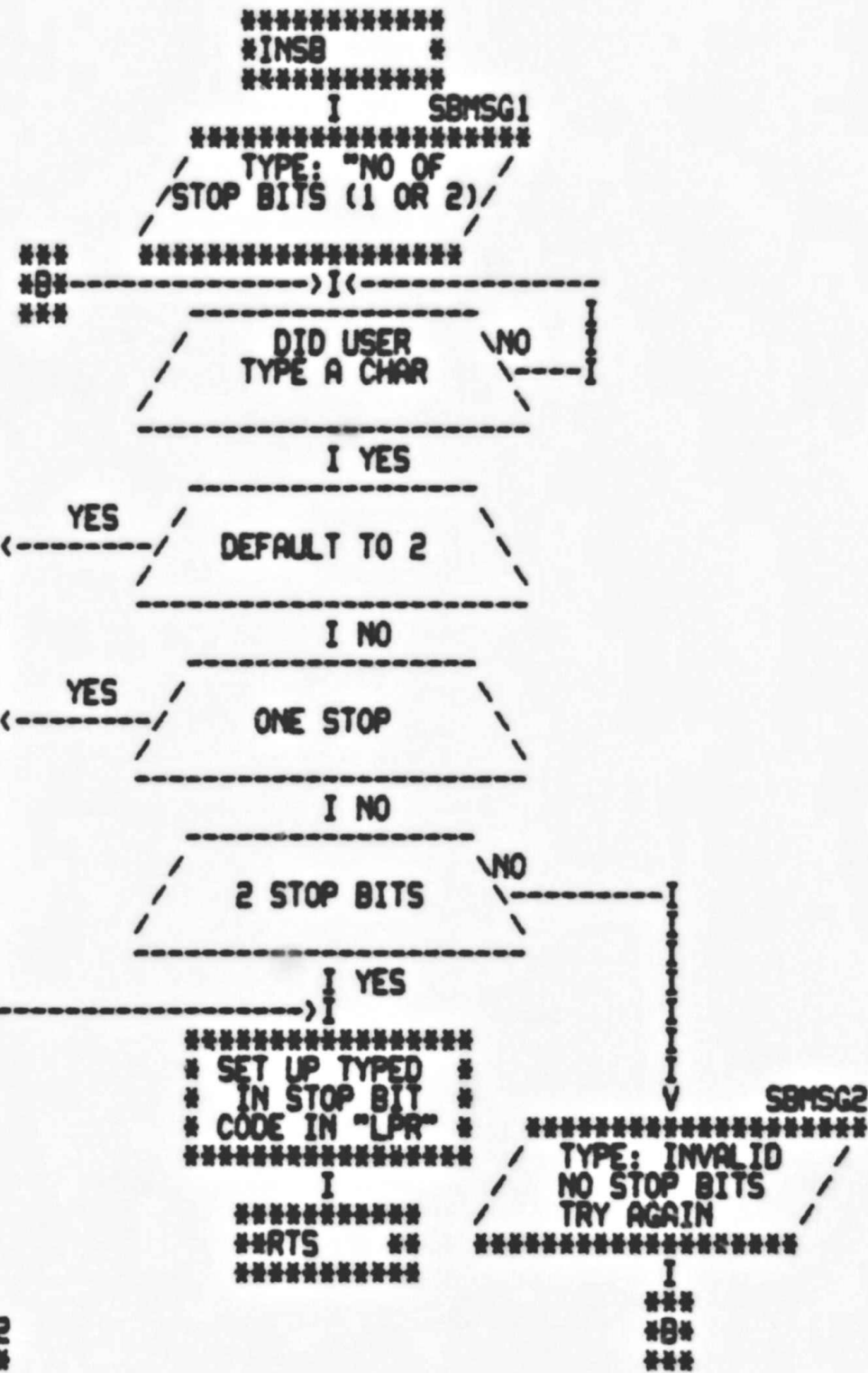
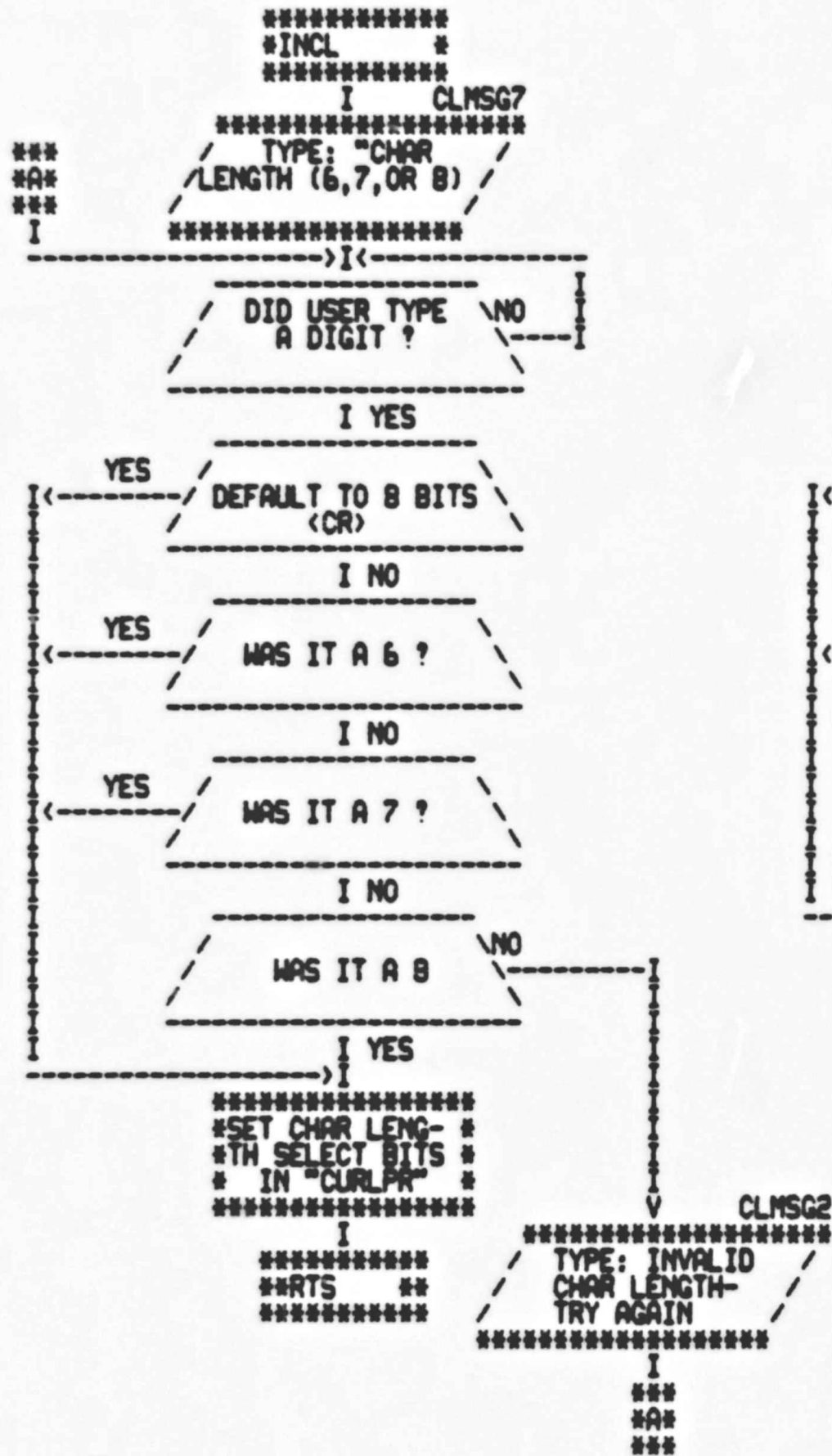
```
*****  
#SUPATD #  
*****  
I CLALL(28)  
*****  
** GO CLEAR XMIT **  
** AND RCVR BUFFERS **  
**  
*****  
I  
*****  
* INIT POINTER TO *  
* BUFFER - INIT CHAR *  
* GENERATOR *  
*****  
I  
IS  
*****  
* MOV A BYTE INTO *  
* THE BUFFER-UPDATE *  
* POINTER +1 *  
*****  
I  
*****  
* GENERATE NEXT *  
* CHARACTER *  
*****  
I  
-----  
NO / END OF THE /  
-----  
I YES  
*****  
**RTS **  
*****
```

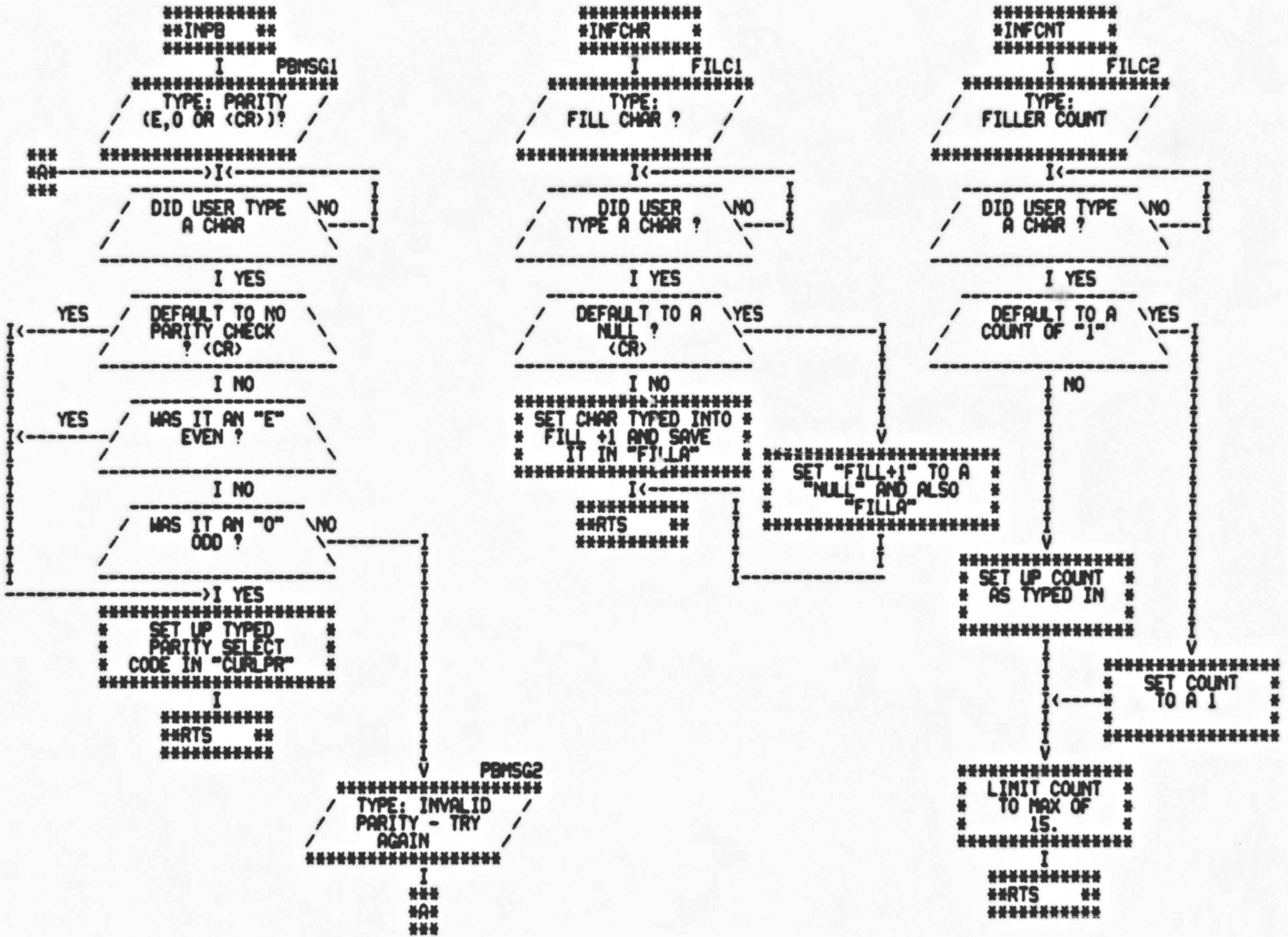


NOTES: "CKRST1" CALLED FROM BUS
ERROR OR RSVD INSTR ERROR
SERVICE ROUTINES

"CKRS 2" CALLED FROM
POWER FAIL SERVICE ROUTINE







```
*****  
#CHPS1 *  
*****  
I  
*****  
* PUSH NEW PSM=000 *  
* AND NEW PC=1S *  
* ON TO STACK *  
*****  
I  
*****  
* DO AN RTI TO *  
* LOAD NEW PSM *  
* GO TO 1S *  
*****  
1S I  
*****  
* DO RTS TO *  
* RETURN TO *  
* CALLER *  
*****  
I  
*****  
**RTS **  
*****
```

```
*****  
#CHPS2 *  
*****  
I  
*****  
* PUSH PSM=340 *  
* AND PC=1S *  
* ON THE STACK *  
*****  
I  
*****  
* DO AN RTI TO *  
* LOAD NEW PSM *  
* GO TO 1S *  
*****  
1S I  
*****  
* DO AN RTS *  
* TO RETURN TO *  
* CALLER *  
*****  
I  
*****  
**RTS **  
*****
```

```
*****  
#SAPS *  
*****  
I  
*****  
* PUSH STACK TO *  
* SAVE SPACE *  
* FOR SAVPSM *  
*****  
I  
*****  
* PUSH THE CURRENT *  
* CONTENTS OF THE *  
* TRAP VECTOR *  
*****  
I  
*****  
* SET UP TRAP VECT- *  
* OR TO GO TO 1S *  
* *  
*****  
I  
*****  
* DO A "TRAP" *  
* INSTRUCTION *  
* *  
*****  
1S I  
*****  
*SAVE PSM, PUSH *  
* 2S ON STACK *  
* DO RTI *  
*****  
2S I  
*****  
* RESTORE TRAP VECTOR *  
* POP STACK (SAVPS) *  
* INTO STIPO *  
*****  
I  
*****  
**RTS **  
*****
```

NOTE: I "CHPS1", "CHPS2", AND "SAPS" ARE CALLED WHENEVER
THE MAINLINE CODE HAS TO CLEAR THE PSM, LOCK OUT INTRs,
AND SAVE THE PSM RESPECTIVELY

THESE ROUTINES ARE REQUIRED FOR LS111 COMPATIBILITY

12	OPERATIONAL SWITCH SETTINGS
14	ACT11 HOOKS
15	APT PARAMETER BLOCK
16	TRAP CATCHER
(1)	STARTING ADDRESS(ES)
23	BASIC DEFINITIONS
24	COMMON TAGS
(2)	APT MAILBOX-ETABLE
(1)	ERROR POINTER TABLE
237	T1 SUB-PROGRAM 1 - DATA RELIABILITY TESTS
621	SUB-PROGRAM 2 - SINGLE LINE ECHO/CABLE TESTS
846	SUB-PROGRAM THREE - DATA PATTERNS TESTS
1350	END OF PASS ROUTINE
1351	SCOPE HANDLER ROUTINE
1352	ERROR HANDLER ROUTINE
1353	ERROR MESSAGE TIMEOUT ROUTINE
1354	BINARY TO OCTAL (ASCII) AND TYPE
1355	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
1356	TYPE ROUTINE
1357	APT COMMUNICATIONS ROUTINE
1358	TTY INPUT ROUTINE
1359	READ AN OCTAL NUMBER FROM THE TTY
1360	READ A DECIMAL NUMBER FROM THE TTY
1361	TRAP DECODER
(3)	TRAP TABLE
1362	POWER DOWN AND UP ROUTINES
2130	DH11 PROGRAM CONSTANTS AND VARIABLES
2349	STANDARD ERROR MESSAG BUFFERS
2467	MISCELANEOUS TABLES AND MESSAGE AND DATA BUFFERS


```

(1) ;INTERFACE SPEC.
(1)
(1) 000000 SAPTHD:
(1) 000000 000000 SHIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
(1) 000002 001230 SMBADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
(1) 000004 001604 STSTM: .WORD 1604 ;;RUN TIM OF LONGEST TEST
(1) 000006 001604 SPASTM: .WORD 1604 ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
(1) 000010 001604 SUNITH: .WORD 1604 ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
(1) 000012 000052 .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-$TABLE(WORDS)
16
(1) .SBTTL TRAP CATCHER
(1)
(1) 000000 .=0
(1) ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
(1) ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1) ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
(1)
(1) 000174 000174 =174
(1) 000174 000000 DISPREG: .WORD 0 ;;SOFTWARE DISPLAY REGISTER
(1) 000176 000000 SMREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER
(1)
(1) .SBTTL STARTING ADDRESS(ES)
(1) 000200 000137 014322 JMP @#INPARX ;;JUMP TO STARTING ADDRESS OF PROGRAM
17
18 000204 000137 001624 JMP @#BEGIN ;;BEGIN EXECUTION WITH DEFAULT PARAMETERS
19 000210 000137 014334 JMP @#INPARC ;;INPUT PARAMETERS - DEVICE SELECTION ONLY
20 000214 000137 004604 JMP @#ECHO ;;GO START LINE ECHO TESTS
21 000220 000137 006004 JMP @#EXPAT ;;GO START DATA PATTERNS TESTS

```



```

24
(1) ;*****
(1) .SBTTL COMMON TAGS
(1) ;#THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(1) ;#USED IN THE PROGRAM.
(1)
(1)      001100      SCMTAG:      .=1100      ;;START OF COMMON TAGS
(1) 001100 000000      STSTNM:      .WORD      0      ;;CONTAINS THE TEST NUMBER
(1) 001102      000      SERFLG:      .BYTE      0      ;;CONTAINS ERROR FLAG
(1) 001103      000      SICNT:      .WORD      0      ;;CONTAINS SUBTEST ITERATION COUNT
(1) 001104 000000      SLPADR:      .WORD      0      ;;CONTAINS SCOPE LOOP ADDRESS
(1) 001106 000000      SLPERR:      .WORD      0      ;;CONTAINS SCOPE RETURN FOR ERRORS
(1) 001110 000000      SERTTL:      .WORD      0      ;;CONTAINS TOTAL ERRORS DETECTED
(1) 001112 000000      SITEMB:      .BYTE      0      ;;CONTAINS ITEM CONTROL BYTE
(1) 001114      000      SERMAX:      .BYTE      1      ;;CONTAINS MAX. ERRORS PER TEST
(1) 001115      001      SERRPC:      .WORD      0      ;;CONTAINS PC OF LAST ERROR INSTRUCTION
(1) 001116 000000      SGPADR:      .WORD      0      ;;CONTAINS ADDRESS OF 'GOOD' DATA
(1) 001120 000000      SBDADR:      .WORD      0      ;;CONTAINS ADDRESS OF 'BAD' DATA
(1) 001122 000000      SGGDAT:      .WORD      0      ;;CONTAINS 'GOOD' DATA
(1) 001124 000000      SBODAT:      .WORD      0      ;;CONTAINS 'BAD' DATA
(1) 001126 000000      .WORD      0      ;;RESERVED--NOT TO BE USED
(1) 001130 000000      .WORD      0
(1) 001132 000000      .WORD      0
(1) 001134 000000      .WORD      0
(1) 001136 177570      SWR:      .WORD      DSWR      ;;ADDRESS OF SWITCH REGISTER
(1) 001140 177570      DISPLAY:      .WORD      DDISP      ;;ADDRESS OF DISPLAY REGISTER
(1) 001142 177560      STKS:      177560      TTY KBD STATUS
(1) 001144 177562      STKB:      177562      TTY KBD BUFFER
(1) 001146 177564      STPS:      177564      TTY PRINTER STATUS REG. ADDRESS
(1) 001150 177566      STPB:      177566      TTY PRINTER BUFFER REG. ADDRESS
(1) 001152      000      SNUL:      .BYTE      0      ;;CONTAINS NULL CHARACTER FOR FILLS
(1) 001153      002      SFILLS:      .BYTE      2      ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
(1) 001154      012      SFILLC:      .BYTE      12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
(1) 001155      000      STPFLG:      .BYTE      0      ;;"TERMINAL AVAILABLE" FLAG (BIT(07)=0=YES)
(1) 001156 000000      SREGAD:      .WORD      0      ;;CONTAINS THE ADDRESS FROM
(1)      .WORD      0      WHICH (SREGAD) WAS OBTAINED
(3) 001160 000000      SREG0:      .WORD      0      ;;CONTAINS ((SREGAD)+0)
(3) 001162 000000      SREG1:      .WORD      0      ;;CONTAINS ((SREGAD)+2)
(3) 001164 000000      SREG2:      .WORD      0      ;;CONTAINS ((SREGAD)+4)
(3) 001166 000000      SREG3:      .WORD      0      ;;CONTAINS ((SREGAD)+6)
(3) 001170 000000      SREG4:      .WORD      0      ;;CONTAINS ((SREGAD)+10)
(3) 001172 000000      SREG5:      .WORD      0      ;;CONTAINS ((SREGAD)+12)
(3) 001174 000000      SREG6:      .WORD      0      ;;CONTAINS ((SREGAD)+14)
(3) 001176 000000      SREG7:      .WORD      0      ;;CONTAINS ((SREGAD)+16)
(3) 001200 000000      STMP0:      .WORD      0      ;;USER DEFINED
(3) 001202 000000      STMP1:      .WORD      0      ;;USER DEFINED
(3) 001204 000000      STMP2:      .WORD      0      ;;USER DEFINED
(3) 001206 000000      STMP3:      .WORD      0      ;;USER DEFINED
(3) 001210 000000      STMP4:      .WORD      0      ;;USER DEFINED
(3) 001212 000000      STMP5:      .WORD      0      ;;USER DEFINED
(3) 001214 000000      STMP6:      .WORD      0      ;;USER DEFINED
(3) 001216 000000      STMP7:      .WORD      0      ;;USER DEFINED

```

(1) 001220 000000
(1) 001222 000000
(1) 001224 077
(1) 001225 015
(1) 001226 000012

\$TIMES: 0
\$ESCAPE: 0
\$QUES: .ASCII /?/
\$CRLF: .ASCII <15>
\$LF: .ASCII <12>

::: MAX. NUMBER OF ITERATIONS
::: ESCAPE ON ERROR ADDRESS
::: QUESTION MARK
::: CARRIAGE RETURN
::: LINE FEED

(2) 001312 000000
(2) 001314 000000
(2) 001316 000000
(2) 001320 000000
(2) 001322 000000
(2) 001324 000000
(2) 001326 000000
(2) 001328 000000
(2) 001330 000000
(2) 001332 000000
(2) 001334 000000
(2) 001336 000000
(2) 001338 000000
(2) 001340 000000
(2) 001342 000000
(2) 001344 000000
(2) 001346 000000
(2) 001350 000000
(2) 001352 000000

SCDW2: .WORD ACDW2 ;; CONTROLLER DESCRIPTION WORD#2
SDOW0: .WORD ADDW0 ;; DEVICE DESCRIPTOR WORD#0
SDOW1: .WORD ADDW1 ;; DEVICE DESCRIPTOR WORD#1
SDOW2: .WORD ADDW2 ;; DEVICE DESCRIPTOR WORD#2
SDOW3: .WORD ADDW3 ;; DEVICE DESCRIPTOR WORD#3
SDOW4: .WORD ADDW4 ;; DEVICE DESCRIPTOR WORD#4
SDOW5: .WORD ADDW5 ;; DEVICE DESCRIPTOR WORD#5
SDOW6: .WORD ADDW6 ;; DEVICE DESCRIPTOR WORD#6
SDOW7: .WORD ADDW7 ;; DEVICE DESCRIPTOR WORD#7
SDOW8: .WORD ADDW8 ;; DEVICE DESCRIPTOR WORD#8
SDOW9: .WORD ADDW9 ;; DEVICE DESCRIPTOR WORD#9
SDOW10: .WORD ADDW10 ;; DEVICE DESCRIPTOR WORD#10
SDOW11: .WORD ADDW11 ;; DEVICE DESCRIPTOR WORD#11
SDOW12: .WORD ADDW12 ;; DEVICE DESCRIPTOR WORD#12
SDOW13: .WORD ADDW13 ;; DEVICE DESCRIPTOR WORD#13
SDOW14: .WORD ADDW14 ;; DEVICE DESCRIPTOR WORD#14
SDOW15: .WORD ADDW15 ;; DEVICE DESCRIPTOR WORD#15

001354

SETEND:

.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 SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;; NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
;; 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

001354

SERRTB:

;ERROR TABLE ITEM FOR ERROR 1

001354 020166
001356 020235
001360 020312
001362 020330

EM1 ;; "NON EX MEMORY ERROR - DROPPED LINE # "
DH1 ;; " (PC) CURLPR DEVAOR REGADR WAS S/B"
DT1 ;; "SERRPC,CURLPR,SREG1,SREG2,SREG3,SREG4"
DF2 ;; PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 2

001364 020340
001366 020235
001370 020312
001372 020330

EM2 ;; "TRANSMITTER FALSE INTERRUPT - DROPPED LINE# "
DH1 ;; " (PC) CURLPR DEVAOR REGADR WAS S/B"
DT1 ;; "SERRPC,CURLPR,SREG1,SREG2,SREG3,SREG4"
DF2 ;; PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR 3

15383788453K1-8388YDXX

```

42 001374 020417          EM3          : "BUFFER ACTIVE REGISTER ERROR - DROPPED LINE # "
43 001376 020235          DH1          : " (PC)  CURLPR  DEVAOR  REGADR  WAS  S/B"
44 001400 020312          DT1          : "SERAPC, CURLPR, SREG1, SREG2, SREG3, SREG4
45 001402 020330          DF2          : PRINT ALL OCTAL
46
47 ;ERROR TABLE ITEM FOR ERROR 4
48
49 001404 020477          EM4          : "BYTE COUNT REGISTER ERROR - DROPPED LINE # "
50 001406 020235          DH1          : " (PC)  CURLPR  DEVAOR  REGADR  WAS  S/B"
51 001410 020312          DT1          : "SERAPC, CURLPR, SREG1, SREG2, SREG3, SREG4
52 001412 020330          DF2          : PRINT ALL OCTAL
53
54 ;ERROR TABLE ITEM FOR ERROR 5
55
56 001414 020554          EM5          : "CURRENT ADDRESS REGISTER ERROR - DROPPED LINE # "
57 001416 020235          DH1          : " (PC)  CURLPR  DEVAOR  REGADR  WAS  S/B"
58 001420 020312          DT1          : "SERAPC, CURLPR, SREG1, SREG2, SREG3, SREG4
59 001422 020330          DF2          : PRINT ALL OCTAL
60
61 ;ERROR TABLE ITEM FOR ERROR 6
62
63 001424 020636          EM6          : "SILO OVERFLOW ERROR - DROPPED LINE # "
64 001426 020235          DH1          : " (PC)  CURLPR  DEVAOR  REGADR  WAS  S/B"
65 001430 020312          DT1          : "SERAPC, CURLPR, SREG1, SREG2, SREG3, SREG4
66 001432 020330          DF2          : PRINT ALL OCTAL
67
68 ;ERROR TABLE ITEM FOR ERROR 7
69
70 001434 020705          EM7          : "RECEIVER FALSE INTERRUPT - LINE # "
71 001436 020235          DH1          : " (PC)  CURLPR  DEVAOR  REGADR  WAS  S/B"
72 001440 020312          DT1          : "SERAPC, CURLPR, SREG1, SREG2, SREG3, SREG4
73 001442 020330          DF2          : PRINT ALL OCTAL
74
75 ;ERROR TABLE ITEM FOR ERROR 10
76
77 001444 020761          EM10         : "INVALID DATA IN SILO - DROPPED LINE # "
78 001446 021031          DH2          : " (PC)  CURLPR  CHAR #  WASADR  SHBADR  WAS  S/B"
79 001450 021116          DT2          : "SERAPC, CURLPR, SREG0, SREG1, SREG2, SREG3, SREG4
80 001452 020330          DF2          : PRINT ALL OCTAL
81
82 ;ERROR TABLE ITEM FOR ERROR 11
83
84 001454 021136          EM11         : "DATA ERROR - LINE # "
85 001456 021031          DH2          : " (PC)  CURLPR  CHAR #  WASADR  SHBADR  WAS  S/B"
86 001460 021116          DT2          : "SERAPC, CURLPR, SREG0, SREG1, SREG2, SREG3, SREG4
87 001462 020330          DF2          : PRINT ALL OCTAL
88
89 ;ERROR TABLE ITEM FOR ERROR 12
90
91 001464 021164          EM12         : "TEST TIMEOUT - DROPPED LINE # "
92 001466 021224          DH3          : " (PC)  CURLPR  RTOTAL  XTOTAL  RDONE"
93 001470 021272          DT3          : "SERAPC, CURLPR, STMPO, STMPIRONE"
94 001472 020330          DF2          : PRINT ALL OCTAL
95

```

96					
97					
98	001474	000000	0		: NO MESSAGE
99	001476	000000	0		: NO DATA HEADER
100	001500	021306	DT4		: STMP0, STMP1, STMP2, STMP3, STMP4, STMP5, STMP6
101	001502	021326	DF1		: PRINT ALL DECIMAL
102					
103					
104					
105	001504	021336	EM14		: "BUS ERROR TRAP TO 04"
106	001506	021363	DH4		: " (PC) (PS) (SP) TRAPPC TRAPPS"
107	001510	021432	DT5		: SERRPC, STMP0, SREG6, SREG1, SREG2"
108	001512	020330	DF2		: PRINT ALL OCTAL
109					
110					
111					
112	001514	021446	EM15		: "RSVD INSTR TRAP TO 10"
113	001516	021363	DH4		: " (PC) (PS) (SP) TRAPPC TRAPPS"
114	001520	021432	DT5		: SERRPC, STMP0, SREG6, SREG1, SREG2"
115	001522	020330	DF2		: PRINT ALL OCTAL
116					
117					
118					
119	001524	021474	EM16		: "SINGLE LINE ECHO TEST - INTR WAIT TIMEOUT"
120	001526	021546	DH5		: " (PC) DEVADR LINE (SCR) CURLPR EXFLAG"
121	001530	021626	DT6		: SERRPC, SREG1, LINE, STMP0, CURLPR, EXFLAG"
122	001532	020330	DF2		: PRINT ALL OCTAL
123					
124					
125					
126	001534	021644	EM17		: "ALTERNATING I/O PATTERN TEST DONE"
127	001536	021706	DH6		: " (PC) DEVADR LINE CURLPR ICOUNT"
128	001540	021756	DT7		: SERRPC, DHADR, LINE, CURLPR, SREG0
129	001542	020330	DF2		: PRINT ALL OCTAL
130					
131					
132					
133	001544	021772	EM20		: "BINARY UP COUNT PATTERN TEST DONE"
134	001546	021706	DH6		: " (PC) DEVADR LINE CURLPR ICOUNT"
135	001550	021756	DT7		: SERRPC, DHADR, LINE, CURLPR, SREG0
136	001552	020330	DF2		: PRINT ALL OCTAL
137					
138					
139					
140	001554	022034	EM21		: "BINARY DOWN COUNT PATTERN TEST DONE"
141	001556	021706	DH6		: " (PC) DEVADR LINE CURLPR ICOUNT"
142	001560	021756	DT7		: SERRPC, DHADR, LINE, CURLPR, SREG0
143	001562	020330	DF2		: PRINT ALL OCTAL
144					
145					
146					
147	001564	022100	EM22		: "RANDOM DATA PATTERN TEST DONE"
148	001566	021706	DH6		: " (PC) DEVADR LINE CURLPR ICOUNT"
149	001570	021756	DT7		: SERRPC, DHADR, LINE, CURLPR, SREG0

```

150 001572 020330          DF2          ;PRINT ALL OCTAL
151
152          ;ERROR TABLE ITEM FOR ERROR 23
153
154 001574 022136          EM23          ;SINGLE CHAR PATTERN TEST DONE"
155 001576 021706          DH6           ;" (PC)  DEVADR  LINE  CURLPR  ICOUNT"
156 001600 021756          DT7           ;SERRPC DHADR,LINE,CURLPR,SREGO
157 001602 020330          DF2          ;PRINT ALL OCTAL
158
159          ;ERROR TABLE ITEM FOR ERROR 24
160
161 001604 022174          EM24          ;"TYPED BUFFER PATTERN TEST DONE"
162 001606 021706          DH6           ;" (PC)  DEVADR  LINE  CURLPR  ICOUNT"
163 001610 021756          DT7           ;SERRPC DHADR,LINE,CURLPR,SREGO
164 001612 020330          DF2          ;PRINT ALL OCTAL
165
166          ;ERROR TABLE ITEM FOR ERROR 25
167
168 001614 022233          EM25          ;"DATA PATTERNS TEST TIMEOUT"
169 001616 022266          DH7           ;" (PC) DEVADR  LINE  CURLPR  ICOUNT  PATCDE"
170 001620 022346          DT10          ;SERRPC,DHADR,LINE,CURLPR,SREGO,SREGI
171 001622 020330          DF2          ;PRINT ALL OCTAL
172
173
174
175
176 001624 005000          BEGIN: CLR      R0          ;INIT R0 TO INDICATE DEFAULT PARAMETERS
177 001626 005067 015776    CLR      VCFLG          ;INIT VECTOR ADDR SET UP FLAG
178 001632 005067 016234    CLR      DPFLG          ;CLEAR DATA PATTERNS TEST FLAG
179 001636 005067 016242    CLR      RETFLG         ;CLEAR ECHO TEST RETURN FLAG
180 001642 005067 016214    BEGINA: CLR      TITFLG          ;INIT TITLE MESSAGE FLAG
182 001646 012706 001100    MOV      @SCMTAG,R6      ;FIRST LOCATION TO BE CLEARED
(1) 001652 005026          CLR      (R6)+          ;CLEAR MEMORY LOCATION
(1) 001654 022706 001126    CMP      @SBODAT,R6      ;DONE?
(1) 001660 001374          BNE      -6             ;LOOP BACK IF NO
(1) 001662 012706 001100    MOV      @STACK,SP      ;SETUP THE STACK POINTER
(1) 001666 012737 011006 000020  MOV      @SCOPE,@IOTVEC  ;IOT VECTOR FOR SCOPE ROUTINE
(1) 001674 012737 000340 000022  MOV      @340,@IOTVEC+2  ;LEVEL 7
(1) 001702 012737 011250 000030  MOV      @SERROR,@ENTVEC ;ENT VECTOR FOR ERROR ROUTINE
(1) 001710 012737 000340 000032  MOV      @340,@ENTVEC+2  ;LEVEL 7
(1) 001716 012737 013460 000034  MOV      @STRAP,@TRAPVEC ;TRAP VECTOR FOR TRAP CALLS
(1) 001724 012737 000340 000036  MOV      @340,@TRAPVEC+2;LEVEL 7
(1) 001732 012737 013524 000024  MOV      @SPWRON,@PWRVEC ;POWER FAILURE VECTOR
(1) 001740 012737 000340 000026  MOV      @340,@PWRVEC+2 ;LEVEL 7
(1) 001746 005067 177246    CLR      $TIMES          ;INITIALIZE NUMBER OF ITERATIONS
(1) 001752 005067 177244    CLR      $ESCAPE          ;CLEAR THE ESCAPE ON ERROR ADDRESS
(1) 001756 112767 000001 177131  MOVB     @1,$ERRMAX      ;ALLOW ONE ERROR PER TEST
(1) 001764 012767 001764 177114  MOV      @,$SLPADR        ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
(1) 001772 012767 001772 177110  MOV      @,$SLPEAR        ;SETUP THE ERROR LOOP ADDRESS
(2) 002000 013746 000004          MOV      @@4,-(SP)      ;SAVE ERROR VECTOR
(2) 002004 013746 000006          MOV      @@6,-(SP)
(2) 002010 012767 002024 175766  MOV      @64$,4          ;SET UP TIME OUT VECTOR
(2) 002016 005777 177114    TST      @SWR            ;TRY TO REFERENCE HARDWARE SWR
(2) 002022 000407          BR       @55            ;BRANCH IF NO TIMEOUT TRAP OCCURS

```

(2)	002024	012767	000176	177104	64\$:	MOV	#SWREG,SWR	::POINT TO SOFTWARE SWR
(2)	002032	012767	000174	177100		MOV	#DISPREG,DISPLAY	::POINT TO SOFTWARE DISPLAY REG
(2)	002040	022626				CMP	(SP)+,(SP)+	::RESTORE STACK
(2)	002042	012637	000006		65\$:	MOV	(SP)+,0#6	::RESTORE ERROR VECTOR
(2)	002046	012637	000004			MOV	(SP)+,0#4	
(1)	002052	005067	177160			CLR	\$PASS	::CLEAR PASS COUNT
(1)	002056	132767	000200	177165		BITB	#APTSIZE,SENVN	::TEST USER SIZE UNDER APT
(1)	002064	001403				BEG	3\$::YES,USE NON-APT SWITCH
(1)	002066	012767	001252	177042		MOV	#SSWREG,SWR	::NO,USE APT SWITCH REGISTER
(1)	002074				3\$:			

185	002074	012767	014750	175702	START1:	MOV	#USER,ERRVEC	;SET UP THE BUS ERROR VECTOR
186	002102	012767	000340	175676		MOV	#340,ERRVEC+2	
187	002110	012767	015012	175672		MOV	#RESERR,RESVEC	;SET UP THE RSVD INSTR VECTOR
188	002116	012767	000340	175666		MOV	#340,RESVEC+2	
189	002124	005767	015732			TST	TITFLG	;HAVE WE TYPED TITLE ONCE ?
190	002130	001004				BNE	1S	;BR IF YES
191	002136	104400				TYPE		;GO TYPE PROGRAM TITLE
192	002134	022364				TITLE		
193	002136	005167	015720			COM	TITFLG	;SET FLAG - TYPE TITLE ONLY ONCE PER LOAD
194	002142	005767	015462		1S:	TST	VCFLG	;START AT 200 ??
195	002146	001404				BEQ	13S	;BR IF NOT
196	002150	004767	012072			JSR	PC,INPARA	;GO ASK FOR PARAMETERS
197	002154	005067	015450			CLR	VCFLG	;RE INIT START FLAG
198	002160	005767	015720		13S:	TST	RETFLG	;RETURN TO ECHO TESTS ?
199	002164	001402				BEQ	11S	;BR IF NOT
200	002166	000167	002434			JMP	ECHO1	;RETURN TO ECHO TEST START-UP
201	002172	005767	015674		11S:	TST	DPFLG	;RETURN TO DATA PATTERNS TEST ?
202	002176	001402				BEQ	12S	;BR IF NOT
203	002200	000167	003622			JMP	EXPAT1	;GO BACK TO DATA PATTERNS TESTS
204	002204	005700			12S:	TST	RD	;USE DEFAULT PARAMETERS ?
205	002206	001407				BEQ	START2	;BR IF YES
206	002210	022700	177777			CMP	#-1,RD	;CHANGE DH SELECT PARAM ONLY ?
207	002214	001002				BNE	2S	;BR IF NOT
208	002216	000167	012176			JMP	INPAR3	;GO ASK FOR SELECT PARAM.
209	002222	000167	012132		2S:	JMP	INPAR	;GO ASK FOR ALL PARAMETERS
210								
211	002226	012767	017526	015620	START2:	MOV	#DHADTB-2,ADPTR	;GET POINTER TO ADDRESS TABLE
212	002234	012767	017566	015614		MOV	#DHVCTB-2,VCPTR	;GET POINTER TO VECTOR TABLE
213	002242	012767	017630	015610		MOV	#BRVL-2,BRPTR	;GET POINTER TO BR LEVEL TABLE
214	002250	012767	177777	015416		MOV	#-1,DHNUM	;START WITH DH #00
215	002256	012767	000001	015156		MOV	#1,SELMSK	;SET UP DH11 BIT TEST MARKER
216								
217	002264	005267	015404		RESTR1:	INC	DHNUM	;GENERATE DH11 DEV NUMBER
218	002270	062767	000002	015556		ADD	#2,ADPTR	;UPDATE TABLE POINTERS
219	002276	062767	000002	015552		ADD	#2,VCPTR	
220	002304	062767	000002	015546		ADD	#2,BRPTR	
221	002312	036767	015124	015124		BIT	SELMSK,DHSEL	;TEST FOR SELECTED DH11
222	002320	001004				BNE	RSTRTA	;BR IF SELECTED FOR TEST
223	002322	006367	015114		REST1:	ASL	SELMSK	;SHIFT MARKER TO TEST NEXT DH11
224	002326	001737				BEQ	START2	;BR IF 16 TESTED - START OVER
225	002330	000755				BR	RESTR1	;GO TEST IF THIS ONE SELECTED
226	002332	017767	015516	015076	RSTRTA:	MOV	#ADPTR,DHADR	;SET UP DH11 ADDRESS
227	002340	017767	015512	015072		MOV	#VCPTR,DHVCT	;SET UP THE DH11 VECTOR ENTRY
228	002346	017767	015506	015316		MOV	#BRPTR,DHRLVL	;GET BR LEVEL VALUES
229	002354	004567	011560			JSR	RS,SUNUM	;GO SET DH NUMBER IN THE MESSAGE BUFFER
230	002360	017674				DHNUM		
231	002362	022466				TITLE2+20		
232	002364	104400				TYPE		;GO PRINT "TESTING DH11 #XX"
233	002366	022446				TITLE2		

Line	Address	Op	Opnd	Opnd	Opnd	Comment
236	002370	MOV	#,SLPADR			:INIT SCOPE RETURN
237						:*****
(3)						:*TEST 1 SUB-PROGRAM 1 - DATA RELIABILITY TESTS
(3)						:*****
(2)	002376	TST1:	SCOPE			
(1)	002400	MOV	#1,STINES			::DO 1 ITERATION
(1)	002406	MOV	#STN-1,STESTN			::SET TEST NUMBER IN MAIL BOX
238	002414	STDH1:	JSR	PC,CLSTAT		::GO CLEAR THE STATISTICS TABLES
239	002420		JSR	PC,KYBD1		::GO SET UP FOR KEYBOARD INTR.
240	002424		CLR	QUICK		::INIT THE QUICK TEST FLAG
241	002430		CLR	DRPLIN		::INIT DROPPED LINE FLAGS
242	002434		CLR	LINE		::INIT LINE NO. TO 00
243	002440		MOV	DHVT,R2		::SET UP THE VECTORS
244	002444		MOV	#RINT1,(R2)+		::GO TO RINT1 ON RCVR INTR
245	002450		MOVB	DHRLVL,(R2)+		:
246	002454		TSTB	(R2)+		:
247	002456		MOV	#TINT1,(R2)+		::GO TO TINT1 ON XMITTR INTR
248	002462		MOVB	DHTLVL,(R2)		:
249	002466		MOV	DHADR,R1		::SET UP DEVICE ADDRESS
250	002472	NEWLIN:	JSR	PC,SELIN		::GO SELECT NEW LINE FOR TEST
251	002476		BR	1\$::BR IF TESTED ALL SELECTED LINES
252	002500		BR	2\$::BR IF NOT DONE
253	002502	1\$:	JMP	PRSTAT		::GO CHECK AND REPORT STATISTICS
254	002506	2\$:	JSR	RS,SUNUM		::PUT LINE NO. IN MSG
255	002512		LINE			
256	002514		STMSG2+20			
257	002516		TYPE			::TYPE LINE NO. BEING TESTED
258	002520		STMSG2			
259	002522		MOV	#LPRTAB,LPRPTR		::SET UP LPR TABLE POINTER
260	002530	NEWLPR:	JSR	RS,SETLPR		::GO SET UP LPR CONSTANT
261	002534		BR	NEWLIN		::BR IF DONE ALL BAUD RATES AT THIS LINE
262	002536		MOV	#177760,CHRCNT		::INIT CHAR COUNT
263	002544		MOV	#-1,CLSEL		::INIT CHR LNGTH SELECT CODE
264						
265						
266						
267	002552		CLR	QUICKX		::INIT QUICK TEST EXIT FLAG
268	002556	NEWCL:	JSR	RS,SETCL		::GO SET UP THE CHAR LENGTH
269	002562		BR	NEWLPR		::BR IF DONE ALL FOUR LENGTHS
270	002564		CLR	PARBIT		::INIT PARITY SELECT BIT CODE
271						
272	002570	NEWPAR:	JSR	RS,SETPAR		::GO SET PARITY SELECT
273	002574		BR	NEWCL		::BR IF DONE ALL COMBOS
274						
275	002576	DHST1:	JSR	PC,DHSET1		::GO SET UP FOR TESTING THIS LINE
276	002602		BIS	LINMSK,BAR(R1)		::ACTIVATE THE SELECTED LINE
277	002610		JSR	PC,CHPS1		::GO CLEAR PSW
278						
279						
280	002614		MOV	#100,TIMEA		::INIT TIMER A
281	002622		CLR	TIMEB		::INIT TIMER B
282	002626	1\$:	CHP	#3,ROONE		::BOTH RCVR AND XMITTR DONE ?
283	002634		BEQ	3\$::BR IF YES
284	002636		JSR	PC,TIMEIT		::CALL THE TIMER

```

285 002642 000771          BR      1S          ;TIMER STEPS AROUND THIS BRANCH IF
286                                     ;TIMEOUT OCCURS
287
288 002644 052777 004000 014564      BIS      #BIT11,2DHADR ;CLEAR OUT THE DH11
289 002652 116700 015020          MOVB     LINE,R0       ;GET LINE NO.
290 002655 006300          ASL      R0           ;FORM TABLE INDEX
291 002660 016067 025416 176312      MOV      RTOTAL(R0),STMP0 ;SAVE XMITTED COUNT
292 002666 016067 025456 176306      MOV      XTOTAL(R0),STMP1 ;SAVE THE RCVD COUNT
293 002674 004567 011240          JSR      RS,SUNUM     ;PUT LINE NO. IN MESSAGE
294 002700 017676          LINE
295 002702 021221          EM12+35
296 002704 012767 002714 176176      MOV      #25,SLPERR   ;SET UP ERROR LOOP RETURN
297 002712 104012          ERROR  12           ;LINE FAILED TO FINISH ON TIME - HUNG
298 002714 056767 014530 014530 2S:   BIS      LINMSK,DRPLIN ;SET DROP FLAG
299 002722 012706 001100          MOV      #STACK,SP   ;RESET STACK POINTER
300 002726 000661          BR      NEWLIN       ;GO TRY ANOTHER LINE
301
302
303 002730 012711 004000          3S:   MOV      #BIT11 (R1)  ;CLEAR THE WORLD OUT IN THE DH11
304 002734 004767 001050          JSR      PC,CKER     ;GO UPDATE THE DATA ERROR TABLES
305 002740 000713          BR      NEWPAR       ;GO TRY NEXT PARITY COMBINATION

```

```

308
309
310 ;TRANSMITTER INTERRUPT SERVICE ROUTINE ONE
311 002742 032711 002000 TINT1: BIT #BIT10,(R1) ;NON EX MEM ERROR ??
312 002746 001432 BEQ 25 ;BR IF NOT
313
314 002750 011103 MOV (R1),R3 ;SAVE THE SCR
315 002752 004767 014400 JSR PC,CHPS2 ;GO LOCK OUT INTRs
316 002756 012711 004000 MOV #BIT11,(R1) ;CLEAR OUT THE DH11
317 002762 116704 014710 MOV#B LINE,R4 ;SET UP THE S/B DATA
318 002766 042703 175760 BIC #175760,R3 ;CLEAR OUT SUPERFLUOUS BITS
319 002772 010102 MOV R1,R2 ;SET UP REGADR
320 002774 004767 011224 JSR PC,SUER1 ;GO SET UP ERROR INFO
321 003000 004567 011134 JSR RS,SUNUM ;GO SET UP LINE NO. IN MSG
322
323 003004 017676 LINE
324 003006 020232 EM1+44
325 003010 012767 003020 176072 MOV #15,SLPERR ;SET UP THE ERROR LOOP RETURN
326 003016 104001 ERROR 1 ;NON EX MEM ERROR
327 003020 022626 15: CMP (SP)+,(SP)+ ;POP THE STACK
328 003022 056767 014422 014422 BIS LINMSK,DRPLIN ;SET THE DROPPED FLAG FOR THIS LINE
329 003030 000167 177436 JMP NEWLIN ;GO TRY NEXT LINE
330
331 003034 011103 25: MOV (R1),R3 ;GET THE SCR REG CONTENTS
332 003036 100433 BMI 45 ;BR IF XMIT DONE SET
333
334 003040 004767 014312 JSR PC,CHPS2 ;GO LOCK OUT INTRs
335 003044 012711 004000 MOV #BIT11,(R1) ;CLEAR THE DH11 - FATAL ERROR
336 003050 012704 100000 MOV #BIT15,R4 ;SET UP S/B DATA
337 003054 156704 014616 BIS#B LINE,R4
338 003060 042703 077760 BIC #77760,R3 ;CLEAR OUT SUPERFLUOUS BITS
339 003064 010102 MOV R1,R2 ;SET UP REGADR
340 003066 004767 011132 JSR PC,SUER1 ;GO SET UP ERROR INFO
341 003072 004567 011042 JSR RS,SUNUM ;GO SET UP LINE NO. IN MSG
342
343 003076 017676 LINE
344 003100 020414 EM2+54
345 003102 012767 003112 176000 MOV #35,SLPERR ;SET UP ERROR LOOP RETURN
346 003110 104002 ERROR 2 ;XMITR FALSE INTERRUPT
347 003112 022626 35: CMP (SP)+,(SP)+ ;POP THE STACK
348 003114 056767 014330 014330 BIS LINMSK,DRPLIN ;SET THE DROPPED FLAG FOR THIS LINE
349 003122 000167 177344 JMP NEWLIN ;GO TRY NEXT LINE
350
351 003126 005761 000012 45: TST BAR(R1) ;DID BAR BIT CLEAR ??
352 003132 001432 BEQ 65 ;BR IF YES
353
354 003134 004767 014216 JSR PC,CHPS2 ;GO LOCK OUT INTRs
355 003140 016103 000012 MOV BAR(R1),R3 ;GET THE WAS DATA
356 003144 012711 004000 MOV #BIT11,(R1) ;CLEAR THE DH11
357 003150 005004 CLR R4 ;SET UP S/B DATA
358 003152 010102 MOV R1,R2 ;SET UP REGADR
359 003154 062702 000012 ADD #BAR,R2
360 003160 004767 011040 JSR PC,SUER1 ;GO SET UP ERROR INFO
361 003164 004567 010750 JSR RS,SUNUM ;GO SET UP LINE NO. IN MSG
362
363 003170 017676 LINE
364 003172 020474 EM3+55

```

362	003174	012767	003204	175706		MOV	#55,SLPERR	:SAVE THE ERROR LOOP RETURN
363	003202	104003				ERROR	3	:BUFFER ACTIVE REG FAILED TO CLEAR
364	003204	022626			55:	CMP	(SP)+,(SP)+	:POP GOES THE STACK
365	003206	056767	014236	014236		BIS	LINMSK,DRPLIN	:SET THE DROPPED FLAG FOR THIS LINE
366	003214	000167	177252			JMP	NEWLIN	:GO TRY NEXT LINE
367								
368	003220	005761	000010		65:	TST	BCR(R1)	:DID BYTE COUNT GO TO ZERO ??
369	003224	001432				BEQ	BS	:BR IF YES
370								
371	003226	004767	014124			JSR	PC,CHPS2	:GO LOCK OUT INTRs
372	003232	016103	000010			MOV	BCR(R1),R3	:GET THE WAS DATA
373	003236	012711	004000			MOV	#BIT11,(R1)	:CLEAR THE DH11
374	003242	005004				CLR	R4	:SET UP S/B DATA
375	003244	010102				MOV	R1,R2	:SET UP REGADR
376	003246	062702	000010			ADD	#BCR,R2	
377	003252	004767	010746			JSR	PC,SUER1	:GO SET UP THE ERROR INFO
378	003256	004567	010656			JSR	RS,SUNUM	:GO SET UP LINE NO. IN MSG
379	003262	017676				LINE		
380	003264	020551				EM4+52		
381	003266	012767	003276	175614		MOV	#75,SLPERR	:SET UP ERROR LOOP RETURN
382	003274	104004				ERROR	4	:BYTE COUNT REG FAILED TO GO TO 000000
383	003276	022626			75:	CMP	(SP)+,(SP)+	:POP GOES THE STACK
384	003300	056767	014144	014144		BIS	LINMSK,DRPLIN	:SET THE DROPPED FLAG FOR THIS LINE
385	003306	000167	177160			JMP	NEWLIN	:GO TRY NEXT LINE
386								
387	003312	016103	000006		85:	MOV	CAR(R1),R3	:GET THE WAS DATA
388	003316	016704	014174			MOV	CHRCNT,R4	:SET UP S/B DATA
389	003322	005404				NEG	R4	
390	003324	062704	030212			ADD	#TBUF,R4	
391	003330	020304				CMP	R3,R4	:WAS CAR CORRECT ??
392	003332	001425				BEQ	10\$:BR IF YES
393								
394	003334	004767	014016			JSR	PC,CHPS2	:GO LOCK OUT INTRs
395	003340	010102				MOV	R1,R2	:SET UP REGADR
396	003342	062702	000006			ADD	#CAR,R2	
397	003346	004767	010652			JSR	PC,SUER1	:GO SET UP ERROR INFO
398	003352	004567	010562			JSR	RS,SUNUM	:GO SET UP LINE NO. IN MSG
399	003356	017676				LINE		
400	003360	020633				EM5+57		
401	003362	012767	003372	175520		MOV	#95,SLPERR	:SET UP THE ERROR RETURN
402	003370	104005				ERROR	5	:CURRENT ADDRESS REG NOT CORRECT
403	003372	022626			95:	CMP	(SP)+,(SP)+	:POP THE STACK
404	003374	056767	014050	014050		BIS	LINMSK,DRPLIN	:SET THE DROPPED FLAG FOR THIS LINE
405	003402	000167	177064			JMP	NEWLIN	:GO TRY NEXT LINE
406								
407	003406				105:			
(2)	003406	010346				MOV	R3,-(SP)	::PUSH R3 ON STACK
(2)	003410	010446				MOV	R4,-(SP)	::PUSH R4 ON STACK
408	003412	016703	014100			MOV	CHRCNT,R3	
409	003416	005403				NEG	R3	:CHAR COUNT IN R3
410	003420	116704	014252			MOVB	LINE,R4	:GET LINE NO.
411	003424	006304				ASL	R4	:DOUBLE IT
412	003426	060364	025456			ADD	R3,XTOTAL(R4)	:UPDATE TOTAL XMIT COUNT
413	003432	012604				MOV	(SP)+,R4	::POP STACK INTO R4

MAINDEC-11-DZDHN-A
DZDHN.A.P11 T1

MACY11 27(663) 15-DEC-75 09:29 PAGE 4-2
SUB-PROGRAM 1 - DATA RELIABILITY TESTS

SEQ 0103

(2) 003434 012603
414 003436 052767
415 003444 000002

000001 014060

MOV (SP)+,R3
BIS #BIT0,RDONE
RTI

::POP STACK INTO R3
::SET XMIT DONE FLAG
::RETURN TO WAIT LOOP

;RECEIVER INTERRUPT SERVICE ROUTINE ONE

```

418
419
420
421 003446 032711 040000 RINT1: BIT #BIT14,(R1) ;SILO OVERFLOW ERROR ??
422 003452 001431 BEQ 25 ;BR IF NOT
423
424 003454 004767 013676 JSR PC,CHPS2 ;GO LOCK OUT INTRs
425 003460 011103 MOV (R1),R3 ;GET THE MSG DATA
426 003466 012711 004000 MOV #BIT11,(R1) ;NOW CLEAR THE DH11
427 003472 042703 177760 BIC #177760,R3 ;CLEAR JUNK
428 003478 116704 014200 MOV#B LINE,R4
429 003484 004767 010522 JSR PC,SUER1 ;GO SET UP ERROR INFO
430 003490 004567 010432 JSR RS,SUNUM ;GO SET UP LINE NO. IN MSG
431 003496 017676 LINE
432 003502 020702 EM6+44
433 003508 012767 003522 175370 MOV #15,SLPERR ;SET UP ERROR LOOP RETURN
434 003514 104006 ERROR 6 ;SILO OVERFLOW - BAD,BAD,BAD !!!
435 003520 022626 15: CMP (SP)+,(SP)+ ;POP GOES THE STACK
436 003526 056767 013720 013720 BIS LINMSK,DRPLIN ;SET THE DROPPED FLAG FOR THIS LINE
437 003532 000167 176734 JMP NEMLIN ;GO TRY NEXT LINE
438
439 003536 105711 25: TSTB (R1) ;CHAR AVAIL SET ??
440 003540 100434 BMI 45 ;BR IF YES
441
442 003542 004767 013610 JSR PC,CHPS2 ;GO LOCK OUT INTRs
443 003548 011103 MOV (R1),R3 ;GET MSG DATA
444 003554 042703 177560 BIC #177560,R3 ;CLEAN IT UP
445 003560 012711 004000 MOV #BIT11,(R1) ;NOW CLEAR DH11
446 003566 012704 000200 MOV #BIT07,R4 ;SET UP S/B DATA
447 003572 156704 014106 BISB LINE,R4
448 003578 010102 MOV R1,R2 ;SET UP REGADR
449 003584 004767 010426 JSR PC,SUER1 ;GO SET UP ERROR INFO
450 003590 004567 010336 JSR RS,SUNUM ;GO SET UP LINE NO. IN MSG
451 003596 017676 LINE
452 003602 020756 EM7+51
453 003608 012767 003616 175274 MOV #35,SLPERR ;SET UP THE ERROR LOOP RETURN
454 003614 104007 ERROR 7 ;RECEIVER FALSE INTERRUPT
455 003620 022626 35: CMP (SP)+,(SP)+ ;POP GOES THE SP
456 003626 056767 013624 013624 BIS LINMSK,DRPLIN ;SET THE DROPPED FLAG FOR THIS LINE
457 003632 000167 176640 JMP NEMLINE ;GO TRY NEXT LINE
458
459 003632 016167 000002 175340 45: MOV NRC(R1),STMP0 ;SAVE THE DATA RECEIVED
460 003640 100431 BMI 65 ;BR IF IT WAS VALID DATA
461
462 003642 004767 013510 JSR PC,CHPS2 ;GO LOCK OUT INTRs
463 003648 012711 004000 MOV #BIT11,(R1) ;NOW CLEAR THE DH11
464 003654 162767 025732 022046 SUB #RBUF,RFBPTR ;WHICH CHAR WAS IT ??
465 003660 016702 022042 MOV RBPTR,R2 ;SAVE CHAR NUMBER
466 003666 004767 010330 JSR PC,SUER2 ;GO SET UP ERROR INFO
467 003672 004567 010244 JSR RS,SUNUM ;GO SET UP LINE NO. IN MSG
468 003678 017676 LINE
469 003684 021026 EM10+45
470 003700 012767 003710 175202 MOV #55,SLPERR ;SET UP ERROR RETURN
471 003706 104010 ERROR 10 ;RECEIVED INVALID DATA
472 003710 022626 55: CMP (SP)+,(SP)+ ;POP GOES THE STACK

```

```

472 003712 056767 013532 013532      BIS      LINMSK,DRPLIN      ;SET THE DROPPED FLAG FOR THIS LINE
473 003720 000167 176546              JMP      NEWLIN          ;GO TRY ANOTHER LINE
474
475 003724              65:
(2) 003724 010346      MOV      R3,-(SP)        ;;PUSH R3 ON STACK
(2) 003726 010446      MOV      R4,-(SP)        ;;PUSH R4 ON STACK
476 003730 016777 175244 021770      MOV      STMP0,RBFPTR    ;STORE CHAR IN THE BUFFER
477 003736 052767 000002 021762      ADD     #2,RBFPTR        ;UPDATE THE POINTER
478 003744 026767 013556 021754      CMP     REND,RBFPTR      ;END OF BUFFER ??
479 003752 001013      BNE     75              ;BR IF NOT
480 003754 016703 013536      MOV     CHRCNT,R3        ;GET CHAR COUNT
481 003760 005403      NEG     R3
482 003762 116704 013710      MOV8    LINE,R4          ;GET THE LINE NO.
483 003766 006304      ASL     R4              ;DOUBLE IT
484 003770 050364 025416      ADD     R3,RTOTAL(R4)    ;UPDATE TOTAL RECEIVED COUNT
485 003774 052767 000002 013522      BIS     #BIT1,RDONE      ;SET THE RCVR DONE FLAG
486 004002              75:
(2) 004002 012604      MOV     (SP)+,R4        ;;POP STACK INTO R4
(2) 004004 012603      MOV     (SP)+,R3        ;;POP STACK INTO R3
487 004006 000002      RTI

```

490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543

004010 012767 025732 021710
004016 012767 030212 021704
004024 012767 025516 021664
004032 012767 025556 021660
004040 012767 025616 021654
004046 012767 025656 021650
004054 116705 013616
004060 006305
004062 060567 021630
004066 060567 021626
004072 060567 021624
004076 060567 021622

004102 117704 021622
004106 000304
004110 105004
004112 156704 013560
004116 000304
004120 052704 100000
004124 017703 021576
004130 020304
004132 001435

004134 010367 175056
004140 010146
004142 016701 021560
004146 016702 021556
004152 010200
004154 162700 030212
004160 004767 000066
004164 004767 010030
004170 004567 007744
004174 017676
004176 021161
004200 012767 004210 174702
004206 104011

004210 012601
004212 032767 070000 174776
004220 001402

004222 004767 000036

004226 005267 021476
004232 062767 000002 021466
004240 026767 013262 021460
004246 001315

004250 000207
004252 120304

; THIS ROUTINE IS CALLED TO CHECK THE RECEIVED DATA, REPORT ALL ERRORS,
; AND UPDATE THE STATISTICS TABLE ENTRIES FOR ALL LINES ACTIVE

CKER: MOV #RBUF, RBFPTR ;SET UP POINTERS
MOV #TBUF, TBFPTR
MOV #DATAERR, DEPTR ;SET UP POINTERS TO STATISTICS TABLES
MOV #PARERR, PEPTR
MOV #OVREARR, ORPTR
MOV #FRMERR, FRPTR
MOVSB LINE, RS ;GET LINE NO. AND DOUBLE IT
RSL RS
ADD RS, DEPTR ;POINT TO CORRECT LINE ENTRY IN TABLE
ADD RS, PEPTR
ADD RS, ORPTR
ADD RS, FRPTR

1S: MOVSB #TBFPTR, R4 ;GET THE S/B DATA
SWAB R4 ;PUT LINE NO. IN HIGH BYTE
CLRB R4
BISB LINE, R4
SWAB R4
BIS #BIT15, R4 ;AND FINALLY THE VALID DATA BIT
MOV #RBFPTR, R3 ;GET THE WAS DATA
CMP R3, R4 ;WAS = S/B ?????
BEQ 3S ;BR IF YES

MOV R3, STMP7 ;SAVE THE WAS DATA
MOV R1, -(SP) ;SAVE THE DEVADR
MOV RBFPTR, R1 ;GET THE SBADR
MOV TBFPTR, R2 ;GET THE WASADR
MOV R2, R0 ;GET XMIT BUFFER ADDR
SUB #TBUF, R0 ;GENERATE CHAR #
JSR PC, UPDER ;GO CHECK AND UPDATE THE DATA ERROR TABLE
JSR PC, SUER2 ;GO SET UP ERROR INFO
JSR RS, SUNUM ;GO PUT LINE NO. IN MSG
LINE
EM11+23
MOV #2S, SLPERR ;SET UP ERROR RETURN
ERROR 11 ;DATA COMPARE ERROR OR PARITY, FRAMING
;OR OVERRUN
2S: MOV (SP)+, R1 ;RESTORE THE DEVADR
BIT #70000, STMP7 ;ANY PARITY, OVERRUN, OR FRAMING ERROR
BEQ 3S ;BR IF NOT

JSR PC, SOFT ;GO TAKE CARE OF SOFT ERROR REPORT

3S: INC TBFPTR ;UPDATE POINTERS
ADD #2, RBFPTR
CMP RBFEND, RBFPTR ;COMPARED ALL CHARS ??
BNE 1S ;BR IF NOT

RTS PC ;RETURN TO WAIT LOOP

UPDER: CMPB R3, R4 ;DATA BYTES CORRECT ??

544	004254	001402		BEO	1S	:BR IF YES
545	004256	005277	021434	INC	2DEPTR	:COUNT THE DATA ERROR
546	004262	000207		1S: RTS	PC	:RETURN
547						
548	004264	006367	174726	SOFT: ASL	STMP7	:TEST FOR OVERRUN ERRORS
549	004270	100002		BPL	1S	:BR IF NONE
550	004272	005277	021424	INC	2ORPTR	:COUNT IT
551	004276	006367	174714	1S: ASL	STMP7	:TEST FOR FRAMING ERRORS
552	004302	100002		BPL	2S	:BR IF NONE
553	004304	005277	021414	INC	2FRPTR	:COUNT IT
554	004310	006367	174702	2S: ASL	STMP7	:TEST FOR PARITY ERRORS
555	004314	100002		BPL	3S	:BR IF NONE
556	004316	005277	021376	INC	2PEPTR	:COUNT IT
557	004322	000207		3S: RTS	PC	:RETURN
558						
559						

```

;THIS ROUTINE IS CALLED TO PRINT OUT THE TEST STATISTICS
561
562
563 004324 012767 000001 013116 PRSTAT: MOV      #1,LINMSK ;SET UP BIT TEST MARKER
564 004332 005001                CLR      R1      ;R1 CONTAINS THE LINE NO.
565 004334 004567 007600                JSR      RS,SUNUM ;GO SET UP DH11 # IN STAT MESSAGE
566
567 004340 017674                DHNUM
568 004342 023262                STMSG1+6
569 004344 104400                TYPE      ;GO TYPE THE STATISTICS HEADER
570 004346 023254                STMSG1
571 004350 104400                TYPE      ;TYPE HEADER
572 004352 023357                STMSG4
573
574 004354 036767 013070 013070 1$: BIT      LINMSK,DRPLIN ;DID THIS LINE GET DROPPED ?
575 004362 001411                BEQ      2$      ;BR IF NOT
576
577 004364 010167 174626                MOV      R1,STMP7 ;SAVE THE LINE NO.
578 004370 004567 007544                JSR      RS,SUNUM ;GO PUT LINE NO. IN MESSAGE
579 004374 001216                STMP7
580 004376 023336                STMSG3+10
581 004400 104400                TYPE
582 004402 023326                STMSG3
583 004404 000436                BR       3$      ;GO TEST NEXT LINE
584
585 004406 010102                2$: MOV      R1,R2 ;SET UP R2 WITH TABLE INDEX
586 004410 006302                ASL      R2
587 004412 036767 013032 013026 BIT      LINMSK,LINSEL ;WAS THIS LINE SELECTED ??
588 004420 001430                BEQ      3$      ;BR IF NOT
589 004422 010167 174552                MOV      R1,STMP0 ;SET UP THE ERROR INFORMATION FROM
590                                ;THE TABLES INTO THE MESSAGE POINTERS
591 004426 016267 025416 174546                MOV      RTOTAL(R2),STMP1
592 004434 016267 025456 174542                MOV      XTOTAL(R2),STMP2
593 004442 016267 025516 174536                MOV      DATERR(R2),STMP3
594 004450 016267 025556 174532                MOV      PARERR(R2),STMP4
595 004456 016267 025616 174526                MOV      OVRERR(R2),STMP5
596 004464 016267 025656 174522                MOV      FRMERR(R2),STMP6
597 004472 012767 004502 174410                MOV      #3$,SLPERR ;RETURN TO 3$ AFTER PRINTING LINE
598 004500 104013                ERROR 13
599 004502 005201                3$: INC      R1 ;STEP TO NEXT LINE
600 004504 006367 012740                ASL      R1 ;SHIFT THE MARKER
601 004510 001401                BEQ      ENDA ;BR IF ALL LINES REPORTED
602 004512 000720                BR       1$      ;GO BACK AND DO THIS LINE
603

```

607	004514	000004			ENDA:	SCOPE			
608	004516	105067	174360			CLAB	STSTN		:RE-INIT TEST NUMBER FOR NEXT PASS
609	004522	012767	000240	004134		MOV	#240,SEOP		:NOP THE SCOPE IN ENDPASS ROUTINE
610	004530	005267	013140			INC	DHNUM		:GENERATE NEW DH11 NUMBER
611	004534	062767	000002	013312		ADD	#2,ADPTR		:UPDATE THE TABLE POINTERS
612	004542	062767	000002	013306		ADD	#2,VCPTR		
613	004550	062767	000002	013302		ADD	#2,BRPTR		
614	004556	006367	012660			ASL	SELMSK		:SHIFT MARKER TO TEST NEXT DH11
615	004562	001002				BNE	IS		:BR IF NOT TESTED ALL DH11'S
616	004564	000167	004074			JMP	SEOP		:JUMP TO EOP IF WE HAVE
617	004570	036767	012646	012646	15:	BIT	SELMSK,DHSEL		:IS THIS DH11 SELECTED ?
618	004576	001746				BEQ	ENDA		:BR IF NOT
619	004600	000167	175526			JMP	RSTRTA		:GO TEST THIS DH11

```

621          .SBTTL SUB-PROGRAM 2 - SINGLE LINE ECHO/CABLE TESTS
622          :
623          :
624          :
625          :
626          :
627 004604 012767 177777 013272 ECHO:  MOV    #-1,RETFLG      ;SET RETURN FLAG - COME BACK
628 004612 005067 013254          CLR    DPFLG        ;CLEAR PATTERNS TEST FLAG
629 004616 005067 013006          CLR    VCFLG        ;INIT VECTOR SETUP FLAG
630 004622 000167 175014          JMP    BEGINA       ;TO "ECHO1" AFTER SETUP
631
632 004626 012767 160020 012602 ECHO1: MOV    #160020,DHADR ;SET UP DH11 DEFAULT ADDRESS
633 004634 012767 000330 012576  MOV    #330,DHVCT   ;SET UP DH11 DEFAULT VECTOR
634 004642 104400          TYPE   ECMSG1       ;PRINT I.D. MESSAGE
635 004644 023467          ECMSG1              ;"SINGLE LINE ECHO TEST - CONNECT
636          :
637          :
638 004646 000167 007506          JMP    INPAR        ;TERMINAL TO TEST LINE
639          :
640          :
641          :
642          :
643          :
644          :
645          :
646          :
647          :
648          :
649          :
650          :
651          :
652          :
653          :
654          :
655          :
656          :
657          :
658          :
659          :
660          :
661          :
662          :
663          :
664          :
665          :
666          :
667          :
668          :
669          :
670          :
671          :
672          :
673          :
674          :

```

675 005016 000167 000574 JMP SENDP1 ;GO TO SEND ROUTINE
676

679	005022	004767	012330		ECH03:	JSR	PC,CHPS2	:GO LOCK OUT INTRs
680	005026	005067	013036			CLR	CEXIT	:INIT CONTROL-C EXIT FLAG
681	005032	005067	013060			CLR	EXFLAG	:CLEAR TEST EXIT FLAGS
682	005036	012767	120240	012626		MOV	#120240,DHRLVL	:INIT FOR BR LEVEL 5
683	005044	016701	012366			MOV	DHADR,R1	:SET UP DEVICE ADDRESS
684	005050	012711	004000			MOV	#BIT11,(R1)	:CLEAR THE SELECTED DH11
685	005054	016700	012360			MOV	DHVC1,R0	:GET THE FIRST VECTOR ADDRESS
686	005060	012720	005374			MOV	#RINT2,(R0)+	:SET UP THE VECTORS
687	005064	116710	012602			MOVB	DHRLVL,(R0)	
688	005070	005720				TST	(R0)+	
689	005072	012720	005226			MOV	#TINT2,(R0)+	
690	005076	116710	012571			MOVB	DH1VL,(R0)	
691	005102	016711	012570			MOV	LINE,(R1)	:SET THE LINE SELECT BITS
692	005106	012702	030212			MOV	#BUF,R2	:INIT BUFFER POINTER
693	005112	052711	000100			BIS	#BIT06,(R1)	:ENABLE RCVR INTRs
694	005116	016761	012370	000004		MOV	CURLPR,LPR(R1)	:SET UP LINE PARAMETERS
695	005124	004567	007010			JSR	RS,SUNUM	:PUT LINE NO. IN MESSAGE
696	005130	017676				LINE		
697	005132	023644				ECMSG3+20		
698	005134	104400				TYPE		:GIVE DIRECTIONS
699	005136	023624				ECMSG3		: "GO TYPE CHARS ON TEST TERMINAL"
700	005140	104400				TYPE		:PRINT DIRECTIONS
701	005142	023703				ECMSG4		: "[CONTROL-C TO EXIT] [CONTROL-E TO ECHO BUFFER]"
702	005144	004767	012172			JSR	PC,CHPS1	:GO CLEAR PSM
703								
704	005150	012767	000100	012706	DHWAIT:	MOV	#100,TIMEA	:INIT TIMER "A"
705	005156	005067	012704			CLR	TIMEB	:INIT TIMER "B"
706	005162	005767	012702		1S:	TST	CEXIT	:CONTROL-C EXIT ??
707	005166	001015				BNE	2S	:BR IF YES
708	005170	004767	007706			JSR	PC,TIMEIT	:CALL TIMER
709	005174	000772				BR	1S	:BR IF NO TIMEOUT
710								
711	005176	010167	173760			MOV	R1,SREG1	:SAVE DEVADR
712	005202	011167	173772			MOV	(R1),STMPD	:SAVE CONTENT OF SCR
713	005206	052711	004000			BIS	#BIT11,(R1)	:CLEAR OUT THE DH11
714	005212	012767	005222	173670		MOV	#2S,SLPERR	:SET ERROR LOOP RETURN
715	005220	104016				ERROR	16	:REPORT RCVR WAIT TIMEOUT
716	005222	000167	177424		2S:	JMP	ECH02	:GO RESTART
717								
718								


```

798 005616 104400 SENDP1: TYPE ;ASK FOR DIRECTIONS
799 005620 024640 SNMSG2 ;"TYPE SEND BUFFER - TERMINATE WITH CONTROL-C"
800 005622 012705 030212 MOV #TBUF,R5 ;SET UP BUFFER POINTER
801 005626 104405 1S: RDCHR ;GET CHAR
802 005630 012600 MOV (SP)+,R0
803 005632 110067 015622 MOVB R0,EC2 ;ECHO CHAR
804 005636 104400 TYPE
805 005640 023460 EC2
806 005642 022700 000003 CMP #3,R0 ;WAS IT A CONTROL-C ??
807 005646 001421 BEQ 4S ;BR IF YES
808
809 005650 026727 011642 000400 CMP CHRCNT,#256. ;BUFFER FULL ??
810 005656 003015 BGT 4S ;BR IF YES
811 005660 022700 000012 CMP #12,R0 ;WAS IT A LINE FEED ?
812 005664 001010 BNE 3S ;BR IF NOT
813
814 005666 110025 MOVB R0,(R5)+ ;LOAD CHAR TYPED
815 005670 116704 012270 MOVB FILLB,R4 ;GET FILLER COUNT
816 005674 116725 012262 2S: MOVB FILLA,(R5)+ ;LOAD A FILLER
817 005700 005304 DEC R4 ;COUNT IT
818 005702 001374 BNE 2S ;BR IF NOT DONE
819 005704 000750 BR 1S ;GET SOME MORE INPUT
820
821 005706 110025 3S: MOVB R0,(R5)+ ;LOAD BUFFER
822 005710 000746 BR 1S ;GO GET SOME MORE
823
824 005712 004767 007646 4S: JSR PC,SENDP2 ;GO XMIT THE BUFFER
825 005716 105067 015536 5S: CLRB EC2 ;CLEAR ECHO BUFFER
826 005722 104400 TYPE
827 005724 024720 SNMSG3 ;"CHANGE PARAMETERS- Y OR N"
828 005726 104405 6S: RDCHR
829 005730 012600 MOV (SP)+,R0 ;GET CHAR
830 005732 122700 000015 CMPB #15,R0 ;WAS IT A <CR> HE TYPED ??
831 005736 001405 BEQ 7S ;BR IF IT WAS
832 005740 110067 015514 MOVB R0,EC2 ;ECHO IT
833 005744 104400 TYPE
834 005746 023460 EC2
835 005750 000766 BR 6S ;GO WAIT FOR TERMINATOR
836
837 005752 105767 015502 7S: TSTB EC2 ;<CR> ONLY ??
838 005756 001717 BEQ SENDP1 ;BR IF YES
839 005760 122767 000116 015472 CMPB #116,EC2 ;DID HE SAY NO ??
840 005766 001713 BEQ SENDP1 ;BR IF HE DID
841 005770 122767 000131 015462 CMPB #131,EC2 ;DID HE SAY YES ??
842 005776 001347 BNE 5S ;GO ASK ALL OVER AGAIN
843 006000 000167 176646 JMP ECHO2 ;GO ASK FOR NEW PARAMETERS

```

.SBTTL SUB-PROGRAM THREE - DATA PATTERNS TESTS

```

846
847
848
849
850
851
852 006004 012767 177777 012060 EXPAT: MOV #-1,DPFLG ;SET PATTERNS TEST FLAG
853 006012 005067 012066 CLR RETFLG ;CLR ECHO TESTS FLAG
854 006016 005067 011606 CLR VCFLG ;CLEAR VECTOR SETUP FLAG
855 006022 000167 173614 JMP BEGINA ;GO SET UP RETURN TO "EXPAT1"
856
857 006026 012767 160020 011402 EXPAT1: MOV #160020,DHADR ;SET UP DEFAULT DH11 ADDR
858 006034 012767 000330 011376 MOV #330,DHVCT ;AND VECTOR TOO
859 006042 104400 TYPE ;
860 006044 024757 DPMSG1 ;"DATA PATTERNS TESTS - CONNECT TEST JUMPAR"
861 006046 000167 006306 JMP INPAR ;GO GET SOME PARAMETERS RETURN TO EXPAT2
862
863 006052 004767 176574 EXPAT2: JSR PC,ECHO2 ;GO GET REST OF THE PARAMETERS
864 006056 004767 011206 JSR PC,SUCLMK ;GO SET UP CHAR LENGTH MASK
865 006062 104400 1S: TYPE ;
866 006064 025035 DPMSG2 ;"BUFFER SIZE ? (1-512)"
867 006066 104410 RDDEC ;GET THE SIZE TYPED
868 006070 012600 MOV (SP)+,RO ;
869 006072 001406 BEQ 2S ;BR IF DEFAULT TO 256. <CR>
870
871 006074 020027 001001 CMP RO,#513. ;TOO BIG ?
872 006100 002405 BLT 3S ;BR IF NOT
873 006102 104400 TYPE ;
874 006104 025067 DPMSG3 ;"INVALID SIZE - TRY AGAIN"
875 006106 000765 BR 1S ;GO ASK AGAIN
876
877 006110 012700 000400 2S: MOV #256.,RO ;DEFAULT TO 256. BYTE BUFFER
878 006114 005400 3S: NEG RO ;MAKE IT NEG BYTE COUNT
879 006116 010067 011374 MOV RO,CHRCNT ;SAVE IT FOR TEST
880
881 006122 012767 120240 011542 MOV #120240,DHRLVL ;SET BR LEVELS TO BR5
882 006130 016700 011304 MOV DHVCT,RO ;SET UP VECTORS
883 006134 012720 010162 MOV #RINT3,(RO)+ ;
884 006140 116710 011526 MOV# DHRLVL,(RO) ;
885 006144 005720 TST (RO)+ ;
886 006146 012720 007540 MOV #TINT3,(RO)+ ;
887 006152 116710 011515 MOV# DHTLVL,(RO) ;

```

```

890 006155 005067 011716 EXPAT3: CLR PATFLG ;CLEAR <CR> SEQUENCE FLAG
891 006162 105067 015272 CLR EC2 ;CLEAR ECHO BUFFER
892 006165 016701 011244 MOV DHADR,R1 ;INIT R1 TO POINT TO SCR REG
893 006172 104400 TYPE ;"PATTERN TYPE ? (A,U,D,R,S, OR B)"
894 006174 025124 DPMSG4
895 006176 104405 ROCHR
896 006176 012600 7S: MOV (SP)+,RO ;GET WHAT HE TYPED
897 006202 120027 000015 CHPB RO,#15 ;WAS IT A <CR> ??
898 006206 001407 BEQ 9S ;BR IF YES
899 006210 010067 011662 MOV RO,DATPAT
900 006214 110067 015240 MOV#B RO,EC2 ;ECHO IT
901 006220 104400 TYPE
902 006222 023460 EC2
903 006224 000764 BR 7S ;GO WAIT FOR TERMINATOR
904 006226 104400 9S: TYPE
905 006230 025176 DPMSG5 ;"SET SR07=1 TO LOCK ON TEST PATTERN"
906 006232 105767 015222 TSTB EC2 ;<CR> ONLY ??
907 006236 001005 BNE 8S ;BR IF NOT
908 006240 012767 000015 011632 MOV #15,PATFLG
909 006246 000167 000506 JMP DPATCR ;GO SEQUENCE A,U,D,R PATTERNS
910 006252 022767 000101 011616 8S: CMP #101,DATPAT ;ALTERNATING I/O ?
911 006260 001002 BNE 1S ;BR IF NOT
912 006262 000167 000102 JMP DPATA ;GO DO IT
913 006266 022767 000125 011602 1S: CMP #125,DATPAT ;UP COUNT PATTERN ?
914 006274 001002 BNE 2S ;BR IF NOT
915 006276 000167 000164 JMP DPATU ;GO DO IT
916 006302 022767 000104 011566 2S: CMP #104,DATPAT ;DOWN COUNT PATTERN ?
917 006310 001002 BNE 3S ;BR IF NOT
918 006312 000167 000246 JMP DPATD ;GO DO IT
919 006316 022767 000122 011552 3S: CMP #122,DATPAT ;RANDOM PATTERN ?
920 006324 001002 BNE 4S ;BR IF NOT
921 006326 000167 000330 JMP DPATR ;GO DO IT
922 006332 022767 000123 011536 4S: CMP #123,DATPAT ;SINGLE CHAR PATTERN ?
923 006340 001002 BNE 5S ;BR IF NOT
924 006342 000167 000474 JMP DPATS ;GO DO IT
925 006346 022767 000102 011522 5S: CMP #102,DATPAT ;TYPE IN BUFFER ?
926 006354 001002 BNE 6S ;BR IF NOT
927 006356 000167 000620 JMP DFATB ;GO DO IT
928 006362 104400 6S: TYPE
929 006364 025240 DPMSG6 ;"INVALID PATTERN - TRY AGAIN"
930 006366 000673 BR EXPAT3 ;GO ASK AGAIN

```



```

996 006662 005067 011206 DPATR: CLR DATCNT ;INIT ITERATION COUNTER
997 006666 004767 010270 1S: JSR PC,SUPATR ;GO SET UP THE PATTERN
998 006672 004767 000524 JSR PC,DHST2 ;GO EXECUTE IT ON SELECTED DH11
999 006676 005267 011172 INC DATCNT ;COUNT IT
1000 006702 026767 011200 011164 CMP PATLIM,DATCNT ;DONE IT ENOUGH TIMES
1001 006710 001366 BNE 1S ;BR IF NOT DO IT AGAIN
1002
1003 006712 016767 011156 172240 MOV DATCNT,SREGO ;SAVE ITERATION COUNT
1004 006720 005067 011150 CLR DATCNT ;INIT COUNTER
1005 006724 012767 006734 172156 MOV #25,SLPERR ;COME BACK TO 25
1006 006732 104022 ERROR 22 ;REPORT DONE SPECIFIED NO. OF ITERATIONS
1007 006734 022767 000015 011136 2S: CMP #15,PATFLG ;CYCLING FOUR PATTERNS ?
1008 006742 001001 BNE 3S ;BR IF NOT
1009 006744 000207 RTS PC ;RETURN TO EXECUTE NEXT PATTERN
1010 006746 105777 172164 3S: TSTB #SWR ;LOCK ON THIS PATTERN ??
1011 006752 100745 BMT 1S ;BR IF YES
1012 006754 000167 177176 JMP EXPAT3 ;GO ASK FOR NEW PATTERNS
1013
1014 006760 012767 000101 011110 DPATCR: MOV #101,DATPAT ;FLAG 1/0 PATTERN
1015 006766 004767 177376 JSR PC,DPATA ;CALL FOR 1/0 PATTERN
1016 006772 012767 000125 011076 MOV #125,DATPAT ;FLAG UP COUNT PATTERN
1017 007000 004767 177462 JSR PC,DPATU ;CALL FOR UP COUNT PATTERN
1018 007004 012767 000104 011064 MOV #104,DATPAT ;FLAG DOWN COUNT PATTERN
1019 007012 004767 177546 JSR PC,DPATD ;CALL FOR DOWN COUNT PATTERN
1020 007016 012767 000122 011052 MOV #122,DATPAT ;FLAG RANDOM DATA PATTERN
1021 007024 004767 177632 JSR PC,DPATR ;CALL FO RANDOM PATTERN
1022 007030 105777 172102 TSTB #SWR ;LOCK ON ALL FOUR PATTERNS
1023 007034 100751 BMT DPATCR ;BR IF YES
1024 007036 000167 177114 JMP EXPAT3 ;GO ASK FOR NEW PATTERN
1025
1026 007042 105067 014412 DPATS: CLRB EC2 ;CLEAR THE ECHO BUFFER
1027 007046 005067 011022 CLR DATCNT ;INIT ITERATION COUNTER
1028 007052 104400 TYPE ;
1029 007054 025300 DPMSC7 ;
1030 007056 104405 3S: RDCMR ;"TYPE SINGLE TEST CHAR"
1031 007060 012600 MOV (SP)+,R0 ;GET CHAR
1032 007062 122700 000015 CMPB #15,R0 ;GET WHAT HE TYPED
1033 007066 001407 BEQ 4S ;WAS IT A <CR> ??
1034 007070 010067 011006 MOV R0,SINGLE ;BR IF YES
1035 007074 110067 014360 MOVB R0,EC2 ;SAVE IT FOR LOADING BUFFER
1036 007100 104400 TYPE ;ECHO IT ON TTY
1037 007102 023460 EC2 ;
1038 007104 000764 BR 3S ;GO WAIT FOR TERMINATOR
1039
1040 007106 105767 014346 4S: TSTB EC2 ;WAS SINGLE CHAR A <CR> ??
1041 007112 001003 BNE 1S ;BR IF NOT A <CR> ONLY
1042 007114 012767 000015 010760 MOV #15,SINGLE ;SET UP TO LOAD ALL <CR>'S
1043 007122 004767 010114 1S: JSR PC,SUPATS ;GO SET IT UP IN BUFFER
1044 007126 004767 000270 JSR PC,DHST2 ;GO EXECUTE IT ON DH11
1045 007132 005267 010736 INC DATCNT ;COUNT ONE TIME
1046 007136 026767 010744 010730 CMP PATLIM,DATCNT ;DONE REQUIRED ITERATIONS ?
1047 007144 001366 BNE 1S ;BR IF NOT
1048
1049 007146 016767 010722 172004 MOV DATCNT,SREGO ;SAVE ITERATION COUNT

```

1050	007154	005067	010714		CLR	DATCNT	: INIT ITERATION COUNTER
1051	007160	012767	007170	171722	MOV	#25,SLPERR	: COME BACK TO 25 ALWAYS
1052	007166	104023			ERROR	23	: REPORT DONE SINGLE CHAR PATTERN
1053	007170	105777	171742	25:	TSTB	#SWR	: LOCK ON THIS PATTERN ??
1054	007174	100752			BMI	15	: BR IF YES
1055	007176	000167	176754		JMP	EXPAT3	: GO ASK FOR NEW PATTERN
1056							

```

1059 007202 005067 010666 DPATB: CLR DATCNT ;INIT ITERATION COUNTER
1060 007206 104400 TYPE ;"TYPE IN BUFFER - TERMINATE WITH CONTROL-C"
1061 007210 025333 DPMSGA ;POINT TO XMIT BUFFER
1062 007212 012705 030212 MOV #TBUF,R5 ;GET A CHAR
1063 007216 104405 1S: ROCHR ;SAVE IT
1064 007220 012667 010656 MOV (SP)+,SINGLE ;ECHO IT
1065 007224 116767 010652 014226 MOVB SINGLE,EC2
1066 007232 104400 TYPE
1067 007234 023460 EC2
1068
1069 007236 022767 000003 010636 CMP #3,SINGLE ;WAS IT A CONTROL-C ??
1070 007244 001423 BEQ 3S ;BR IF YES
1071
1072 007246 020527 031213 CMP R5,#TBUF+513. ;BUFFER FULL ??
1073 007252 001420 BEQ 3S ;BR IF YES
1074
1075 007254 022767 000012 010620 CMP #12,SINGLE ;WAS IT A LINE FEED ??
1076 007262 001011 BNE 2S ;BR IF NOT
1077
1078 007264 016700 010674 MOV FILLB,R0 ;LOAD LF PLUS FILLERS
1079 007270 116725 010606 MOVB SINGLE,(R5)+
1080 007274 116725 010662 11S: MOVB FILLA,(R5)+ ;LOAD A FILLER CHAR
1081 007300 005300 DEC R0
1082 007302 001374 BNE 11S ;BR TILL REQUIRED FILLERS LOADED
1083 007304 000744 BR 1S ;GO ASK FOR ANOTHER CHAR
1084
1085 007306 116725 010570 2S: MOVB SINGLE,(R5)+ ;LOAD IT IN BUFFER
1086 007312 000741 BR 1S ;GO GET NEXT CHAR
1087
1088 007314 112767 000136 014140 3S: MOVB #136,EC3 ;ECHO CONTROL-C
1089 007322 112767 000103 014133 MOVB #103,EC3+1
1090 007330 104400 TYPE
1091 007332 023462 EC3
1092 007334 162705 030212 SUB #TBUF,R5 ;SET UP CHAR COUNT
1093 007340 005405 NEG R5
1094 007342 010567 010150 MOV R5,CHRCNT
1095 007346 004767 000050 4S: JSR PC,DHST2 ;GO EXECUTE PATTERN
1096 007352 005267 010516 INC DATCNT
1097 007356 026767 010524 010510 CMP PATLIM,DATCNT ;DONE REQUIRED ITERATIONS
1098 007364 001370 BNE 4S ;BR IF NOT
1099
1100 007366 016767 010502 171564 MOV DATCNT,SREGD ;SAVE ITERATION COUNT
1101 007374 005067 010474 CLR DATCNT ;INIT ITERATION COUNTER
1102 007400 012767 007410 171502 MOV #55,SLPERR ;RETURN TO 5S
1103 007406 104024 ERROR 24 ;DONE REQUIRED ITERATIONS
1104 007410 105777 171522 5S: TSTB #SWR ;LOCK ON THIS BUFFER ??
1105 007414 100754 BMT 4S ;BR IF YES
1106 007416 000167 176534 JMP EXPAT3 ;GO ASK FOR NEW PATTERN
1107
    
```

```

1110 007422 004767 004322          DHST2: JSR    PC,DHSET1    ;GO SET UP THE DH11
1111 007426 056761 010016 000012    BIS    LINMSK,BAR(R1) ;ACTIVATE THE LINE
1112 007434 004767 007702          JSR    PC,CHPS1     ;GO CLEAR PSW
1113
1114 007440 012767 000100 010416    PTWAIT: MOV    #100,TIMEA ;INIT TIMERS
1115 007446 005067 010414          CLR    TIMEB
1116 007452 005767 010046          1S:   TST    ROONE    ;DONE ENTIRE PATTERN ?
1117 007456 001023 005416          BNE    3S           ;BR IF YES
1118 007460 004767 005416          JSR    PC,TIMEIT   ;CALL THE TIMER
1119 007464 000772 005416          BR     1S          ;EXECUTED IF NO TIMEOUT
1120
1121 007466 012777 004000 007742    MOV    #BIT11,2DHADR ;CLEAR THE DH11
1122 007474 016767 010374 171456    MOV    DATCNT,SREG0 ;SAVE ITERATION COUNTER
1123 007502 016767 010370 171452    MOV    DATPAT,SREG1 ;SAVE PATTERN TYPE
1124 007510 012767 007520 171372    MOV    #2S,SLPERR  ;SET UP ERROR RETURN
1125 007516 104025 007520 171372    ERROR  2S         ;DATA PATTERNS TEST TIMEOUT ERROR
1126
1127 007520 005726 000167 176430    2S:   TST    (SP)+   ;FIX STACK SINCE WE ARE SKIPPING RTS
1128 007522 000167 176430          JMP    EXPAT3     ;GO ASK FOR NEW PATTERN
1129
1130 007526 052711 004000 000740    3S:   BIS    #BIT11,(R1) ;CLEAR THE DH11
1131 007532 004767 000740          JSR    PC,CKERDP  ;GO CHECK DATA BUFFERS
1132 007536 000207 000740          RTS    PC         ;RETURN TO CONTROL ROUTINE
1133
1134

```

```

1137
1138
1139
1140 007540 032711 002000      TINT3: BIT      #BIT10,(R1)      ;NON EX MEM ERROR ??
1141 007544 001430              BEQ      25                    ;BR IF NOT
1142
1143 007546 011103              MOV      (R1),R3              ;SAVE THE SCR
1144 007550 004767 007602          JSR      PC,CHPS2            ;GO LOCK OUT INTRs
1145 007554 012711 004000          MOV      #BIT11,(R1)        ;CLEAR OUT THE DH11
1146 007560 116704 010112          MOV      LINE,R4            ;SET UP THE S/B DATA
1147 007564 042703 175760          BIC      #175760,R3         ;CLEAR OUT SUPERFLUOUS BITS
1148 007570 010102          MOV      R1,R2              ;SET UP REGADR
1149 007572 004767 004426          JSR      PC,SUER1           ;GO SET UP ERROR INFO
1150 007576 004567 004336          JSR      RS,SUNUM          ;GO SET UP LINE NO. IN MSG
1151 007602 017676          LINE
1152 007604 020232          EM1+44
1153 007606 012767 007616 171274      MOV      #15,SLPERR        ;SET UP THE ERROR LOOP RETURN
1154 007614 104001          ERROR 1                    ;NON EX MEM ERROR
1155 007616 022626          15:  CMP      (SP)+,(SP)+    ;POP THE STACK
1156 007620 005726          TST      (SP)+            ;FIX STACK SINCE NO RTS IS EXECUTED
1157 007622 000167 176330          JMP      EXPAT3           ;GO ASK FOR NEW PATTERN
1158
1159 007626 011103          25:  MOV      (R1),R3         ;GET THE SCR REG CONTENTS
1160 007630 100431          BMI      45              ;BR IF XMIT DONE SET
1161
1162 007632 004767 007520          JSR      PC,CHPS2            ;GO LOCK OUT INTRs
1163 007636 012711 004000          MOV      #BIT11,(R1)        ;CLEAR THE DH11 - FATAL ERROR
1164 007642 012704 100000          MOV      #BIT15,R4         ;SET UP S/B DATA
1165 007646 156704 010024          BISB    LINE,R4
1166 007652 042703 077760          BIC      #77760,R3         ;CLEAR OUT SUPERFLUOUS BITS
1167 007656 010102          MOV      R1,R2              ;SET UP REGADR
1168 007660 004767 004340          JSR      PC,SUER1           ;GO SET UP ERROR INFO
1169 007664 004567 004250          JSR      RS,SUNUM          ;GO SET UP LINE NO. IN MSG
1170 007670 017676          LINE
1171 007672 020414          EM2+54
1172 007674 012767 007704 171206      MOV      #35,SLPERR        ;SET UP ERROR LOOP RETURN
1173 007702 104002          ERROR 2                    ;XMITTR FALSE INTERRUPT
1174 007704 022626          35:  CMP      (SP)+,(SP)+    ;POP THE STACK
1175 007706 005726          TST      (SP)+            ;FIX STACK SINCE NO RTS IS EXECUTED
1176 007710 000167 176242          JMP      EXPAT3           ;GO ASK FOR NEW PATTERN
1177
1178 007714 005761 000012          45:  TST      BAR(R1)         ;DID BAR BIT CLEAR ??
1179 007720 001430          BEQ      65              ;BR IF YES
1180
1181 007722 004767 007430          JSR      PC,CHPS2            ;GO LOCK OUT INTRs
1182 007726 016103 000012          MOV      BAR(R1),R3        ;GET THE WRS DATA
1183 007732 012711 004000          MOV      #BIT11,(R1)        ;CLEAR THE DH11
1184 007736 005004          CLR      R4                ;SET UP S/B DATA
1185 007740 010102          MOV      R1,R2              ;SET UP REGADR
1186 007742 062702 000012          ADD      #BAR,R2
1187 007746 004767 004252          JSR      PC,SUER1           ;GO SET UP ERROR INFO
1188 007752 004567 004162          JSR      RS,SUNUM          ;GO SET UP LINE NO. IN MSG
1189 007756 017676          LINE
1190 007760 020474          EM3+55
    
```

1191	007762	012767	007772	171120	MOV	#5S, SLPERR	:SAVE THE ERROR LOOP RETURN
1192	007770	104003			ERROR	3	:BUFFER ACTIVE REG FAILED TO CLEAR
1193	007772	022626			5S: CMP	(SP)+, (SP)+	:POP GOES THE STACK
1194	007774	005726			TST	(SP)+	:FIX STACK SINCE NO RTS IS EXECUTED
1195	007776	000167	176154		JMP	EXPAT3	:GO ASK FOR NEW PATTERN
1196							
1197	010002	005761	000010		6S: TST	BCR(R1)	:DID BYTE COUNT GO TO ZERO ??
1198	010006	001430			BEG	BS	:BR IF YES
1199							
1200	010010	004767	007342		JSR	PC, CHPS2	:GO LOCK OUT INTRs
1201	010014	016103	000010		MOV	BCR(R1), R3	:GET THE WAS DATA
1202	010020	012711	004000		MOV	#BIT11, (R1)	:CLEAR THE DH11
1203	010024	005004			CLR	R4	:SET UP S/B DATA
1204	010026	010102			MOV	R1, R2	:SET UP REGADR
1205	010030	062702	000010		ADD	#BCR, R2	
1206	010034	004767	004154		JSR	PC, SUER1	:GO SET UP THE ERROR INFO
1207	010040	004567	004074		JSR	RS, SUNUM	:GO SET UP LINE NO. IN MSG
1208	010044	017676			LINE		
1209	010046	020551			EM4+52		
1210	010050	012767	010060	171032	MOV	#7S, SLPERR	:SET UP ERROR LOOP RETURN
1211	010056	104004			ERROR	4	:BYTE COUNT REG FAILED TO GO TO 000000
1212	010060	022626			7S: CMP	(SP)+, (SP)+	:POP GOES THE STACK
1213	010062	005726			TST	(SP)+	:FIX STACK SINCE NO RTS IS EXECUTED
1214	010064	000167	176066		JMP	EXPAT3	:GO ASK FOR NEW PATTERN
1215							
1216	010070	016103	000006		8S: MOV	CAR(R1), R3	:GET THE WAS DATA
1217	010074	016704	007416		MOV	CHRCNT, R4	:SET UP S/B DATA
1218	010100	005404			NEG	R4	
1219	010102	062704	030212		ADD	#TBUF, R4	
1220	010106	020304			CMP	R3, R4	:WAS CAR CORRECT ??
1221	010110	001423			BEG	10S	:BR IF YES
1222							
1223	010112	004767	007240		JSR	PC, CHPS2	:GO LOCK OUT INTRs
1224	010116	010102			MOV	R1, R2	:SET UP REGADR
1225	010120	062702	000006		ADD	#CAR, R2	
1226	010124	004767	004074		JSR	PC, SUER1	:GO SET UP ERROR INFO
1227	010130	004567	004004		JSR	RS, SUNUM	:GO SET UP LINE NO. IN MSG
1228	010134	017676			LINE		
1229	010136	020633			EM5+57		
1230	010140	012767	010150	170742	MOV	#9S, SLPERR	:SET UP THE ERROR RETURN
1231	010146	104005			ERROR	5	:CURRENT ADDRESS REG NOT CORRECT
1232	010150	022626			9S: CMP	(SP)+, (SP)+	:POP THE STACK
1233	010152	005726			TST	(SP)+	:FIX STACK SINCE NO RTS IS EXECUTED
1234	010154	000167	175776		JMP	EXPAT3	:GO ASK FOR NEW PATTERN
1235							
1236	010160	000002			10S: RTI		

```

;RECEIVER INTERRUPT SERVICE ROUTINE THREE
1239
1240
1241 010162 032711 040000 RINT3: BIT #BIT14,(R1) ;SILO OVERFLOW ERROR ??
1242 010166 001427 BEQ 25 ;BR IF NOT
1243
1244 010170 004767 007162 JSR PC,CHPS2 ;GO LOCK OUT INTRs
1245 010174 011103 MOV (R1),R3 ;GET THE WAS DATA
1246 010176 012711 004000 MOV #BIT11,(R1) ;NOW CLEAR THE DH11
1247 010202 042703 177760 BIC #177760,R3 ;CLEAR JUNK
1248 010206 116704 007464 MOV#B LINE,R4
1249 010212 004767 004006 JSR PC,SUER1 ;GO SET UP ERROR INFO
1250 010216 004567 003716 JSR RS,SUNUM ;GO SET UP LINE NO. IN MSG
1251 010222 017676 LINE
1252 010224 020702 EM6+44
1253 010226 012767 010236 170654 MOV #15,SLPERR ;SET UP ERROR LOOP RETURN
1254 010234 104006 ERROR 6 ;SILO OVERFLOW - BAD,BAD,BAD !!!
1255 010236 022626 15: CMP (SP)+,(SP)+ ;POP GOES THE STACK
1256 010240 005726 TST (SP)+ ;FIX STACK SINCE NO RTS IS EXECUTED
1257 010242 000167 175710 JMP EXPAT3 ;GO ASK FOR NEW PATTERN
1258
1259 010246 105711 25: TSTB (R1) ;CHAR AVAIL SET ??
1260 010250 100432 BMI 45 ;BR IF YES
1261
1262 010252 004767 007100 JSR PC,CHPS2 ;GO LOCK OUT INTRs
1263 010256 011103 MOV (R1),R3 ;GET WAS DATA
1264 010260 042703 177560 BIC #177560,R3 ;CLEAN IT UP
1265 010264 012711 004000 MOV #BIT11,(R1) ;NOW CLEAR DH11
1266 010270 012704 000200 MOV #BIT07,R4 ;SET UP S/B DATA
1267 010274 156704 007376 BISB LINE,R4
1268 010300 010102 MOV R1,R2 ;SET UP REGADR
1269 010302 004767 003716 JSR PC,SUER1 ;GO SET UP ERROR INFO
1270 010306 004567 003626 JSR RS,SUNUM ;GO SET UP LINE NO. IN MSG
1271 010312 017676 LINE
1272 010314 020756 EM7+51
1273 010316 012767 010326 170564 MOV #35,SLPERR ;SET UP THE ERROR LOOP RETURN
1274 010324 104007 ERROR 7 ;RECEIVER FALSE INTERRUPT
1275 010326 022626 35: CMP (SP)+,(SP)+ ;POP GOES THE SP
1276 010330 005726 TST (SP)+ ;FIX STACK SINCE NO RTS IS EXECUTED
1277 010332 000167 175620 JMP EXPAT3 ;GO ASK FOR NEW PATTERN
1278
1279 010336 016167 000002 170634 45: MOV NRC(R1),STMPD ;SAVE THE DATA RECEIVED
1280 010344 100427 BMI 65 ;BR IF IT WAS VALID DATA
1281
1282 010346 004767 007004 JSR PC,CHPS2 ;GO LOCK OUT INTRs
1283 010352 012711 004000 MOV #BIT11,(R1) ;NOW CLEAR THE DH11
1284 010356 162767 025732 015342 SUB #RBUF,RBFPTR ;WHICH CHAR WAS IT ??
1285 010364 016702 015336 MOV RBFPTR,R2 ;SAVE CHAR NUMBER
1286 010370 004767 003624 JSR PC,SUER2 ;GO SET UP ERROR INFO
1287 010374 004567 003540 JSR RS,SUNUM ;GO SET UP LINE NO. IN MSG
1288 010400 017676 LINE
1289 010402 021026 EM10+45
1290 010404 012767 010414 170476 MOV #55,SLPERR ;SET UP ERROR RETURN
1291 010412 104010 ERROR 10 ;RECEIVED INVALID DATA
1292 010414 022626 55: CMP (SP)+,(SP)+ ;POP GOES THE STACK
    
```

1293	010416	005726				TST	(SP)+		:FIX STACK SINCE NO RTS IS EXECUTED
1294	010420	000167	175532			JMP	EXPAT3		:GO ASK FOR NEW PATTERN
1295									
1296	010424	016777	170550	015274	6S:	MOV	STMP0, RBFPTR		:STORE CHAR IN THE BUFFER
1297	010432	062767	000001	015266		ADD	#1, RBFPTR		:UPDATE THE POINTER
1298	010440	026767	007062	015260		CMP	RBFEND, RBFPTR		:END OF BUFFER ??
1299	010446	001407				BEQ	7S		:BR IF YES
1300	010450	062767	000001	015250		ADD	#1, RBFPTR		
1301	010456	026767	007044	015242		CMP	RBFEND, RBFPTR		
1302	010464	001003				BNE	8S		:BR IF NOT DONE
1303	010466	052767	000001	007030	7S:	BIS	#1, RDONE		:SET SOFTWARE DONE FLAG
1304	010474	000002			8S:	RTI			:RETURN TO WAIT LOOP

```

1307                                     ;THIS ROUTINE IS CALLED TO CHECK THE RECEIVED DATA AND REPORT ALL ERRORS
1308                                     ;FOR THE DATA PATTERNS TESTS
1309
1310 010476 012767 025732 015222 CKERDP: MOV      @RBUF,RBFPTR ;SET UP POINTERS
1311 010504 012767 030212 015216      MOV      @TBUF,TBFPTR
1312
1313 010512 117704 015212          1S:  MOV@B   @TBFPTR,R4 ;GET THE S/B DATA
1314 010516 000304                SWAB    R4 ;PUT LINE NO. IN HIGH BYTE
1315 010520 105004                CLRB   R4
1316 010522 156704 007150        BISB   LINE,R4
1317 010526 000304                SWAB   R4
1318 010530 052704 100000        BIS    @BIT15,R4 ;AND FINALLY THE VALID DATA BIT
1319 010534 046704 007354        BIC    CLMSK,R4 ;MASK OFF BITS NOT XMITTED
1320 010540 017703 015162        MOV    @RBFPTR,R3 ;GET THE WAS DATA
1321 010544 020304                CMP    R3,R4 ;WAS = S/B ?????
1322 010546 001425                BEQ    3S ;BR IF YES
1323
1324 010550 010367 170442                MOV    R3,$TMP7 ;SAVE THE WAS DATA
1325 010554 010146                MOV    R1,-(SP) ;SAVE THE DEVADR
1326 010556 016701 015144        MOV    RBFPTR,R1 ;GET THE SBADR
1327 010562 016702 015142        MOV    TBFPTR,R2 ;GET THE WASADR
1328 010566 010200                MOV    R2,R0 ;GET XMIT BUFFER ADDR
1329 010570 162700 030212        SUB    @TBUF,R0 ;GENERATE CHAR #
1330 010574 004767 003420        JSR    PC,SUER2 ;GO SET UP ERROR INFO
1331 010600 004567 003334        JSR    RS,SUNUM ;GO PUT LINE NO. IN MSG
1332 010604 017676                LINE
1333 010606 021161                EM11+23
1334 010610 012767 010620 170272    MOV    #2S,$LPERR ;SET UP ERROR RETURN
1335 010616 104011                ERROR  11 ;DATA COMPARE ERROR OR PARITY,FRAMING
1336                                     ;OR OVERRUN
1337 010620 012601          2S:  MOV    (SP)+,R1 ;RESTORE THE DEVADR
1338
1339 010622 005267 015102          3S:  INC    TBFPTR ;UPDATE POINTERS
1340 010626 062767 000001 015072    ADD    #1,RBFPTR
1341 010634 026767 006666 015064    CMP    RBFEND,RBFPTR ;COMPARED ALL CHARS ??
1342 010642 001407                BEQ    4S ;BR IF YES
1343 010644 062767 000001 015054    ADD    #1,RBFPTR ;UPDATE IT AGAIN
1344 010652 026767 006650 015046    CMP    RBFEND,RBFPTR ;DONE YET ?
1345 010660 001314                BNE    1S ;BR IF NOT
1346
1347 010662 000207          4S:  RTS    PC ;RETURN TO WAIT LOOP
1348

```



```

(1) 011006          $SCOPE:
(2) 011006 005067 006664          CLR          LINE          :INIT LINE COUNTER
(1) 011012 032777 040000 170116 1S:  BIT          #BIT14,2SWR  :LOOP ON PRESENT TEST?
(1) 011020 001104          BNE          $OVER          :YES IF SW14=1
(1) 011022 000416          :#####START OF CODE FOR THE XOR TESTER#####
(1) 011024 013746 000004          $XTSTR: BR          6S          :IF RUNNING ON THE "XOR" TESTER CHANGE
(1) 011030 012737 011050 000004          MOV          2#ERRVEC, -(SP)  :THIS INSTRUCTION TO A "NOP" (NOP=240)
(1) 011036 005737 177060          MOV          #5, 2#ERRVEC    :SAVE THE CONTENTS OF THE ERROR VECTOR
(1) 011042 012637 000004          TST          2#177060        :SET FOR TIMEOUT
(1) 011046 000453          MOV          (SP)+, 2#ERRVEC  :TIME OUT ON XOR?
(1) 011050 022626          BR          $SVLAD          :RESTORE THE ERROR VECTOR
(1) 011052 012637 000004          5S:  CMP          (SP)+, (SP)+  :GO TO THE NEXT TEST
(1) 011056 000413          MOV          (SP)+, 2#ERRVEC  :CLEAR THE STACK AFTER A TIME OUT
(1) 011060          BR          7S          :RESTORE THE ERROR VECTOR
(1) 011060 105767 170017          6S:; #####END OF CODE FOR THE XOR TESTER##### :LOOP ON THE PRESENT TEST
(1) 011064 001421          2S:  TSTB         SERFLG      :HAS AN ERROR OCCURRED?
(1) 011066 126767 170023 170007          BC          3S          :BR IF NO
(1) 011074 101015          CMPB        SERMAX, SERFLG   :MAX. ERRORS FOR THIS TEST OCCURRED?
(1) 011076 032777 001000 170032          BHI          3S          :BR IF NO
(1) 011104 001404          BIT          #BIT09, 2SWR    :LOOP ON ERROR?
(1) 011106 016767 167776 167772 7S:  BEQ          4S          :BR IF NO
(1) 011114 000446          MOV          $LPERR, $LPADR   :SET LOOP ADDRESS TO LAST SCOPE
(1) 011116 105067 167761          BR          $OVER          :
(1) 011122 005067 170072          4S:  CLRB         SERFLG      :ZERO THE ERROR FLAG
(1) 011126 000415          CLR          $TIMES          :CLEAR THE NUMBER OF ITERATIONS TO MAKE
(1) 011130 032777 004000 170000 3S:  BR          1S          :ESCAPE TO THE NEXT TEST
(1) 011136 001011          BIT          #BIT11, 2SWR    :INHIBIT ITERATIONS?
(1) 011140 005767 170072          BNE          1S          :BR IF YES
(1) 011144 001406          TST          $PASS          :IF FIRST PASS OF PROGRAM
(1) 011146 005267 167732          BEQ          1S          :INHIBIT ITERATIONS
(1) 011152 026767 170042 167724          INC          $ICNT          :INCREMENT ITERATION COUNT
(1) 011160 002024          CMP          $TIMES, $ICNT   :CHECK THE NUMBER OF ITERATIONS MADE
(1) 011162 012767 000001 167714 1S:  BGE          $OVER          :BR IF MORE ITERATION REQUIRED
(1) 011170 016767 000052 170022          MOV          #1, $ICNT      :REINITIALIZE THE ITERATION COUNTER
(1) 011176 105267 167700          $SVLAD: INCB         $STNM    :SET NUMBER OF ITERATIONS TO DO
(1) 011202 116767 167674 170024          MOV          $STNM, $TESTN   :COUNT TEST NUMBERS
(1) 011210 011667 167672          MOV          (SP), $LPADR    :SET TEST NUMBER IN APT MAILBOX
(1) 011214 011667 167670          MOV          (SP), $LPERR    :SAVE SCOPE LOOP ADDRESS
(1) 011220 005067 167776          CLR          $ESCAPE        :SAVE ERROR LOOP ADDRESS
(1) 011224 112767 000001 167663          MOV          #1, $SERMAX    :CLEAR THE ESCAPE FROM ERROR ADDRESS
(1) 011232 016777 167644 167700 $OVER:  MOV          $STNM, 2$DISPLAY :ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1) 011240 016716 167642          MOV          $LPADR, (SP)   :DISPLAY TEST NUMBER
(1) 011244 000002          RTI          :FUZZ RETURN ADDRESS
(1) 011246 000010          $SMXCNT: 10          :FIXES PS

```

```

1352 (1) ;#####
(1) (1)
(1) .SBTTL ERROR HANDLER ROUTINE
(1) ;#THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
(1) ;#SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
(1) ;#AND GO TO $ERRTYP ON ERROR
(1) ;#THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:

```

```

(1) ;#SW15=1 HALT ON ERROR
(1) ;#SW13=1 INHIBIT ERROR TYPEOUTS
(1) ;#SW09=1 LOOP ON ERROR
(1) ;#CALL
(1) ;* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
(1)
(1) 011250
(1) 011250 105267 167627
(1) 011250 001775
(1) 011254 016777 167620 167654
(1) 011254 005267 167622
(1) 011270 011667 167622
(1) 011274 162767 000002 167614
(1) 011302 117767 167610 167604
(1) 011310 032777 020000 167620
(1) 011316 001004
(1) 011320 004767 000072
(1) 011324 104400 001225
(1) 011330
(1) 011330 122767 000001 167712
(1) 011336 001007
(1) 011340 116767 167550 000004
(1) 011346 004767 001166
(1) 011350 000
(1) 011354 000
(1) 011354 000777
(1) 011356 005777 167554
(1) 011360 100001
(1) 011364 000000
(1) 011366 032777 001000 167542
(1) 011374 001402
(1) 011376 016716 167506
(1) 011402 005767 167614
(1) 011406 001402
(1) 011410 016716 167606
(1) 011414
(1) 011414 000002
1353 ;*****
(1) ;.SBTTL ERROR MESSAGE TYPEOUT ROUTINE
(1)
(1) ;#THIS ROUTINE USES THE "ITEM CONTROL BYTE" (SITEMB) TO DETERMINE WHICH
(1) ;#ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),
(1) ;#AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
(1)
(1) 011416
(1) 011416 104400 001225
(1) 011422 010046
(1) 011424 005000
(1) 011426 153700 001114
(1) 011432 001004
(2) 011434 016746 167456
(2)

```

```

SERROR:
75: INCB SERFLG ;; SET THE ERROR FLAG
BEQ 75 ;; DON'T LET THE FLAG GO TO ZERO
MOV $STNM,$DISPLAY ;; DISPLAY TEST NUMBER AND ERROR FLAG
INC SERRTL ;; INC THE ERROR COUNT
MOV (SP),SERRPC ;; GET ADDRESS OF ERROR INSTRUCTION
SUB #2,SERRPC
MOVB @SERRPC,SITEMB ;; STRIP AND SAVE THE ERROR ITEM CODE
BIT #BIT13,$SWR ;; SKIP TYPEOUT IF SET
BNE 20$ ;; SKIP TYPEOUTS
JSR PC,SERRTYP ;; GO TO USER ERROR ROUTINE
TYPE ,SCRLF

20$: CMPB #APTENV,$ENV ;; RUNNING IN APT MODE
BNE 2$ ;; NO, SKIP APT ERROR REPORT
MOVB $ITEMB,21$ ;; SET ITEM NUMBER AS ERROR NUMBER
JSR PC,$ATY4 ;; REPORT FATAL ERROR TO APT

21$: .BYTE 0
.BYTE 0

22$: BR 22$ ;; APT ERROR LOOP
2$: TST $SWR ;; HALT ON ERROR
BPL 3$ ;; SKIP IF CONTINUE
HALT ;; HALT ON ERROR!
3$: BIT #BIT09,$SWR ;; LOOP ON ERROR SWITCH SET?
BEQ 4$ ;; BR IF NO
MOV $LERR,(SP) ;; FUDGE RETURN FOR LOOPING
TST $ESCAPE ;; CHECK FOR AN ESCAPE ADDRESS
BEQ 5$ ;; BR IF NONE
MOV $ESCAPE,(SP) ;; FUDGE RETURN ADDRESS FOR ESCAPE

5$: RTI ;; RETURN
;*****

```



```

(1)                                     ;*
(1)                                     ;*STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
(1)                                     ;*STYPOS OR STYPOC
(1)                                     ;*CALL:
(1)                                     ;*   MOV    NUM,-(SP)           ;;NUMBER TO BE TYPED
(1)                                     ;*   TYPON                      ;;CALL FOR TYPEOUT
(1)                                     ;*
(1)                                     ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
(1)                                     ;*CALL:
(1)                                     ;*   MOV    NUM,-(SP)           ;;NUMBER TO BE TYPED
(1)                                     ;*   TYPOC                      ;;CALL FOR TYPEOUT
(1)
(1) 011572 017646 000000          STYPOS: MOV    2(SP),-(SP)           ;;PICKUP THE MODE
(1) 011576 116667 000001 000211  MOVB  1(SP),SOFILL          ;;LOAD ZERO FILL SWITCH
(1) 011604 112667 000207          MOVB  (SP)+,SOMODE+1        ;;NUMBER OF DIGITS TO TYPE
(1) 011610 062716 000002          ADD   #2,(SP)              ;;ADJUST RETURN ADDRESS
(1) 011614 000406
(1) 011616 112767 000001 000171  STYPOC: MOVB  #1,SOFILL          ;;SET THE ZERO FILL SWITCH
(1) 011624 112767 000006 000165  MOVB  #6,SOMODE+1          ;;SET FOR SIX(6) DIGITS
(1) 011632 112767 000005 000154  STYPON: MOVB  #5,SOCNT          ;;SET THE ITERATION COUNT
(1) 011640 010346          MOV   R3,-(SP)             ;;SAVE R3
(1) 011642 010446          MOV   R4,-(SP)             ;;SAVE R4
(1) 011644 010546          MOV   R5,-(SP)             ;;SAVE R5
(1) 011646 116704 000145          MOVB  SOMODE+1,R4          ;;GET THE NUMBER OF DIGITS TO TYPE
(1) 011652 005404          NEG   R4
(1) 011654 062704 000006          ADD   #6,R4                ;;SUBTRACT IT FOR MAX. ALLOWED
(1) 011660 110467 000132          MOVB  R4,SOMODE           ;;SAVE IT FOR USE
(1) 011664 116704 000125          MOVB  SOFILL,R4           ;;GET THE ZERO FILL SWITCH
(1) 011670 016605 000012          MOV   12(SP),R5           ;;PICKUP THE INPUT NUMBER
(1) 011674 005003          CLR   R3                  ;;CLEAR THE OUTPUT WORD
(1) 011676 006105          18:  ROL   R5                ;;ROTATE MSB INTO "C"
(1) 011700 000404          BR    36
(1) 011702 006105          25:  ROL   R5                ;;GO DO MSB
(1) 011704 006105          ROL   R5                ;;FORM THIS DIGIT
(1) 011706 006105          ROL   R5
(1) 011710 010503          MOV   R5,R3
(1) 011712 006103          35:  ROL   R3                ;;GET LSB OF THIS DIGIT
(1) 011714 105367 000076          DECB  SOMODE              ;;TYPE THIS DIGIT?
(1) 011720 100016          BPL   75                  ;;BR IF NO
(1) 011722 042703 177770          BIC   #177770,R3          ;;GET RID OF JUNK
(1) 011726 001002          BNE   45                  ;;TEST FOR 0
(1) 011730 005704          TST   R4                  ;;SUPPRESS THIS 0?
(1) 011732 001403          BEQ   55                  ;;BR IF YES
(1) 011734 005204          45:  INC   R4                ;;DON'T SUPPRESS ANYMORE 0'S
(1) 011736 052703 000060          BIS   #'0,R3              ;;MAKE THIS DIGIT ASCII
(1) 011742 052703 000040          55:  BIS   #' ,R3           ;;MAKE ASCII IF NOT ALREADY
(1) 011746 110367 000040          MOVB  R3,R5                ;;SAVE FOR TYPING
(1) 011752 104400 012012          TYPE  #5                    ;;GO TYPE THIS DIGIT
(1) 011756 105367 000032          75:  DECB  SOCNT            ;;COUNT BY 1
(1) 011762 003347          BGT   25                  ;;BR IF MORE TO DO
(1) 011764 002402          BLT   65                  ;;BR IF DONE
(1) 011766 005204          INC   R4                    ;;INSURE LAST DIGIT ISN'T A BLANK
(1) 011770 000744          BR    25                  ;;GO DO THE LAST DIGIT
(1) 011772 012605          65:  MOV   (SP)+,R5          ;;RESTORE R5

```



```

(1) 012142 005720
(1) 012144 020027 000010
(1) 012150 002746
(1) 012152 003002
(1) 012154 010502
(1) 012156 000764
(1) 012160 105726 8S:
(1) 012162 100003
(1) 012164 116663 177777 177776
(1) 012172 105013 9S:
(3) 012174 012605
(3) 012176 012603
(3) 012200 012602
(3) 012202 012601
(3) 012204 012600
(1) 012206 104400 012234
(1) 012212 016666 000002 000004
(1) 012220 012616
(1) 012222 000002
(1) 012224 023420
(1) 012226 001750
(1) 012230 000144
(1) 012232 000012
(1) 012234 000004

```

```

TST (R0)+
CMP R0,#10
BLT 2S
BGT 8S
MOV R5,R2
BR 6S
8S: TSTB (SP)+
BPL 9S
MOVB -1(SP),-2(R3)
9S: CLRB (R3)
MOV (SP)+,R5
MOV (SP)+,R3
MOV (SP)+,R2
MOV (SP)+,R1
MOV (SP)+,R0
TYPE $DBLK
MOV 2(SP),4(SP)
MOV (SP)+,(SP)
RTI
SDTBL: 10000.
1000.
100.
10.
$DBLK: .BLKW 4

```

```

;; JUST INCREMENTING
;; CHECK THE TABLE INDEX
;; GO DO THE NEXT DIGIT
;; GO TO EXIT
;; GET THE LSD
;; GO CHANGE TO ASCII
;; WAS THE LSD THE FIRST NON-ZERO?
;; BR IF NO
;; YES--SET THE SIGN FOR TYPING
;; SET THE TERMINATOR
;; POP STACK INTO R5
;; POP STACK INTO R3
;; POP STACK INTO R2
;; POP STACK INTO R1
;; POP STACK INTO R0
;; NOW TYPE THE NUMBER
;; ADJUST THE STACK
;; RETURN TO USER

```

1356

.SBTTL TYPE ROUTINE

```

;#ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
;#THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
;#NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
;#NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
;#NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
;#
;#CALL:
;#1) USING A TRAP INSTRUCTION
;# TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
;#OR
;# TYPE
;# MESADR
;#

```

```

(1) 012244 105767 166705
(1) 012250 100002
(1) 012252 000000
(1) 012254 000430
(1) 012256 010046
(1) 012260 017600 000002
(1) 012264 122767 000001 166756
(1) 012272 001011
(1) 012274 132767 000100 166747
(1) 012302 001405
(1) 012304 010067 000004
(1) 012310 004767 000214

```

```

STYPE: TSTB STPFLG
BPL 1S
HALT
BR 3S
1S: MOV R0,-(SP)
MOV 2(SP),R0
CMPB @APTENV,SENV
BNE 62S
BITB @APTPOOL,SENV
BEQ 62S
MOV R0,61S
JSR PC,$ATY3

```

```

;; IS THERE A TERMINAL?
;; BR IF YES
;; HALT HERE IF NO TERMINAL
;; LEAVE
;; SAVE R0
;; GET ADDRESS OF ASCIZ STRING
;; RUNNING IN APT MODE
;; NO, GO CHECK FOR APT CONSOLE
;; SPOOL MESSAGE TO APT
;; NO, GO CHECK FOR CONSOLE
;; SETUP MESSAGE ADDRESS FOR APT
;; SPOOL MESSAGE TO APT

```

```

(1) 012314 000000 61$: WORD 0 :: MESSAGE ADDRESS
(1) 012316 132767 000040 166725 62$: BITB #APTCSUP,SENVH :: APT CONSOLE SUPPRESSED
(1) 012324 001003 BNE 60$ :: YES, SKIP TYPE OUT
(1) 012326 112046 2$: MOVB (RO)+,-(SP) :: PUSH CHARACTER TO BE TYPED ONTO STACK
(1) 012330 001005 BNE 4$ :: BR IF IT ISN'T THE TERMINATOR
(1) 012332 005726 TST (SP)+ :: IF TERMINATOR POP IT OFF THE STACK
(1) 012334 012600 60$: MOV (SP)+,RO :: RESTORE RO
(1) 012336 062716 000002 3$: ADD #2,(SP) :: ADJUST RETURN PC
(1) 012342 000002 RTI :: RETURN
(1) 012344 122716 000011 4$: CMPB #THT,(SP) :: BRANCH IF <HT>
(1) 012350 001426 BEQ 8$ :: BRANCH IF NOT <CRLF>
(1) 012352 122716 000200 CMPB #TCRLF,(SP)
(1) 012356 001004 BNE 5$
(1) 012360 005726 TST (SP)+ :: POP <CR><LF> EQUIV
(1) 012362 104400 TYPE :: TYPE A CR AND LF
(1) 012364 001225 SCRLF
(1) 012366 000757 BR 2$ :: GET NEXT CHARACTER
(1) 012370 004767 000056 5$: JSR PC,STYPEC :: GO TYPE THIS CHARACTER
(1) 012374 126726 166554 6$: CMPB $FILLC,(SP)+ :: IS IT TIME FOR FILLER CHARS.?
(1) 012400 001352 BNE 2$ :: IF NO GO GET NEXT CHAR.
(1) 012402 016746 166544 MOV $NULL,-(SP) :: GET # OF FILLER CHARS. NEEDED
(1) 012406 105366 000001 7$: DECB 1(SP) :: AND THE NULL CHAR.
(1) 012412 002770 BLT 6$ :: DOES A NULL NEED TO BE TYPED?
(1) 012414 004767 000032 JSR PC,STYPEC :: BR IF NO—GO POP THE NULL OFF OF STACK
(1) 012420 105367 000072 DECB $CHARCNT :: GO TYPE A NULL
(1) 012424 000770 BR 7$ :: DO NOT COUNT AS A COUNT
(1) :: LOOP
(1) ;HORIZONTAL TAB PROCESSOR
(1) 012426 112716 000040 8$: MOVB #40,(SP) :: REPLACE TAB WITH SPACE
(1) 012432 004767 000014 9$: JSR PC,STYPEC :: TYPE A SPACE
(1) 012436 132767 000007 000052 BITB #7,$CHARCNT :: BRANCH IF NOT AT
(1) 012444 001372 BNE 9$ :: TAB STOP
(1) 012446 005726 TST (SP)+ :: POP SPACE OFF STACK
(1) 012450 000726 BR 2$ :: GET NEXT CHARACTER
(1) 012452 105777 166470 STYPEC: TSTB #STPS :: WAIT UNTIL PRINTER IS READY
(1) 012456 100375 BPL STYPEC
(1) 012460 116677 000002 166462 MOVB 2(SP),#STPB :: LOAD CHAR TO BE TYPED INTO DATA REG.
(1) 012466 122766 000015 000002 CMPB #15,2(SP) :: BRANCH IF
(1) 012474 001003 BNE 1$ :: NOT <CR>
(1) 012476 105067 000014 CLRB $CHARCNT
(1) 012502 000406 BR STYPEC :: EXIT
(1) 012504 122766 000012 000002 1$: CMPB #12,2(SP) :: BRANCH IF
(1) 012512 002002 BGE STYPEC :: <LF>
(1) 012514 105227 INCB (PC)+ :: INC SPACE
(1) 012516 000000 $CHARCNT: WORD 0 :: COUNT
(1) 012520 000207 STYPEC: RTS PC
(1) :: EQUATES
(1) THT=11
(1) TCRLF=200
(1) ;*****

```

```

(1)          .SBTTL  APT COMMUNICATIONS ROUTINE
(1) 012522 112767 000001 000236 $ATY1:  MOV  #1,SFFLG ;TO REPORT FATAL ERROR
(1) 012530 112767 000001 000226 $ATY3:  MOV  #1,SMFLG ;TO TYPE A MESSAGE
(1) 012536 000403          BR      SATYC
(1) 012540 112767 000001 000220 $ATY4:  MOV  #1,SFFLG ;TO ONLY REPORT FATAL ERROR
(2) 012546          $ATYC:
(3) 012546 010046          MOV   RO,-(SP) ;:PUSH RO ON STACK
(3) 012550 010146          MOV   R1,-(SP) ;:PUSH R1 ON STACK
(1) 012556 105767 000206          TSTB  SMFLG ;:SHOULD TYPE A MESSAGE?
(1) 012556 001450          BEQ   SS ;:IF NOT: BR
(1) 012560 122767 000001 166462  CMPB  #APTENV,SENV ;:OPERATING UNDER APT?
(1) 012566 001031          BNE   SS ;:IF NOT: BR
(1) 012570 132767 000100 166453  BITB  #APTSPool,SENVH ;:SHOULD SPOOL MESSAGES?
(1) 012576 001425          BEQ   SS ;:IF NOT: BR
(1) 012600 017600 000004          MOV   #4(SP),RO ;:GET MESSAGE ADDR.
(1) 012604 062766 000002 000004  ADD   #2,4(SP) ;:BUMP RETURN ADDR.
(1) 012612 005767 166412 1S:  TST   SMSGTYPE ;:SEE IF DONE W/ LAST XMISSION?
(1) 012616 001375          BNE   1S ;:IF NOT: WAIT
(1) 012620 010067 166420          MOV   RO,SMSGAD ;:PUT ADDR IN MAILBOX
(1) 012624 105720          TSTB  (RO)+ ;:FIND END OF MESSAGE
(1) 012626 001375          BNE   2S
(1) 012630 166700 166410          SUB   SMSGAD,RO ;:SUB START OF MESSAGE
(1) 012634 006200          ASR   RO ;:GET MESSAGE LNTH IN WORDS
(1) 012636 010067 166404          MOV   RO,SMSGLGT ;:PUT LENGTH IN MAILBOX
(1) 012642 012767 000004 166360  MOV   #4,SMSGTYPE ;:TELL APT TO TAKE MSG.
(1) 012650 000413          BR    SS
(1) 012652 017667 000004 000016 3S:  MOV   #4(SP),4S ;:PUT MSG ADDR IN JSR LINKAGE
(1) 012660 062766 000002 000004  ADD   #2,4(SP) ;:BUMP RETURN ADDRESS
(3) 012666 016746 165104          MOV   177776,-(SP) ;:PUSH 177776 ON STACK
(1) 012672 004767 177346          JSR   PC,STYPE ;:CALL TYPE MACRO
(1) 012676 000000          4S:  .WORD 0
(1) 012700          5S:
(1) 012700 105767 000062          10S: TSTB  SFFLG ;:SHOULD REPORT FATAL ERROR?
(1) 012704 001416          BEQ   12S ;:IF NOT: BR
(1) 012706 005767 166336          TST   SENV ;:RUNNING UNDER APT?
(1) 012712 001413          BEQ   12S ;:IF NOT: BR
(1) 012714 005767 166310          11S: TST   SMSGTYPE ;:FINISHED LAST MESSAGE?
(1) 012720 001375          BNE   11S ;:IF NOT: WAIT
(1) 012722 017667 000004 166302  MOV   #4(SP),SFATAL ;:GET ERROR #
(1) 012730 062766 000002 000004  ADD   #2,4(SP) ;:BUMP RETURN ADDR.
(1) 012736 005267 166266          INC   SMSGTYPE ;:TELL APT TO TAKE ERROR
(1) 012742 105067 000020          12S: CLRB  SFFLG ;:CLEAR FATAL FLAG
(1) 012746 105067 000013          CLRB  SLFLG ;:CLEAR LOG FLAG
(1) 012752 105067 000006          CLRB  SMFLG ;:CLEAR MESSAGE FLAG
(3) 012756 012601          MOV   (SP)+,R1 ;:POP STACK INTO R1
(3) 012760 012600          MOV   (SP)+,RO ;:POP STACK INTO RO
(1) 012762 000207          RTS   PC ;:RETURN
(1) 012764 000          SMFLG: .BYTE 0 ;:MESSG. FLAG
(1) 012765 000          SLFLG: .BYTE 0 ;:LOG FLAG
(1) 012766 000          SFFLG: .BYTE 0 ;:FATAL FLAG
(1) 012770          .EVEN
(1) 000200          APTSIZE=200
(1) 000001          APTENV=001

```

```

(1) 000100 APTSPool=100
(1) 000040 APTCSUP=040
1358 ;*****
(1) .SBTTL TTY INPUT ROUTINE
(1) ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1) ;*CALL:
(1) ;* RDCHR ;: INPUT A SINGLE CHARACTER FROM THE TTY
(1) ;* RETURN HERE ;: CHARACTER IS ON THE STACK
(1) ;
(1) SRDCHR: MOV (SP), -(SP) ;: PUSH DOWN THE PC
(1) 012770 011646 MOV 4(SP), 2(SP) ;: SAVE THE PS
(1) 012772 016666 000004 000002 1S: TSTB 2$TKS ;: WAIT FOR
(1) 013000 105777 166136 BPL 1$ ;: A CHARACTER
(1) 013004 100375 MOV 2$TKB, 4(SP) ;: READ THE TTY
(1) 013006 117766 166132 000004 BIC 2$C<177>, 4(SP) ;: GET RID OF JUNK IF ANY
(1) 013014 042766 177600 000004 RTI ;: GO BACK TO USER
(1) 013022 000002 ;
(2) ;*****
(1) ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1) ;*CALL:
(1) ;* RDLIN ;: INPUT A STRING FROM THE TTY
(1) ;* RETURN HERE ;: ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1) ;* ;: TERMINATOR WILL BE A BYTE OF ALL 0'S
(1) SRDLIN: MOV R3, -(SP) ;: SAVE R3
(1) 013024 010346 1S: MOV #TTYIN, R3 ;: GET ADDRESS
(1) 013026 012703 013132 2S: CMP #TTYIN+8., R3 ;: BUFFER FULL?
(1) 013032 022703 013142 BLOS 4$ ;: BR IF YES
(1) 013036 101405 RDCHR ;: GO READ ONE CHARACTER FROM THE TTY
(1) 013040 104405 MOV (SP)+, (R3) ;: GET CHARACTER
(1) 013042 112613 CMPB #177, (R3) ;: IS IT A RUBOUT
(1) 013044 122713 000177 BNE 3$ ;: SKIP IF NOT
(1) 013050 001003 4S: TYPE 2$QUES ;: TYPE A '?'
(1) 013052 104400 001224 BR 1$ ;: CLEAR THE BUFFER AND LOOP
(1) 013056 000763 3S: MOV (R3), 9$ ;: ECHO THE CHARACTER
(1) 013060 111367 000044 TYPE 9$
(1) 013064 104400 013130 CMPB 15, (R3)+ ;: CHECK FOR RETURN
(1) 013070 122723 000015 BNE 2$ ;: LOOP IF NOT RETURN
(1) 013074 001356 CLR 1(R3) ;: CLEAR RETURN (THE 15)
(1) 013076 105063 177777 TYPE 15F ;: TYPE A LINE FEED
(1) 013102 104400 001226 MOV (SP)+, R3 ;: RESTORE R3
(1) 013106 012603 MOV (SP), -(SP) ;: ADJUST THE STACK AND PUT ADDRESS OF THE
(1) 013110 011646 MOV 4(SP), 2(SP) ;: FIRST ASCII CHARACTER ON IT
(1) 013112 016666 000004 000002 MOV 2$TTYIN, 4(SP)
(1) 013120 012766 013132 000004 RTI ;: RETURN
(1) 013126 000002 9S: .BYTE 0 ;: STORAGE FOR ASCII CHAR. TO TYPE
(1) 013130 000 .BYTE 0 ;: TERMINATOR
(1) 013131 000 STTYIN: .BLKB 8. ;: RESERVE 8 BYTES FOR TTY INPUT
(1) 013132 000010 ;*****
(1) .SBTTL READ AN OCTAL NUMBER FROM THE TTY
(1)
(1)
(1)

```

```

(1)                                     ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
(1)                                     ;*CHANGE IT TO BINARY.
(1)                                     ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
(1)                                     ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
(1)                                     ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
(1)                                     ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
(1)                                     ;*CALL:
(1)                                     ;*
(1)                                     ;*      RDOCT          ;: READ AN OCTAL NUMBER
(1)                                     ;*      RETURN HERE   ;: LOW ORDER BITS ARE ON TOP OF THE STACK
(1)                                     ;*                  ;: HIGH ORDER BITS ARE IN SHIOCT
(1)
(1) 013142 011646 000004 000002 SRDOCT: MOV      (SP), -(SP)      ;: PROVIDE SPACE FOR THE
(1) 013144 016666 000004 000002      MOV      4(SP), 2(SP)      ;: INPUT NUMBER
(3) 013152 010046 000004 000002      MOV      R0, -(SP)      ;: PUSH R0 ON STACK
(3) 013154 010146 000004 000002      MOV      R1, -(SP)      ;: PUSH R1 ON STACK
(3) 013156 010246 000004 000002      MOV      R2, -(SP)      ;: PUSH R2 ON STACK
(1) 013160 104406 000004 000002 1S:  ROLIN      ;: READ AN ASCII LINE
(1) 013162 012600 000004 000002      MOV      (SP)+, R0      ;: GET ADDRESS OF 1ST CHARACTER
(1) 013164 010067 000100      MOV      R0, 5S        ;: AND SAVE IT
(1) 013170 005001 000004 000002      CLR      R1            ;: CLEAR DATA WORD
(1) 013172 005002 000004 000002      CLR      R2
(1) 013174 112046 000004 000002 2S:  MOVVB     (R0)+, -(SP)      ;: PICKUP THIS CHARACTER
(1) 013176 001420 000004 000002      BEQ      3S            ;: IF ZERO GET OUT
(1) 013200 122716 000060      CMPB     #'0', (SP)     ;: MAKE SURE THIS CHARACTER
(1) 013204 003026 000060      BGT      4S            ;: IS AN OCTAL DIGIT
(1) 013206 122716 000067      CMPB     #'7', (SP)
(1) 013212 002423 000067      BLT      4S
(1) 013214 006301 000067      ASL      R1            ;: #2
(1) 013216 006102 000067      ROL      R2
(1) 013220 006301 000067      ASL      R1            ;: #4
(1) 013222 006102 000067      ROL      R2
(1) 013224 006301 000067      ASL      R1            ;: #8
(1) 013226 006102 000067      ROL      R2
(1) 013230 042716 177770      BIC      #'C7', (SP)    ;: STRIP THE ASCII JUNK
(1) 013234 062601 000067      ADD      (SP)+, R1      ;: ADD IN THIS DIGIT
(1) 013236 000756 000067      BR       2S            ;: LOOP
(1) 013240 005726 000067 3S:  TST      (SP)+          ;: CLEAN TERMINATOR FROM STACK
(1) 013242 010166 000012      MOV      R1, 12(SP)     ;: SAVE THE RESULT
(1) 013246 010267 000026      MOV      R2, SHIOCT
(3) 013252 012602 000026      MOV      (SP)+, R2      ;: POP STACK INTO R2
(3) 013254 012601 000026      MOV      (SP)+, R1      ;: POP STACK INTO R1
(3) 013256 012600 000026      MOV      (SP)+, R0      ;: POP STACK INTO R0
(1) 013260 000002 000026      RTI
(1) 013262 005726 000026 4S:  TST      (SP)+          ;: CLEAN PARTIAL FROM STACK
(1) 013264 105010 000026      CLRB     (R0)          ;: SET A TERMINATOR
(1) 013266 104400 000026      TYPE     ;: TYPE UP THRU THE BAD CHAR.
(1) 013270 000000 000026 5S:  .WORD    0
(1) 013272 104400 001224      TYPE     'SQUES
(1) 013276 000730 000026      BR       1S            ;: "?" "CR" & "LF"
(1) 013300 000000 000026 SHIOCT: .WORD    0      ;: TRY AGAIN
;***** HIGH ORDER BITS GO HERE *****
(1)
(1) .SBTTL READ A DECIMAL NUMBER FROM THE TTY
(1)

```

1360

(1) 013450 000000
(1) 013452 004400 001224
(1) 013456 000720
1361

6S: WORD 0 ; POINTER GOES HERE
TYPE 0 SQUES ; "?" "CR" & "LF"
BR 1S ; TRY AGAIN
;*****

.SBTTL TRAP DECODER

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

(1) 013460 010046
(1) 013462 016600 000002
(1) 013466 005740
(1) 013470 111000
(1) 013472 006300
(1) 013474 016000 013502
(1) 013500 000200

STRAP: MOV RO, -(SP) ; SAVE RO
MOV 2(SP), RO ; GET TRAP ADDRESS
TST -(RO) ; BACKUP BY 2
MOVB (RO), RO ; GET RIGHT BYTE OF TRAP
ASL RO ; POSITION FOR INDEXING
MOV STRPAD(RO), RO ; INDEX TO TABLE
RTS RO ; GO TO ROUTINE

.SBTTL TRAP TABLE

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

ROUTINE

STRPAD: STYPE ; CALL=TYPE TRAP+0(104400) TTY TYPEOUT ROUTINE
STYPOC ; CALL=TYPOC TRAP+1(104401) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
STYPOS ; CALL=TYPOS TRAP+2(104402) TYPE OCTAL NUMBER (NO LEADING ZEROS)
STYPON ; CALL=TYPON TRAP+3(104403) TYPE OCTAL NUMBER (AS PER LAST CALL)
STYPOS ; CALL=TYPDS TRAP+4(104404) TYPE DECIMAL NUMBER (WITH SIGN)
SRDCHR ; CALL=RDCHR TRAP+5(104405) TTY TYPEIN CHARACTER ROUTINE
SRDLIN ; CALL=RDLIN TRAP+6(104406) TTY TYPEIN STRING ROUTINE
SRDOCT ; CALL=RDOCT TRAP+7(104407) READ AN OCTAL NUMBER FROM TTY
SRDDEC ; CALL=RDOEC TRAP+10(104410) READ A DECIMAL NUMBER FROM TTY
;*****

(3) 013502
(3) 013502 012244
(3) 013504 011616
(3) 013506 011572
(3) 013510 011632
(3) 013512 012020
(3) 013514 012770
(3) 013516 013024
(3) 013520 013142
(3) 013522 013302
1362

.SBTTL POWER DOWN AND UP ROUTINES

:POWER DOWN ROUTINE
\$PWRDN: MOV #SILLUP, @#PWRVEC ; SET FOR FAST UP
MOV #340, @#PWRVEC+2 ; PRIO:7
MOV RO, -(SP) ; PUSH RO ON STACK
MOV R1, -(SP) ; PUSH R1 ON STACK
MOV R2, -(SP) ; PUSH R2 ON STACK
MOV R3, -(SP) ; PUSH R3 ON STACK
MOV R4, -(SP) ; PUSH R4 ON STACK
MOV R5, -(SP) ; PUSH R5 ON STACK
MOV SP, \$SAVR6 ; SAVE SP
MOV @#PWRUP, @#PWRVEC ; SET UP VECTOR
HALT

(1) 013524 012737 013652 000024
(1) 013532 012737 000340 000026
(3) 013540 010046
(3) 013542 010146
(3) 013544 010246
(3) 013546 010346
(3) 013550 010446
(3) 013552 010546
(1) 013554 010667 000076
(1) 013560 012737 013572 000024
(1) 013566 000000

```

(1) 013570 000776 BR .-2 ;;HANG UP
(1)
(1)
(1) 013572 016706 000060 .POWER UP ROUTINE
(1) 013576 005067 000054 $PWRUP: MOV $SAVR6, SP ;;GET SP
(1) 013602 005267 000050 1S: CLR $SAVR6 ;;WAIT LOOP FOR THE TTY
(1) 013606 001375 BNE 1S ;;WAIT FOR THE INC
(3) 013610 012605 MOV (SP)+, R5 ;;OF WORD
(3) 013612 012604 MOV (SP)+, R4 ;;POP STACK INTO R5
(3) 013614 012603 MOV (SP)+, R3 ;;POP STACK INTO R4
(3) 013616 012602 MOV (SP)+, R2 ;;POP STACK INTO R3
(3) 013620 012601 MOV (SP)+, R1 ;;POP STACK INTO R2
(3) 013622 012600 MOV (SP)+, R0 ;;POP STACK INTO R1
(1) 013624 012737 013524 000024 MOV $SPWRON, $PWRVEC ;;SET UP THE POWER DOWN VECTOR
(1) 013632 012737 000340 000026 MOV $340, $PWRVEC+2 ;;PRIO:7
(1) 013640 104400 TYPE REPORT THE POWER FAILURE
(1) 013642 013660 SPWRMG: .WORD SPOWER ;;POWER FAIL MESSAGE POINTER
(1) 013644 012716 MOV (PC)+, (SP) ;;RESTART AT CKRST1
(1) 013646 013670 SPWRAD: .WORD CKRST1 ;;RESTART ADDRESS
(1) 013650 000002 RTI
(1) 013652 000000 $ILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED
(1) 013654 000776 BR .-2 ;;BEFORE THE POWER DOWN WAS COMPLETE
(1) 013656 000000 $SAVR6: 0 ;;PUT THE SP HERE
(1) 013660 005015 047520 042527 SPOWER: .ASCIZ <15><12>"POWER"
(1) 013666 000122 .EVEN

```

```

1365 ;*****
1366 ;COMMON DH11 SERVICE ROUTINES
1367 ;*****
1368
1369 ;THESE ROUTINES DETERMINE RESTART ADDRESS AFTER SYSTEM ERROR
1370 ;(BUS ERROR,RSVD INSTR ERROR, OR POWER FAIL)
1371
1372 013670 005767 004176 CKRST1: TST DPFLG ;IN PATTERNS TEST ?
1373 013674 001005 BNE 1$ ;BR IF YES
1374 013676 005767 004202 TST RETFLG ;IN ECHO TEST ?
1375 013702 001004 BNE 2$ ;BR IF YES
1376 013704 000167 166422 JMP RSTRTA ;GO RESTART RELIABILITY TESTS
1377 013710 000167 172070 1$: JMP EXPAT ;GO TO PATTERNS TESTS
1378 013714 000167 170664 2$: JMP ECHO ;GO TO ECHO TESTS
1379
1380 013720 005767 004146 CKRST2: TST DPFLG ;IN PATTERNS TEST ?
1381 013724 001005 BNE 1$ ;BR IF YES
1382 013726 005767 004152 TST RETFLG ;IN ECHO TEST ?
1383 013732 001004 BNE 2$ ;BR IF YES
1384 013734 000167 166362 JMP REST1 ;GO RESTART RELIABILITY TESTS
1385 013740 000167 172040 1$: JMP EXPAT ;GO TO PATTERNS TESTS
1386 013744 000167 170634 2$: JMP ECHO ;GO TO ECHO TESTS
1387
1388
1389 ;THIS ROUTINE IS CALLED TO SET UP THE DH11 PARAMETERS PRIOR TO TEST
1390
1391 013750 012711 004000 DHSET1: MOV #BIT11,(R1) ;CLEAR THE DH11 UNDER TEST
1392 013754 004767 003376 JSR PC,CHPS2 ;GO LOCK OUT INTR
1393 013760 012711 030100 MOV #30100,(R1) ;ENABLE INTERRUPTS ON XMIT DONE
1394 ;NON-EX MEM, DATA AVAIL, OR SILO OVFLW
1395 013764 156711 003706 BISB LINE,(R1) ;SELECT THE LINE NO.
1396 013770 005067 003530 CLR ROONE ;CLEAR SOFTWARE DONE FLAG
1397 013774 012767 025732 011724 MOV #RBUF,RBFPTR ;SET UP RCVR BUFFER POINTER
1398 014002 012767 025732 003516 MOV #RBUF,RBFEND ;MARK END OF THIS BUFFER
1399 014010 016705 003502 MOV CHCNT,RS ;GET CHAR COUNT
1400 014014 005405 NEG RS ;MAKE IT POSITIVE
1401 014016 060567 003504 ADD RS,RBFEND
1402 014022 016761 003464 000004 MOV CUALPR,LPR(R1) ;LOAD THE LPR REG
1403 014030 016761 003462 000010 MOV CHCNT,BCR(R1) ;LOAD THE BYTE COUNT REG
1404 014036 012761 030212 000006 MOV #RBUF,CAR(R1) ;LOAD CURRENT ADDRESS REG
1405 014044 000207 RTS PC ;RETURN
1406
1407 ;THIS ROUTINE IS CALLED TO SELECT A NEW LINE NO. BASED ON THE
1408 ;VALUE OF THE LINE SELECTION PARAMETER
1409
1410 ;CALLING SEQUENCE:
1411
1412 ;JSR PC,SELINE ;CALL THE ROUTINE
1413 ;BR 1$ ;EXIT BRANCH-ROUTINE MOVES THE RETURN
1414 ;PC AROUND THIS BR IF MORE LINES ARE
1415 ;YET TO BE TESTED
1416
1417 014046 105767 003625 SELINE: TSTB LINE+1 ;FIRST TIME THROUGH FOR ANY TEST ?
1418 014052 001010 BNE 1$ ;BR IF NOT
    
```

```

1419 014054 105167 003617 COMB LINE+1 ;SET ENTRY FLAG
1420 014060 012767 000001 003362 MOV #1,LINMSK ;INIT SELECT TEST MASK TO TEST LINE 00
1421 014066 105067 003604 CLR# LINE ;START WITH LINE #00
1422 014072 000405 BR 25 ;GO TEST FOR LINE #00
1423 014074 105267 003576 15: INCB LINE ;GENERATE NEW LINE NO.
1424 014100 006367 003344 ASL LINMSK ;SHIFT SELECT MASK TO TEST NXT LINE
1425 014104 001407 BEQ 35 ;RETURN TO EXIT BRANCH - ALL LINES DONE
1426 014106 036767 003336 003332 25: BIT LINMSK,LINSEL ;IS THE LINE SELECTED FOR TEST ??
1427 014114 001767 BEQ 15 ;BR IF NOT
1428 014116 062716 000002 ADD #2,(SP) ;MOVE RETURN PC AROUND EXIT BRANCH
1429 014122 000402 BR 45 ;RETURN TO TEST SELECTED LINE
1430 014124 005067 003546 35: CLR LINE ;INIT ENTRY FLAG AND LINE NO. TO 000
1431 014130 142777 000017 003300 45: BICB #17,DZDADR ;INIT LINE SELECT BITS IN "SCR"
1432 014136 000207 RTS PC ;RETURN TO CALLING TEST
    
```

;THIS ROUTINE IS CALLED TO CONVERT EITHER THE "DN" NUMBER OR THE
 ;"LINE" NUMBER TO TWO ASCII CHARACTERS AND MOVE THEM INTO A
 ;PARTICULAR MESSAGE BUFFER FOR ERROR REPORTING

;CALLING SEQUENCE

```

;JSR RS,SUNUM ;CALL TO THIS ROUTINE
;ADDR1 ;ADDRESS OF THE NUMBER TO BE CONVERTED
;ADDR2 ;ADDRESS OF THE MSG BUFFER SLOT
    
```

SUNUM:

```

1440 014140 (2) 010046 MOV R0,-(SP) ;PUSH R0 ON STACK
1441 014146 (2) 010146 MOV R1,-(SP) ;PUSH R1 ON STACK
1442 014148 (2) 010246 MOV R2,-(SP) ;PUSH R2 ON STACK
1443 014146 012500 MOV (RS)+,R0 ;GET ADDRESS OF NUMBER
1444 014150 012501 MOV (RS)+,R1 ;GET MSG BUFFER ADDR
1445 014152 111000 MOV# (R0),R0 ;GET NO. TO BE CONVERTED
1446 014154 010002 MOV R0,R2 ;SAVE IT IN R2
1447 014156 006202 ASR R2 ;SHIFT MSD TO LSD POSITION
1448 014160 006202 ASR R2
1449 014162 006202 ASR R2
1450 014164 042702 177770 BIC #177770,R2 ;CLR JUNK BITS
1451 014170 062702 000060 ADD #60,R2 ;MAKE IT ASCII
1452 014174 110221 MOV# R2,(R1)+ ;PUT IT IN MSG BUFFER
1453 014176 042700 177770 BIC #177770,R0 ;CLR JUNK FROM LSD
1454 014202 062700 000060 ADD #60,R0 ;MAKE IT ASCII
1455 014206 110011 MOV# R0,(R1) ;PUT LSD IN THE BUFFER
1456 014210 012602 MOV (SP)+,R2 ;POP STACK INTO R2
1457 (2) 014212 012601 MOV (SP)+,R1 ;POP STACK INTO R1
1458 (2) 014214 012600 MOV (SP)+,R0 ;POP STACK INTO R0
1459 014216 000205 RTS RS ;RETURN TO CALLER
    
```

;THIS ROUTINE IS CALLED TO SET UP THE ERROR INFORMATION IN THE
 ;MESSAGE BUFFERS

```

SUER2: MOV R0,SREG0 ;STORE THE REGS IN CORE
SUER1: MOV R1,SREG1
        MOV R2,SREG2
        MOV R3,SREG3
    
```

```

1460 014220 010067 164734
1461 014224 010167 164732
1462 014230 010267 164730
1463 014234 010367 164726
    
```

1468	014240	010467	164724		MOV	R4, SREG4	
1469	014244	000207			RTS	PC	; RETURN TO REPORT ERROR
1470							
1471							; THIS ROUTINE IS USED TO ACCEPT INPUT PARAMETERS FROM THE CONSOLE
1472							; TELETYPE
1473							
1474	014246	104400			INPARA:	TYPE	
1475	014250	023164				VCWC	; "ASK FOR NO. WORDS BETWEEN VECTORS"
1476	014252	104407				RDOCT	; READ OCTAL NO. FM TTY
1477	014254	012600				MOV	(SP)+, R0
1478	014256	001407				BEG	2\$
1479	014260	022700	000004			CHP	R4, R0
1480	014264	001404				BEG	2\$
1481	014266	022700	000010			CHP	R10, R0
1482	014272	001404				BEG	3\$
1483	014274	000764				BR	INPARA
1484	014276	012700	000010	2\$:		MOV	R10, R0
1485	014302	000402				BR	4\$
1486	014304	012700	000020	3\$:		MOV	R20, R0
1487	014310	005067	003556	4\$:		CLR	DPFLG
1488	014314	005067	003564			CLR	RETFLG
1489	014320	000207				RTS	PC
1490							; RETURN TO CALLER
1491	014322	012767	177777	003300	INPARX:	MOV	R-1, VCFLG
1492	014330	000167	165306			JMP	BEGINA
1493	014334	012700	177777		INPARC:	MOV	R-1, R0
1494	014340	005067	003264			CLR	VCFLG
1495	014344	005067	003522			CLR	DPFLG
1496	014350	005067	003530			CLR	RETFLG
1497	014354	000167	165262			JMP	BEGINA
1498							; GO ASK FOR SELECT PARAMETER
1499	014360	104400			INPAR:	TYPE	; ASK FOR DEVICE ADDRESS
1500	014362	022473				INMSG1	
1501	014364	104407				RDOCT	; READ IN WHAT IS TYPED
1502	014366	012601				MOV	(SP)+, R1
1503	014370	001403				BEG	INPAR1
1504	014372	004767	000130			JSR	PC, CHKADR
1505	014376	000770				BR	INPAR
1506							; ERROR BRANCH
1507	014400	104400			INPAR1:	TYPE	; ASK FOR VECTOR ADDRESS
1508	014402	022537				INMSG2	
1509	014404	104407				RDOCT	; READ IN WHAT HE TYPES
1510	014406	012601				MOV	(SP)+, R1
1511	014410	001403				BEG	INPAR3
1512	014412	004767	000222			JSR	PC, CHKVCT
1513	014416	000770				BR	INPAR1
1514							; ERROR BRANCH
1515	014420	005767	003460		INPAR3:	TST	RETFLG
1516	014424	001402				BEG	1\$
1517	014426	000167	170220			JMP	ECHO2
1518	014432	005767	003434	1\$:		TST	DPFLG
1519	014436	001402				BEG	2\$
1520	014440	000167	171406			JMP	EXPAT2
1521	014444	104400		2\$:		TYPE	; GO BACK TO PATTERNS TESTS
							; ASK FOR DEVICE SELECTION PARAMETER

1552	014446	022606			INMSG3						
1553	014450	104407			RDOCT					: READ IN WHAT HE TYPES	
1554	014452	012601			MOV	(SP)+,R1				: GET THE SELECT PARAMETER	
1555	014454	001402			BEQ	INPAR4				: BR IF DEFAULT	
1556	014456	010167	002762		MOV	R1,DHSEL				: SET UP DH11 SELECTION PARAMETER	
1557	014458	012767	177777	002756	INPAR4: MOV	#-1,LINSEL				: INIT FOR ALL 16. LINES	
1558	014470	104400			TYPE					: ASK FOR LINE SELECT PARAMETER	
1559	014472	023122			INMSG4						
1560	014474	104407			RDOCT					: READ WHAT HE TYPES	
1561	014476	012601			MOV	(SP)+,R1				: GET IT OFF STACK	
1562	014500	001402			BEQ	15				: BR IF DEFAULT	
1563	014502	010167	002740		MOV	R1,LINSEL				: SET LINE SELECT PARAMETER	
1564	014506	005777	164424		15: TST	25R				: HALT AFTER SET UP ??	
1565	014512	100003			BPL	EXPAR				: BR IF NOT	
1566	014514	104400			TYPE					: TYPE CONTINUE MESSAGE PRIOR TO HALTING	
1567	014516	023052			INMSG7						
1568	014520	000000			HALT					: DEPRESS CONTINUE TO RESUME TESTING	
1569	014522	000167	165500		EXPAR: JMP	START2				: GO START UP THE PROGRAM	
1570	014526	020127	160020		CHKADR: CMP	R1,#160020				: IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT	
1571	014532	002001			BGE	15				: BR IF YES	
1572	014534	000436			BR	45				: BR IF NOT	
1573	014536	020127	160420		15: CMP	R1,#160420				: IS IT BELOW THE HIGH LIMIT?	
1574	014542	002401			BLT	25				: BR IF YES	
1575	014544	000432			BR	45				: BR IF NOT	
1576	014546	032701	000017		25: BIT	#17,R1				: CORRECT BOUNDARY ?	
1577	014552	001027			BNE	45				: BR IF NOT	
1578	014554	062716	000002		ADD	#2,(SP)				: MOVE RETURN PC AROUND ERROR BRANCH	
1579	014560	005767	003320		TST	RETFLG				: ARE WE IN ECHO TESTS ?	
1580	014564	001403			BEQ	215				: BR IF NOT	
1581	014566	010167	002644		MOV	R1,DHADR				: SET UP DH11 DEVICE ADDRESS	
1582	014572	000421			BR	55				: CONTINUE	
1583	014574	005767	003272		215: TST	DPFLG				: PATTERNS TESTS ACTIVE ??	
1584	014600	001403			BEQ	225				: BR IF NOT	
1585	014602	010167	002630		MOV	R1,DHADR				: SET UP DEVICE ADDRESS	
1586	014606	000413			BR	55				: CONTINUE	
1587	014610	012702	017530		225: MOV	#DHADTB,R2				: POINT TO BEGIN OF ADDR TABLE	
1588	014614	010122			35: MOV	R1,(R2)+				: SET UP A TABLE ENTRY	
1589	014616	062701	000020		ADD	#20,R1				: GENERATE NEXT DH11 ADDR	
1590	014622	022702	017570		CMP	#DHADTB+40,R2				: END OF TABLE ?	
1591	014626	001372			BNE	35				: BR IF NOT	
1592	014630	000402			BR	55				: RETURN TO INPUT ROUTINES	
1593	014632	104400			45: TYPE					: TELL HIM HE GOOFED	
1594	014634	022657			INMSG4						
1595	014636	000207			55: RTS	PC				: RETURN TO INPUT ROUTINES	
1596	014640	020127	000300		CHKVCT: CMP	R1,#300				: IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT	
1597	014644	002001			BGE	15				: BR IF YES	
1598	014646	000435			BR	45				: BR IF NOT	
1599	014650	020127	001000		15: CMP	R1,#1000				: IS IT BELOW THE HIGH LIMIT?	
1600	014654	002401			BLT	25				: BR IF YES	
1601	014656	000431			BR	45				: BR IF NOT	
1602	014660	032701	000007		25: BIT	#7,R1				: CORRECT BOUNDARY ?	

```

1576 014664 001026          BNE      45          ;BR IF NOT
1577 014666 062716 000002      ADD      #2,(SP)    ;MOVE RETURN PC AROUND ERROR BRANCH
1578 014672 005767 003206      TST     RETFLG     ;ARE WE IN ECHO TESTS ?
1579 014676 001403          BEQ      215        ;BR IF NOT
1580 014700 010167 002534      MOV     R1,DHVCT   ;SET UP DH11 VECTOR ADDR
1581 014704 000420          BR       55        ;CONTINUE
1582 014706 005767 003160      215:   TST     DPFLG   ;PATTERNS TESTS ACTIVE ??
1583 014712 001403          BEQ      225        ;BR IF NOT
1584 014714 010167 002520      MOV     R1,DHVCT   ;SET UP DEVICE VECTOR
1585 014720 000412          BR       55        ;CONTINUE
1586 014722 012702 017570      225:   MOV     #DHVCTB,R2 ;POINT TO BEGIN OF VECTOR TABLE
1587 014726 010122      35:   MOV     R1,(R2)+  ;SET UP A TABLE ENTRY
1588 014730 060001          ADD     R0,R1      ;GENERATE NEXT DH11 ADDR
1589 014732 022702 017630      CMP     #DHVCTB+40,R2 ;END OF TABLE ?
1590 014736 001373          BNE     35         ;BR IF NOT
1591 014740 000402          BR      55        ;RETURN TO INPUT ROUTINES
1592 014742 104400      45:   TYPE                    ;TELL HIM HE GOOFED
1593 014744 022730          INMSG5
1594 014746 000207      55:   RTS      PC          ;RETURN TO INPUT ROUTINES
1595
1596          ;THESE TWO ROUTINES SERVICE UNEXPECTED BUS ERROR AND RSVD INSTR TRAPS
1597
1598 014750 010667 164220      BUSER:  MOV     SP,SREG6 ;SAVE THE SP
1599 014754 012667 164202      MOV     (SP)+,SREG1 ;GET THE TRAP PC
1600 014760 012667 164200      MOV     (SP)+,SREG2 ;GET THE TRAP PSM
1601 014764 012706 001100      MOV     #STACK,SP  ;RESET THE STACK POINTER
1602 014770 012767 015000 164112  MOV     #15,SLPERR ;ALWAYS COME BACK TO 15
1603 014776 104014          ERROR   14         ;UNEXPECTED BUS ERROR TRAP
1604 015000 000005      15:   RESET                    ;PREPARE TO RESTART
1605 015002 004767 002334      JSR     PC,CHPS1     ;GO CLEAR PSM
1606 015006 000167 176706      JMP     CKRST2       ;GO RESTART THE PROGRAM
1607
1608 015012 010667 164156      RESERR: MOV     SP,SREG6 ;SAVE THE SP
1609 015016 012667 164140      MOV     (SP)+,SREG1 ;GET THE TRAP PC
1610 015022 012667 164136      MOV     (SP)+,SREG2 ;GET THE TRAP PSM
1611 015026 012706 001100      MOV     #STACK,SP  ;RESET THE STACK POINTER
1612 015032 012767 015042 164050  MOV     #15,SLPERR ;ALWAYS COME BACK TO 15
1613 015040 104015          ERROR   15         ;UNEXPECTED RSVD INSTR ERROR TRAP
1614 015042 000005      15:   RESET                    ;PREPARE TO RESTART
1615 015044 004767 002272      JSR     PC,CHPS1     ;GO CLEAR PSM
1616 015050 000167 176644      JMP     CKRST2       ;GO RESTART THE PROGRAM
1617
1618          ;THIS ROUTINE IS CALLED WHEN A TEST NEEDS TO RESTORE THE TRAP
1619          ;CATCHER IN THE DH11 VECTOR
1620
1621 015054 016703 002360      RESTRP: MOV     DHVCT,R3 ;GET VECTOR ADDRESS
1622 015060 010313          MOV     R3,(R3)    ;RESTORE THE TRAP CATCHER
1623 015062 062723 000002      ADD     #2,(R3)+
1624 015066 005023          CLR     (R3)+
1625 015070 010313          MOV     R3,(R3)
1626 015072 062723 000002      ADD     #2,(R3)+
1627 015076 005023          CLR     (R3)+
1628 015100 000207          RTS      PC          ;RETURN TO CALLING TEST
1629

```



```

1684 015226 005767 002224 SETCL: TST QUICKX ;EXIT AFTER ONLY ONE CHAR LNGTH ?
1685 015230 001034 BNE 25 ;BR IF YES
1686 015234 005267 002260 INC CLSEL ;GENERATE NEW CHAR LNGTH SELECT CODE
1687 015240 022767 000004 002252 CMP #4,CLSEL ;DONE FOUR OF THEM ??
1688 015246 001426 BEQ 25 ;BR IF YES
1689 015250 005767 002200 TST QUICK ;QUICK TEST FLAG SET ?
1690 015254 001407 BEQ 15 ;BR IF NOT
1691 015258 005267 002174 INC QUICKX ;SET QUICK TEST EXIT FLAG
1692 015262 005067 002232 CLR CLSEL ;DO ONLY 5 BIT CHARS
1693 015266 012767 177760 002222 MOV #177760,CHRCNT ;DO ONLY 32 CHAR BUFFER
1694 015274 042767 000003 002210 15: BIC #3,CURLPR ;SET UP THE CURRENT LPR
1695 015302 056767 002212 BIS CLSEL,CURLPR
1696 015310 006367 002202 ASL CHRCNT ;GENERATE CHAR COUNT
1697 015314 004767 000006 JSR PC,SUBUF1 ;GO SET UP THE OUTPUT BUFFER
1698 015320 062705 000002 ADD #2,R5 ;MOVE PC AROUND EXIT BRANCH
1699 015324 000205 25: RTS R5 ;RETURN

```

;THIS ROUTINE IS CALLED TO LOAD THE OUTPUT DATA BUFFER WITH THE
;REQUIRED BINARY COUNT PATTERN

;CALLING SEQUENCE:

; JSR PC,SUBUF1 ;CALL

SUBUF1:

```

1700
1701
1702
1703
1704
1705
1706
1707
1708 015326
(2) 015326 010246 MOV R2,-(SP) ;: PUSH R2 ON STACK
(2) 015330 010346 MOV R3,-(SP) ;: PUSH R3 ON STACK
(2) 015332 010446 MOV R4,-(SP) ;: PUSH R4 ON STACK
1709 015334 005004 CLR R4 ;: INIT CHAR GENERATOR
1710 015336 016703 002154 MOV CHRCNT,R3 ;: SET UP LOAD COUNT
1711 015342 012702 030212 MOV #TBUF,R2 ;: SET UP BUFFER POINTER
1712 015346 110422 15: MOVB R4,(R2)+ ;: LOAD A CHAR
1713 015350 005204 INC R4 ;: GENERATE NEXT CHAR
1714 015352 005203 INC R3 ;: COUNT ONE LOADED
1715 015354 001374 BNE 15 ;: BR TIL BUFFER FULL
1716 015356 012604 MOV (SP)+,R4 ;: POP STACK INTO R4
(2) 015360 012603 MOV (SP)+,R3 ;: POP STACK INTO R3
(2) 015362 012602 MOV (SP)+,R2 ;: POP STACK INTO R2
1717 015364 000207 RTS PC ;: RETURN

```

;THIS ROUTINE IS CALLED TO SET UP THE PARITY SELECT BITS
;IN THE CURRENT LPR TEST CONSTANT

;CALLING SEQUENCE:

; JSR R5,SETPAR ;CALL
; BR NEWCL ;EXIT BRANCH

```

1727 015366 022767 177777 002126 SETPAR: CMP #-1,PARBIT ;: DONE ALL PARITY COMBOS ?
1728 015374 001444 BEQ 55 ;: BR IF YES
1729 015376 005767 002052 TST QUICK ;: QUICK TEST FLAG SET ?
1730 015402 001403 BEQ 15 ;: BR IF NOT
1731 015404 012767 000060 002110 MOV #60,PARBIT ;: CHECK ODD PARITY ONLY
1732 015412 042767 000060 002072 15: BIC #60,CURLPR ;: SET PARITY SELECT BITS

```

```

1733 015420 056767 002076 002064      BIS      PARBIT,CURLPR
1734 015426 005767 002070      TST      PARBIT      ;SELECT BITS 00 ?
1735 015432 001004      BNE      2$          ;BR IF NOT
1736 015434 012767 000020 002060      MOV      #20,PARBIT ;SET SELECT BITS TO 01
1737 015442 000417      BR       4$          ;EXIT
1738 015444 022767 000020 002050 2$:  CMP      #20,PARBIT ;SELECT BITS 10 ?
1739 015452 001004      BNE      3$          ;BR IF NOT
1740 015454 012767 000060 002040      MOV      #60,PARBIT ;MAKE SELECT BITS 11
1741 015462 000407      BR       4$          ;EXIT
1742 015464 022767 000060 002030 3$:  CMP      #60,PARBIT ;SELECT BITS 11 ?
1743 015472 001005      BNE      5$          ;BR IF NOT
1744 015474 012767 177777 002020      MOV      #-1,PARBIT ;SET EXIT FLAG
1745 015502 062705 000002      4$:  ADD      #2,RS      ;MOVE RETURN PC AROUND EXIT BRANCH
1746 015506 000205      5$:  RTS       RS        ;RETURN
1747
1748      ;THIS ROUTINE IS CALLED TO SET UP FOR KEYBOARD INTERRUPTS
1749
1750 015510 012767 015534 162342 KYBD1:  MOV      #KYBD2,60      ;SET UP THE INPUT VECTOR
1751 015516 012767 000340 162336      MOV      #340,62
1752 015524 012767 000100 162026      MOV      #100,177560 ;ENABLE KYBD INTR
1753 015532 000207      RTS       PC          ;RETURN TO START TESTING
1754
1755      ;THIS ROUTINE SERVICES THE KEYBOARD INTERRUPT AND LOOKS FOR AN "S"
1756      ;BEING TYPED TO INDICATE ABORT AND PRINT STATISTICS
1757
1758 015534 122767 000323 162020 KYBD2:  CMPB     #323,177562 ;WAS AN "S" TYPED ?
1759 015542 001401      BEQ      1$          ;BR IF YES
1760 015544 000002      RTI      ;RETURN AND FORGET IT
1761 015546 000005      1$:  RESET   ;ZAP THE WORLD
1762 015550 012706 001100      MOV      #STACK,SP ;RESET THE SP
1763 015554 004767 001562      JSR      PC,CHPS1  ;GO CLEAR PSM
1764 015560 000167 166540      JMP      PRSTAT    ;GO DUMP THE STATISTICS
1765
1766      ;THIS ROUTINE SENDS A TEST BUFFER TO REMOTE DH11 LINE
1767
1768 015564 016701 001646      SENDP2: MOV      DHADR,R1      ;SET UP DH SCR ADDR
1769 015570 012711 004000      MOV      #BIT11,(R1) ;CLEAR THE DH11
1770 015574 016711 002076      MOV      LINE,(R1)   ;SET LINE SELECT
1771 015600 162705 030212      SUB      #TBUF,RS    ;SET UP BYTE COUNT
1772 015604 005405      NEG      RS
1773 015606 010561 000010      MOV      RS,BCR(R1)
1774 015612 012761 030212 000006      MOV      #TBUF,CAR(R1) ;SET CURRENT ADDRESS
1775 015620 016761 001666 000004      MOV      CURLPR,LPR(R1) ;SET LINE PARAMETERS
1776 015626 016761 001616 000012      MOV      LINMSK,BAR(R1) ;ACTIVATE THE LINE
1777
1778 015634 005711      1$:  TST      (R1)        ;DONE TRANSMITTING ??
1779 015636 100376      BPL      1$          ;BR IF NOT
1780 015640 000207      RTS       PC          ;RETURN TO CONTROL ROUTINE "SENDP1"
1781
1782      ;THIS ROUTINE IS CALLED TO LOAD FILLERS INTO ECHO BUFFER
1783
1784 015642 116704 002316      LDFILL: MOVB     FILLB,R4      ;GET COUNT OF FILLERS
1785 015646 012703 020121      MOV      #ECBUF+1,R3 ;SET UP BUFFER POINTER
1786 015652 116767 163322 002240      MOVB     #TMO,ECBUF  ;STORE LF CHAR

```

```

1787 015660 116722 163314          MOVB    $TMP0,(R2)+      ;IN ECHO BUFFER TOO
1788 015664 116723 002272          MOVB    FILLA,(R3)+      ;LOAD A FILLER CHAR
1789 015670 116722 002266          MOVB    FILLA,(R2)+
1790 015674 005304          DEC     R4              ;COUNT IT
1791 015676 001372          BNE    1$              ;BR TIL REQUIRED COUNT LOADED
1792 015700 116704 002260          MOVB    FILLB,R4        ;SET UP BYTE COUNT REG
1793 015704 005204          INC     R4
1794 015706 005404          NEG     R4
1795 015710 010461 000010          MOV     R4,BCR(R1)      ;LOAD BCR REG
1796 015714 000207          RTS     PC              ;RETURN TO RINT2
    
```

;THIS ROUTINE IS CALLED TO SET UP XMITTER SPEED

```

1797
1798
1799
1800 015716 104400          INXSP:  TYPE            ;ASK USER TO TYPE SPEED
1801 015720 024046          XMSG1    "TRANSMITTER SPEED ?"
1802 015722 012767 017702 001750 1$:  MOV     #XSPTAB,XSPTR    ;SET UP TABLE POINTER
1803 015730 042767 036000 001554          BIC     #36000,CURLPR   ;INIT SPEED SELECT BITS
1804 015736 104410          RODEC                    ;READ SPEED HE TYPED
1805 015740 005716          TST     (SP)            ;DEFAULT TO 9600. BAUD ?
1806 015742 001426          BEQ     4$              ;BR IF YES
1807 015744 027716 001730          CMP     2XSPTR,(SP)     ;TYPED ENTRY MATCH TABLE ENTRY ?
1808 015750 001010          BNE    3$              ;BR IF NOT
1809 015752 062767 000002 001720          ADD     #2,XSPTR        ;POINT TO SELECT BITS IN TABLE
1810 015760 057767 001714 001524          BIS     2XSPTR,CURLPR   ;SET SPEED SELECT BITS
1811 015766 005726          TST     (SP)+          ;FIX STACK
1812 015770 000417          BR     5$              ;CONTINUE
1813
1814 015772 062767 000004 001700 3$:  ADD     #4,XSPTR        ;POINT TO NEXT ENTRY
1815 016000 022767 017766 001672          CMP     #XSPTAB+52.,XSPTR;END OF TABLE ??
1816 016006 001356          BNE    2$              ;BR IF NOT
1817 016010 104400          TYPE            ;ERROR MESSAGE
1818 016012 024074          XMSG2    "INVALID XMITR SPEED - TRY AGAIN"
1819 016014 005726          TST     (SP)+          ;FIX THE SP
1820 016016 000741          BR     1$              ;GO TRY AGAIN
1821
1822 016020 052767 032000 001464 4$:  BIS     #32000,CURLPR   ;SET UP DEFAULT TO 9600. BAUD
1823 016026 005726          TST     (SP)+          ;FIX STACK POINTER
1824
1825 016030 000207          5$:  RTS     PC              ;RETURN TO CALLER
1826
1827
    
```

;THIS ROUTINE IS CALLED TO SET UP RECEIVER SPEED

```

1828
1829 016032 104400          INRSP:  TYPE            ;ASK USER TO TYPE SPEED
1830 016034 024137          RMSG1    "RECEIVER SPEED ?"
1831 016036 012767 017770 001722 1$:  MOV     #RSPTAB,RSPTR   ;SET UP TABLE POINTER
1832 016044 042767 001700 001440          BIC     #1700,CURLPR   ;INIT SPEED SELECT BITS
1833 016052 104410          RODEC                    ;READ SPEED HE TYPED
1834 016054 005716          TST     (SP)            ;DEFAULT TO 9600. BAUD ?
1835 016056 001426          BEQ     4$              ;BR IF YES
1836 016060 027716 001702          CMP     2RSPTR,(SP)    ;TYPED ENTRY MATCH TABLE ENTRY ?
1837 016064 001010          BNE    3$              ;BR IF NOT
1838 016066 062767 000002 001672          ADD     #2,RSPTR        ;POINT TO SELECT BITS IN TABLE
1839 016074 057767 001666 001410          BIS     2RSPTR,CURLPR   ;SET SPEED SELECT BITS
1840 016102 005726          TST     (SP)+          ;FIX STACK
    
```

```

1841 016104 000417 BR 5$ ;CONTINUE
1842
1843 016106 062767 000004 001652 3$: ADD #4,RSPTR ;POINT TO NEXT ENTRY
1844 016114 022767 020054 001644 CMP #RSPTAB+52.,RSPTR ;END OF TABLE ??
1845 016122 001356 BNE 2$ ;BR IF NOT
1846 016124 104400 TYPE ;ERROR MESSAGE
1847 016126 024162 RSMMSG2 ;"INVALID RCVR SPEED - TRY AGAIN"
1848 016130 005726 TST (SP)+ ;FIX THE SP
1849 016132 000741 BR 1$ ;GO TRY AGAIN
1850
1851 016134 052767 001500 001350 4$: BIS #1500,CURLPR ;SET UP DEFAULT TO 9600. BAUD
1852 016142 005726 TST (SP)+ ;FIX STACK POINTER
1853
1854 016144 000207 5$: RTS PC ;RETURN TO CALLER
1855
1856
1857 ;THIS ROUTINE IS CALLED TO SET UP LINE PARAMETERS FM KYBD
1858
1859 016146 105067 005306 LPRIN: CLRB EC2 ;CLEAR ECHO BUFFER
1860 016152 104400 TYPE ;
1861 016154 023774 LPMMSG ;"DO YOU WANT TO CHANGE "LPR"?"
1862 016156 104405 1$: ROCHR ;
1863 016160 012600 MOV (SP)+,RO ;GET WHAT HE TYPED
1864 016162 122700 000015 CMPB #15,RO ;WAS IT A <CR> ??
1865 016166 001405 BEQ 2$ ;BR IF YES
1866 016170 110067 005264 MOVB RO,EC2 ;ECHO WHAT HE TYPED
1867 016174 104400 TYPE ;
1868 016176 023460 EC2 ;
1869 016200 000766 BR 1$ ;GO WAIT FOR TERMINATOR
1870
1871 016202 105767 005252 2$: TSTB EC2 ;<CR> ONLY ??
1872 016206 001411 BEQ 3$ ;BR IF YES
1873 016210 122767 000116 005242 CMPB #116,EC2 ;WAS IT A "NO" ??
1874 016216 001405 BEQ 3$ ;BR IF IT WAS
1875 016220 122767 000131 005232 CMPB #131,EC2 ;WAS IT A "YES" ??
1876 016226 001347 BNE LPRIN ;GO ASK ALL OVER AGAIN
1877 016230 000407 BR 4$ ;BR IF IT WAS "YES"
1878 016232 005767 001254 3$: TST CURLPR ;HAS LPR BEEN SET UP AT ALL ?
1879 016236 001016 BNE 5$ ;BR IF YES USE PREVIOUS LPR
1880 016240 012767 033503 001244 MOV #33503,CURLPR ;SET DEFAULT 9600 BAUD,8 BITS NO PARITY
1881 016246 000412 BR 5$ ;CONTINUE
1882 016250 004767 177442 4$: JSR PC,INXSP ;GO INPUT AND SET UP XMIT SPEED
1883 016254 004767 177552 JSR PC,INRSP ;GO INPUT AND SET UP RCVR SPEED
1884 016260 004767 000022 JSR PC,INCL ;GO INPUT AND SET UP CHAR LENGTH
1885 016264 004767 000162 JSR PC,INSB ;GO INPUT AND SET UP NO. OF STOP BITS
1886 016270 004767 000274 JSR PC,INPB ;GO INPUT AND SET UP PARITY SELECTION
1887 016274 004767 000410 5$: JSR PC,INFCHR ;GO INPUT AND SET UP FILLER CHAR
1888 016300 004767 000474 JSR PC,INFCNT ;GO INPUT AND SET UP FILLER COUNT
1889 016304 000207 RTS ;RETURN TO CALLER
1890
1891 ;THIS ROUTINE IS CALLED TO SET UP CHAR LENGTH BITS
1892
1893 016306 105067 005146 INCL: CLRB EC2 ;CLEAR THE ECHO BUFFER
1894 016312 104400 TYPE ;ASK FOR INPUT

```

```

1895 016314 024225          CLMSG1          ;"CHAR LENGTH - 6,7, OR 8 ?"
1896 016316 042767 000003 001166 1S:  BIC          #3,CURLPR ;INIT CHAR LENGTH SELECT BITR
1897 016324 104405          RDCHR          ;GET THE CHAR HE TYPED
1898 016326 012600          MOV          (SP)+,RO ;GET WHAT HE TYPED
1899 016330 122700 000015          CMPB         #15,RO  ;WAS IT A <CR> ??
1900 016334 001405          BEQ          11S   ;BR IF IT WAS
1901 016336 110067 005116          MOVB         RO,EC2 ;ECHO WHAT HE TYPED
1902 016342 104400          TYPE
1903 016344 023460          EC2
1904 016346 000763          BR          1S     ;GO WAIT FOR TERMINATOR
1905 016350 105767 005104          11S:  TSTB         EC2   ;<CR> ONLY ??
1906 016354 001432          BEQ          4S     ;BR IF YES
1907 016356 142767 000060 005074          BICB         #60,EC2 ;STRIP ASCII
1908 016364 122767 000006 005066          CMPB         #6,EC2  ;6 BITS ?
1909 016372 001004          BNE         2S     ;BR IF NOT
1910 016374 052767 000001 001110          BIS          #1,CURLPR ;SET UP FOR 6 BIT CHARS
1911 016402 000422          BR          5S     ;CONTINUE
1912 016404 122767 000007 005046 2S:  CMPB         #7,EC2  ;7 BITS ?
1913 016412 001004          BNE         3S     ;BR IF NOT
1914 016414 052767 000002 001070          BIS          #2,CURLPR ;SET UP FOR 7 BIT CHARS
1915 016422 000412          BR          5S     ;CONTINUE
1916 016424 122767 000010 005026 3S:  CMPB         #8,EC2  ;8 BITS ?
1917 016432 001403          BEQ          4S     ;BR IF YES
1918 016434 104400          TYPE          ;ERROR MESSAGE
1919 016436 024263          CLMSG2          ;"INVALID CHAR LENGTH = TRY AGAIN"
1920 016440 000722          BR          INCL   ;GO TRY AGAIN
1921 016442 052767 000003 001042 4S:  BIS          #3,CURLPR ;SET UP FOR 8 BIT CHARS
1922 016450 000207 5S:  RTS          PC    ;RETURN TO CALLER
1923
1924          ;THIS ROUTINE IS CALLED TO SET UP NO. OF STOP BITS
1925
1926 016452 105067 005002  INSB:  CLRB         EC2   ;CLEAR ECHO BUFFER
1927 016456 104400          TYPE          ;ASK FOR INPUT
1928 016460 024327          SBMSG1
1929 016462 104405          1S:  RDCHR          ;"NO. OF STOP BITS - 1 OR 2 ?"
1930 016464 012600          MOV          (SP)+,RO ;GET CHAR TYPED
1931 016466 122700 000015          CMPB         #15,RO  ;GET WHAT HE TYPED
1932 016472 001405          BEQ          11S   ;WAS IT A <CR>
1933 016474 110067 004760          MOVB         RO,EC2 ;BR IF YES
1934 016500 104400          TYPE          ;ECHO WHAT HE TYPED
1935 016502 023460          EC2
1936 016504 000766          BR          1S     ;GO WAIT FOR TERMINATOR
1937 016506 105767 004746          11S:  TSTB         EC2   ;<CR> ONLY ??
1938 016512 001422          BEQ          3S     ;BR IF YES
1939 016514 142767 000060 004736          BICB         #60,EC2 ;CLEAR ASCII JUNK
1940 016522 122767 000002 004730          CMPB         #2,EC2  ;2 STOP BITS ?
1941 016530 001004          BNE         2S     ;BR IF NOT
1942 016532 052767 000004 000752          BIS          #4,CURLPR ;SET UP FOR TWO STOP BITS
1943 016540 000412          BR          4S     ;CONTINUE
1944 016542 122767 000001 004710 2S:  CMPB         #1,EC2  ;ONE STOP BIT ?
1945 016550 001403          BEQ          3S     ;BR IF YES
1946 016552 104400          TYPE          ;ERROR MESSAGE
1947 016554 024366          SBMSG2          ;"INVALID NO. STOP BITS - TRY AGAIN"
1948 016556 000735          BR          INSB   ;GO TRY AGAIN

```

```

1949 016560 042767 000004 000724 3S: BIC #4,CURLPR ;SET UP FOR ONE STOP BIT
1950 016566 000207 4S: RTS PC ;RETURN TO CALLER
1951
1952 ;THIS ROUTINE IS CALLED TO SET UP PARITY SELECT BITS
1953
1954 016570 105067 004664 INPB: CLRB EC2 ;CLEAR ECHO BUFFER
1955 016574 104400 TYPE ;ASK FOR INPUT
1956 016576 024434 PMSG1 ;"PARITY - E,O, OR <CR> ?"
1957 016500 042767 000060 000704 1S: BIC #60,CURLPR ;INIT FOR NO PARITY CHECKING
1958 016606 104405 RDCHR ;GET CHAR TYPED
1959 016610 012600 MOV (SP)+,R0 ;GET WHAT HE TYPED
1960 016612 122700 000015 CMPB #15,R0 ;WAS IT A <CR> ??
1961 016616 001405 BEQ 11S ;BR IF IT WAS
1962 016620 110067 004634 MOVB R0,EC2 ;ECHO THE CHAR TYPED
1963 016624 104400 TYPE
1964 016626 023460 EC2
1965 016630 000763 BR 1S ;GO WAIT FOR TERMINATOR
1966 016632 105767 004622 11S: TSTB EC2 ;<CR> ONLY ??
1967 016636 001423 BEQ 4S ;BR IF YES
1968 016640 122767 000105 004612 CMPB #105,EC2 ;EVEN PARITY ??
1969 016646 001004 BNE 2S ;BR IF NOT
1970 016650 052767 000060 000634 BIS #60,CURLPR ;SET UP FOR EVEN PARITY
1971 016656 000413 BR 4S ;CONTINUE
1972 016660 122767 000117 004572 2S: CMPB #117,EC2 ;ODD PARITY
1973 016666 001004 BNE 3S ;BR IF NOT
1974 016670 052767 000020 000614 BIS #20,CURLPR ;SET UP FOR ODD PARITY
1975 016676 000403 BR 4S ;CONTINUE
1976 016700 104400 3S: TYPE ;ERROR MESSAGE
1977 016702 024502 PMSG2 ;INVALID PARITY - TRY AGAIN"
1978 016704 000731 BR INPB ;GO TRY AGAIN
1979 016706 000207 4S: RTS PC ;RETURN TO CALLER
1980
1981 ;THIS ROUTINE IS CALLED TO SET UP "FILL" CHAR
1982
1983 016710 105067 004544 INFCHR: CLRB EC2 ;CLEAR ECHO BUFFER
1984 016714 005067 001242 CLR FILLA ;INIT TEMP STORAGE FOR CHAR
1985 016720 104400 TYPE ;GO ASK FOR FILLER CHAR
1986 016722 024541 FILC1 ;"FILL CHAR ?"
1987 016724 005067 001230 1S: CLR DHFILL ;ININ FILL LOCATION
1988 016730 104405 RDCHR ;GET CHAR TYPED
1989 016732 012600 MOV (SP)+,R0 ;GET WHAT HE TYPED
1990 016734 122700 000015 CMPB #15,R0 ;WAS IT A <CR> ??
1991 016740 001405 BEQ 2S ;BR IF YES
1992 016742 110067 004512 MOVB R0,EC2 ;ECHO WHAT HE TYPED
1993 016746 104400 TYPE
1994 016750 023460 EC2
1995 016752 000764 BR 1S ;GO WAIT FOR TERMINATOR
1996
1997 016754 105767 004500 2S: TSTB EC2 ;<CR> ONLY ??
1998 016760 001403 BEQ 3S ;BR IF YES
1999 016762 116767 004472 001171 MOVB EC2,DHFILL+1 ;SET UP FILL CHAR
2000 016770 116767 001165 001164 3S: MOVB DHFILL+1,FILLA ;SAVE FILL CHAR
2001 016776 000207 RTS PC ;RETURN TO CALLER
2002
    
```

```

2003                                     ;THIS ROUTINE IS CALLED TO SET UP "FILL" COUNT
2004
2005 017000 005067 001160      INFCNT: CLR      FILLB          ;INIT TEMP STORAGE FOR COUNT
2006 017004 104400              TYPE          ;ASK FOR COUNT
2007 017006 024567              FILC2         ;"FILL COUNT ?"
2008 017010 104407              RDOCT         ;GET OCTAL NO. TYPED
2009 017012 005716              TST      (SP)      ;DEFAULT TO ONE ?
2010 017014 001403              BEQ      1$        ;BR IF YES
2011 017016 111667 001136      MOV      (SP),DHFILL ;SET UP COUNT TYPED
2012 017022 000403              BR      2$        ;CONTINUE
2013 017024 112767 000001 001126 1$: MOV      #1,DHFILL ;SET UP FOR 1 FILLER
2014 017032 005726              TST      (SP)+    ;FIX THE SP
2015 017034 142767 000360 001116 BIC      #360,DHFILL ;LIMIT COUNT TO 15. MAX
2016 017042 116767 001112 001114 MOV      DHFILL,FILLB ;SAVE IT FOR LATER
2017 017050 000207              RTS      PC       ;RETURN TO CALLER
2018                                     ;THIS ROUTINE CALLED TO SET UP ALTERNATING 1/0 PATTERN
2019
2020 017052 004767 000246      SUPATA: JSR     PC,CLALL ;GO CLEAR XMIT AND RCV BUFFERS
2021 017056 016700 000434      MOV      CHRCNT,R0    ;GET CHAR COUNT
2022 017062 012705 030212      MOV      #TBUF,R5    ;POINT TO XMIT BUFFER
2023 017066 112725 000252      1$:  MOV      #252,(R5)+ ;LOAD A BYTE
2024 017072 005200              INC      R0          ;COUNT IT
2025 017074 001374              BNE     1$          ;BR TILL BUFFER FULL
2026 017076 000207              RTS      PC         ;RETURN TO "DPATA" ROUTINE
2027
2028                                     ;THIS ROUTINE IS CALLED TO SET UP UP COUNT PATTERN
2029
2030 017100 004767 000220      SUPATU: JSR     PC,CLALL ;GO CLEAR BUFFERS
2031 017104 016700 000406      MOV      CHRCNT,R0    ;GET COUNT OF CHARS TO LOAD
2032 017110 012705 030212      MOV      #TBUF,R5    ;POINT TO XMITTR BUFFER
2033 017114 005004              CLR      R4          ;INIT CHAR GENERATOR
2034 017116 110425              1$:  MOV      R4,(R5)+ ;LOAD ONE BYTE
2035 017120 105204              INCB    R4          ;GENERATE NEXT BYTE
2036 017122 005200              INC      R0          ;COUNT IT
2037 017124 001374              BNE     1$          ;BR TIL BUFFER FULL
2038 017126 000207              RTS      PC         ;RETURN TO "DPATU" ROUTINE
2039
2040                                     ;THIS ROUTINE IS CALLED TO SET UP DOWN COUNT PATTERN
2041
2042 017130 004767 000170      SUPATD: JSR     PC,CLALL ;CLEAR THE BUFFERS
2043 017134 016700 000356      MOV      CHRCNT,R0    ;SET UP COUNT TO LOAD
2044 017140 012705 030212      MOV      #TBUF,R5    ;POINT TO XMIT BUFFER
2045 017144 012704 000377      MOV      #377,R4     ;INIT CHAR GENERATOR
2046 017150 110425              1$:  MOV      R4,(R5)+ ;LOAD ONE BYTE
2047 017152 105304              DECB    R4          ;GENERATE NEW CHAR
2048 017154 005200              INC      R0          ;COUNT IT
2049 017156 001374              BNE     1$          ;BR TIL BUFFER FULL
2050 017160 000207              RTS      PC         ;RETURN TO "DPATA" ROUTINE
2051
2052                                     ;THIS ROUTINE CALLED TO LOAD RANDOM DATA PATTERN
2053
2054 017162 004767 000136      SUPATR: JSR     PC,CLALL ;GO CLEAR BUFFERS
2055 017166 016700 000324      MOV      CHRCNT,R0    ;SET UP COUNT TO LOAD
2056 017172 012705 030212      MOV      #TBUF,R5    ;POINT TO XMITTR BUFFER

```

```

2057 017176 012767 125252 000704      MOV      #125252,RANA      ;INIT RANDOM NUMBER GENERATOR
2058
2059 017204 066767 000700 000700 1S:    ADD      RANA,RANB      ;GENERATE RANDOM NO.
2060 017212 005567 000672                ADC      RANA
2061 017216 066767 000670 000664      ADD      RANB,RANA
2062 017224 005567 000662                ADC      RANB
2063
2064 017230 116725 000654                MOVB     RANA,(RS)+      ;LOAD A BYTE
2065 017234 005200                INC      RO              ;COUNT IT
2066 017236 001362                BNE     1$              ;BR TIL BUFFER FULL
2067 017240 000207                RTS      PC              ;RETURN TO "DPATR" ROUTINE
2068
2069                ;THIS ROUTINE LOADS A SINGLE CHAR THROUGHOUT BUFFER
2070
2071 017242 004767 000056      SUPATS: JSR      PC,CLALL  ;GO CLEAR BUFFERS
2072 017246 016700 000244                MOV      CHCNT,RO       ;INIT CHAR COUNTER
2073 017252 012705 030212                MOV      #TBUF,RS       ;POINT TO XMIT BUFFER
2074 017256 116725 000620 1S:    MOVB     SINGLE,(RS)+   ;LOAD ONE CHAR
2075 017262 005200                INC      RO              ;COUNT IT
2076 017264 001374                BNE     1$              ;BR TIL BUFFER FULL
2077 017266 000207                RTS      PC              ;RETURN TO "DPATS" ROUTINE
2078
2079                ;THIS ROUTINE CALLED TO INIT CHAR LENGTH MASK FOR PATTERNS TESTS
2080
2081 017270 016700 000216      SUCLMK: MOV      CURLPR,RO  ;GET CURRENT "LPR"
2082 017274 012767 000340 000612      MOV      #340,CLMSK     ;INIT FOR 5 BIT CHARS
2083 017302 042700 177774                BIC      #177774,RO     ;MASK OFF ALL BUT CL BITS
2084 017306 005700 1S:    TST      RO              ;DONE SETUP ?
2085 017310 001404                BEQ     2$              ;BR IF YES
2086 017312 106367 000576                ASLB     CLMSK          ;SHIFT MASK LEFT
2087 017316 005300                DEC      RO              ;COUNT IT
2088 017320 000772                BR      1$              ;GO SEE IF ITS RIGHT ON
2089 017322 000207 2S:    RTS      PC              ;RETURN TO CALLER
2090                ;ROUTINE TO CLEAR XMIT AND RECEIVER BUFFERS
2091
2092 017324 012700 030212      CLALL:  MOV      #TBUF,RO   ;SET UP POINTER
2093 017330 005020 1S:    CLR      (RO)+          ;CLEAR A WORD
2094 017332 022700 031342                CMP      #ENBUFS,RO     ;DONE ALL LOCATIONS ?
2095 017336 001374                BNE     1$              ;BR IF NOT
2096 017340 000207                RTS      PC

```

```

2098      ;THIS ROUTINE IS CALLED TO SET PSM PRIORITY TO 000 IN ORDER
2099      ;TO BE LSI11 COMPATIBLE
2100
2101      017343 012746 000000      CHPS1:  MOV    80,-(SP)      ;NEW PSM
2102      017344 012746 017354      MOV    81$,-(SP)      ;NEW PC
2103      017352 000002      RTI                    ;CHANGE PSM
2104      017354 000207      1$:    RTS     PC      ;RETURN TO CALLING TEST
2105
2106      ;THIS ROUTINE DOES THE SAME THING EXCEPT IT SET THE PSM
2107      ;PRIORITY TO 340 (LEVEL 7 ) TO LOCK OUT INTR$
2108
2109      017356 012746 000340      CHPS2:  MOV    8340,-(SP)  ;NEW PSM
2110      017362 012746 017370      MOV    81$,-(SP)  ;NEW PC
2111      017366 000002      RTI                    ;CHANGE THE PSM
2112      017370 000207      1$:    RTS     PC      ;RETURN TO CALLING TEST
2113
2114      ;THIS ROUTINE IS ALSO FOR LSI11 COMPATIBILITY AND IT IS CALLED
2115      ;TO SAVE THE PSM IN "STMPD"
2116
2117      017372 005046      SAPS:   CLR    -(SP)      ;TEMP STORAGE TO SAVE PSM
2118      017374 016746 160434      MOV    34,-(SP)      ;SAVE TRAP VECTOR POINTER
2119      017400 012767 017410 160426      MOV    81$,34      ;GO TO 1$ ON TRAP
2120      017406 104400      TRAP                    ;GO TO IT
2121      017410 016666 000002 000006      1$:    MOV    2(SP),6(SP)  ;GET PSM SAVED
2122      017416 012716 017424      MOV    82$, (SP)    ;GO TO 2$ ON RTI
2123      017422 000002      RTI                    ;
2124      017424 012667 160404      2$:    MOV    (SP)+,34    ;RESTORE VECTOR
2125      017430 012667 161544      MOV    (SP)+,STMPD  ;FINALLY SAVE PSM IN STMPD
2126      017434 000207      RTS     PC
2127
2128

```

2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183

.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.
017436 000000	DHADR: 0	:HOLDS THE "SCR" ADDRESS OF THE DH11 UNDER TEST
017440 000000	DHVCT: 0	:HOLDS THE 1ST VECTOR ADDRESS OF THE DH11 UNDER TEST
017442 000000	SELSK: 0	:BIT 1ST MARKER FOR SELECTING DH11'S
017444 000001	DHSEL: 1	:SPECIFIES DH11'S SELECTED FOR TEST
017446 177777	LINSEL: 177777	:SPECIFIES LINES TO TEST
017450 000000	LINMSK: 0	:MARKER USED TO TEST FOR LINES TO TEST
017452 000000	DAPLIN: 0	:DROPPED LINE FLAGS
017454 000000	QUICK: 0	:QUICK TEST FLAG - ALLOWS SINGLE PATTERN TEST
017456 000000	QUICKX: 0	:ON ALL TESTS NOT USING 9600. BAUD :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

017460 033500	LPRTAB: 33500	:9600 BAUD
017462 004200	4200	:75 BAUD
017464 006300	6300	:110 BAUD
017466 010400	10400	:134.5 BAUD
017470 012500	12500	:150 BAUD
017472 014600	14600	:200 BAUD
017474 016700	16700	:300 BAUD
017476 021000	21000	:600 BAUD
017500 023100	23100	:1200 BAUD
017502 025200	25200	:1800 BAUD
017504 027300	27300	:2400 BAUD
017506 031400	31400	:4800 BAUD
017510 002100	2100	:50 BAUD
017512 000000	CURLPR: 0	:CONTAINS CURRENT "LPR" CONSTANT
017514 000000	LPRPTR: 0	:CONTAINS POINTER TO LPR TABLE
017516 000000	CHRCNT: 0	:LOADED WITH CURRENT CHAR COUNT
017520 000000	CLSEL: 0	:CHAR LENGTH SELECT PARAMETER
017522 000000	PARBIT: 0	:PARITY SELECT PARAMETER

```

2184
2185 017524 000000
2186 017526 000000
2187
2188
2189
2190
2191
2192 017530 160020
2193 017532 160040
2194 017534 160060
2195 017536 160100
2196 017540 160120
2197 017542 160140
2198 017544 160160
2199 017546 160200
2200 017550 160220
2201 017552 160240
2202 017554 160260
2203 017556 160300
2204 017560 160320
2205 017562 160340
2206 017564 160360
2207 017566 160400
2208
2209
2210
2211 017570 000330
2212 017572 000350
2213 017574 000370
2214 017576 000410
2215 017600 000430
2216 017602 000450
2217 017604 000470
2218 017606 000510
2219 017610 000530
2220 017612 000550
2221 017614 000570
2222 017616 000610
2223 017620 000630
2224 017622 000650
2225 017624 000670
2226 017626 000710
2227
2228 017630 000000
2229
2230
2231
2232
2233
2234
2235
2236 017632 120240
2237 017634 120240
2238 017636 120240

```

```

ROONE: 0 ;SOFTWARE DONE FLAG
RBFEND: 0 ;HOLDS END OF BUFFER ADDRESS

```

```

;DH11 ADDRESS TABLE - THIS TABLE CONTAINS THE "SCR" ADDRESS FOR UP TO
;SIXTEEN DH11'S

```

```

DHADTB: 160020 ;ADDRESS OF FIRST DH11
        160040 ;ADDRESS OF SECOND DH11
        160060
        160100
        160120
        160140
        160160
        160200
        160220
        160240
        160260
        160300
        160320
        160340
        160360
        160400 ;ADDRESS OF THE LAST DH11

```

```

;DH11 VECTOR TABLE - THIS TABLE CONTAINS THE VECTOR ADDRESSES FOR UP
;TO SIXTEEN DH11'S

```

```

DHVCTB: 330 ;ADDRESS OF VECTOR FOR FIRST DH11
        350 ;ADDRESS OF VECTOR FOR SECOND DH11
        370
        410
        430
        450
        470
        510
        530
        550
        570
        610
        630
        650
        670
        710 ;ADDRESS OF VECTOR FOR LAST DH11

```

```

VCFLG: 0 ;VECTOR DISPLACEMENT FLAG

```

```

;BR PRIORITY LEVEL TABLE - THIS TABLE CONTAINS THE PRIORITY LEVELS
;FOR UP TO SIXTEEN DH11'S - THE RCVR LEVEL IS STORED IN THE LOW BYTE
;AND THE XMTTR LEVEL IN THE HIGH BYTE

```

```

VL: 120240 ;BRLEVELS FOR FIRST DH11
     120240 ;BR LEVELS FOR SECOND DH11
     120240

```



```

2292 017770 000062 RSPTAB: 50. ;50. BAUD
2293 017772 000100 100
2294 017774 000113 75. ;75. BAUD
2295 017776 000200 200
2296 020000 000156 110. ;110. BAUD
2297 020002 000300 300
2298 020004 002501 1345. ;134.5 BAUD
2299 020006 000400 400
2300 020010 000226 150. ;150. BAUD
2301 020012 000500 500
2302 020014 000310 200. ;200. BAUD
2303 020016 000600 600
2304 020020 000454 300. ;300 BAUD
2305 020022 000700 700
2306 020024 001130 600. ;600. BAUD
2307 020026 001000 1000
2308 020030 002260 1200. ;1200. BAUD
2309 020032 001100 1100
2310 020034 003410 1800. ;1800. BAUD
2311 020036 001200 1200
2312 020040 004540 2400. ;2400. BAUD
2313 020042 001300 1300
2314 020044 011300 4800. ;4800. BAUD
2315 020046 001400 1400
2316 020050 022600 9600. ;9600. BAUD
2317 020052 001500 1500

;ADDRESS POINTERS TO SET UP TABLES WHEN INPUTTING PARAMETERS
2321 020054 000000 ADPTR: 0 ;POINTS TO ADDRESS TABLE
2322 020056 000000 VCPTR: 0 ;POINTS TO VECTOR TABLE
2323 020060 000000 BRPTR: 0 ;POINTS TO BR LEVEL TABLE

2325 020062 000000 TITFLG: 0 ;FLAG TO ALLOW PRINTING TITLE ONLY ONCE
2326 020064 000000 TIMEA: 0 ;GENERAL PURPOSE TIMERS
2327 020066 000000 TIMEB: 0

2329 020070 000000 CEXIT: 0 ;CONTROL-C EXIT FLAG FM ECHO TESTS
2330 020072 000000 DPFLG: 0 ;PATTERNS TEST FLAG
2331 020074 000000 DATCNT: 0 ;ITERATION COUNTER FOR PATTERNS TEST
2332 020076 000000 DATPAT: 0 ;FLAGS TYPE PATTERN
2333 020100 000000 PATFLG: 0 ;DATA PATTERNS (CR) SEQUENCE FLAG
2334 020102 000000 SINGLE: 0 ;HOLDS SINGLE CHAR TEST PATTERN
2335 020104 000000 RETFLG: 0 ;ECHO TEST RETURN FLAG FM SETUP
2336 020106 000012 PATLIN: 10. ;PATTERNS TESTS ITERATION COUNT
2337 020110 000000 RANA: 0 ;RANDOM NO. ACCUMULATORS
2338 020112 000000 RANB: 0
2339 020114 000000 CLMSK: 0 ;CHAR LENGTH BIT CLR MASK
2340 020116 000000 EXFLAG: 0 ;ECHO TEST EXIT FLAGS
2341 020120 000020 ECBUF: .BLKW 16. ;DATA BUFFER FOR SINGLE LINE ECHO TEST
2342 020160 000000 DMFILL: 0 ;FILL CHAR AND COUNT FOR SINGLE LINE
2343 ;ECHO TESTS
2344 020162 000000 FILLA: 0 ;TEMP STORAGE FOR FILLER CHAR
2345 020164 000000 FILLB: 0 ;SAME FOR COUNT

```

G13

MAINDEC-11-DZDHN-A MACY11 27(663) 15-DEC-75 09:29 PAGE 24-4
DZDHN.A.P11 DH11 PROGRAM CONSTANTS AND VARIABLES

SEQ 0161

2346
2347

2349
2350
2351
2352
2353
2354
2355
2356

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

;INFORMATION FOR MESSAGE 1

020166 047516 020116 054105
020174 046440 046505 051117
020202 020131 051105 047522
020210 020122 020055 051104
020216 050117 042520 020104
020224 044514 042516 021440
020232 020040 000 000

EM1: .ASCIZ 'NON EX MEMORY ERROR - DROPPED LINE # '

2357

020235 040 050050 024503
020242 020040 041440 051125
020250 050114 020122 042040
020256 053105 042101 020122
020264 051040 043505 042101
020272 020122 020040 040527
020300 020123 020040 020040
020306 027523 000102

DH1: .ASCIZ '(PC) CURLPR DEVADR REGADR WAS S/B'

2358

.EVEN
DT1: .WORD SERRPC,CURLPR,SREG1,SREG2,SREG3,SREG4,0

2359

020312 001116 017512 001162
020320 001164 001166 001170
020326 000000

2360

020330 000 000 000
020333 000 000 000
020336 000 000 000

DF2: .BYTE 0,0,0,0,0,0,0,0

2361

;INFORMATION FOR MESSAGE 2

2362

2363

2364

020340 051124 047101 046523
020346 052111 042524 020122
020354 040506 051514 020105
020362 047111 042524 051122
020370 050125 020124 020055
020376 051104 050117 042520
020404 020104 044514 042516
020412 021440 020040 000

EM2: .ASCIZ 'TRANSMITTER FALSE INTERRUPT - DROPPED LINE # '

2365

2366

2367

2368

2369

;INFORMATION FOR MESSAGE 3

020417 102 043125 042506
020424 020122 041501 044524
020432 042526 051040 043505
020440 051511 042524 020122
020446 051105 047522 020122
020454 020055 051104 050117
020462 042520 020104 044514
020470 042516 021440 020040
020476 000

EM3: .ASCIZ 'BUFFER ACTIVE REGISTER ERROR - DROPPED LINE # '

2370

2371 ; INFORMATION FOR MESSAGE 4
 2372
 2373 020477 102 052131 020105 EM4: .ASCIZ 'BYTE COUNT REGISTER ERROR - DROPPED LINE # '
 020504 047503 047125 020124
 020512 042522 044507 052123
 020520 051105 042440 051122
 020526 051117 026440 042040
 020534 047522 050120 042105
 020542 046040 047111 020105
 020550 020043 000040

2374 ; INFORMATION FOR MESSAGE 5
 2375
 2376
 2377 020554 052503 051122 047105 EM5: .ASCIZ 'CURRENT ADDRESS REGISTER ERROR - DROPPED LINE # '
 020562 020124 042101 051104
 020570 051505 020123 042522
 020576 044507 052123 051105
 020604 042440 051122 051117
 020612 026440 042040 047522
 020620 050120 042105 046040
 020626 047111 020105 020043
 020634 000040

2378 ; INFORMATION FOR MESSAGE 6
 2379
 2380
 2381 020636 044523 047514 047440 EM6: .ASCIZ 'SILO OVERFLOW ERROR - DROPPED LINE # '
 020644 042522 043122 047514
 020652 020127 051105 047522
 020660 020122 020055 051104
 020666 050117 042520 020104
 020674 044514 042516 021440
 020702 020040 000

2382 ; INFORMATION FOR MESSAGE 7
 2383
 2384
 2385 020705 122 041505 044505 EM7: .ASCIZ 'RECEIVER FALSE INTERRUPT - DROPPED LINE # '
 020712 042526 020122 040506
 020720 051514 020105 047111
 020726 042524 051122 050125
 020734 020124 020055 051104
 020742 050117 042520 020104
 020750 044514 042516 021440
 020756 020040 000

2386 ; INFORMATION FOR MESSAGE 10
 2387
 2388
 2389 020761 111 053116 046101 EM10: .ASCIZ 'INVALID DATA IN SILO - DROPPED LINE # '
 020766 042111 042040 052101
 020774 020101 047111 051440
 021002 046111 020117 020055
 021010 051104 050117 042520
 021016 020104 044514 042516
 021024 021440 020040 000
 2390 021031 040 050050 024503 DM2: .ASCIZ '(PC) CURLPR CHAR # WASADR SHBADR WAS S/B'

	021036	020040	041440	051125	
	021044	050114	020122	041440	
	021052	040510	020122	020043	
	021060	053440	051501	042101	
	021066	020122	051440	041110	
	021074	042101	020122	020040	
	021102	040527	020123	020040	
	021110	020040	027523	000102	
2391					.EVEN
2392	021116	001116	017512	001160	DT2: .WORD SERRPC,CURLPR,\$REG0,\$REG1,\$REG2,\$REG3,\$REG4,0
	021124	001162	001164	001166	
	021132	001170	000000		
2393					
2394					;INFORMATION FOR MESSAGE 11
2395					
2396	021136	040504	040524	042440	EM11: .ASCIZ 'DATA ERROR - LINE # '
	021144	051122	051117	026440	
	021152	046040	047111	020105	
	021160	020043	000040		
2397					
2398					;INFORMATION FOR MESSAGE 12
2399					
2400	021164	042524	052123	052040	EM12: .ASCIZ 'TEST TIMEOUT - DROPPED LINE # '
	021172	046511	047505	052125	
	021200	026440	042040	047522	
	021206	050120	042105	046040	
	021214	047111	020105	020043	
	021222	000040			
2401	021224	024040	041520	020051	DH3: .ASCIZ '(PC) CURLPR RTOTAL XTOTAL RDONE'
	021232	020040	052503	046122	
	021240	051120	020040	052122	
	021246	052117	046101	020040	
	021254	052130	052117	046101	
	021262	020040	042122	047117	
	021270	000105			
2402					.EVEN
2403	021272	001116	017512	001200	DT3: .WORD SERRPC,CURLPR,STMP0,STMP1,RDONE,0
	021300	001202	017524	000000	
2404					
2405					;INFORMATION FOR MESSAGE 13
2406					
2407	021306	001200	001202	001204	DT4: .WORD STMP0,STMP1,STMP2,STMP3,STMP4,STMP5,STMP6,0
	021314	001206	001210	001212	
	021322	001214	000000		
2408	021326	000	001	001	DF1: .BYTE 0,1,1,1,1,1,1,0
	021331	001	001	001	
	021334	001	000		
2409					
2410					;INFORMATION FOR MESSAGE 14
2411					
2412	021336	052502	020123	051105	EM14: .ASCIZ 'BUS ERROR TRAP TO 04'
	021344	047522	020122	051124	
	021352	050101	052040	020117	
	021360	032060	000		

2413	021363	040	050050	024503	DH4:	.ASCIZ	'(PC)	(PS)	(SP)	TRAPPC	TRAPPS'
	021370	020040	020040	050050							
	021376	024523	020040	020040							
	021404	051450	024520	020040							
	021412	052040	040522	050120							
	021420	020103	052040	040522							
	021426	050120	000123								

2414					.EVEN						
2415	021432	001116	001200	001174	DT5:	.WORD	SERRPC,	STMPO,	SREG6,	SREG1,	SREG2,0
	021440	001162	001164	000000							

;INFORMATION FOR MESSAGE 15

2416					EM15:	.ASCIZ	'RSVD INSTR TRAP TO 10'
2417							
2418							
2419	021446	051522	042126	044440			
	021454	051516	051124	052040			
	021462	040522	020120	047524			
	021470	030440	000060				

;INFORMATION FOR MESSAGE 16

2420					EM16:	.ASCIZ	'SINGLE LINE ECHO TEST - INTR WAIT TIMEOUT'
2421							
2422							
2423	021474	044523	043516	042514			
	021502	046040	047111	020105			
	021510	041505	047510	052040			
	021516	051505	020124	020055			
	021524	047111	051124	053440			
	021532	044501	020124	044524			
	021540	042515	052517	000124			

2424	021546	024040	041520	020051	DH5:	.ASCIZ	'(PC)	DEVAOR	LINE	(SCR)	CURLPR	EXFLAG'
	021554	020040	042504	040526								
	021562	051104	020040	046040								
	021570	047111	020105	020040								
	021576	024040	041523	024522								
	021604	020040	052503	046122								
	021612	051120	020040	054105								
	021620	046106	043501	000								

					.EVEN							
	021626	001116	001162	017676	DT6:	.WORD	SERRPC,	SREG1,	LINE,	STMPO,	CURLPR,	EXFLAG,0
	021634	001200	017512	020116								
	021642	000000										

;INFORMATION FOR MESSAGE 17

2425					EM17:	.ASCIZ	'ALTERNATING I/O PATTERN TEST DONE'
2426							
2427							
2428	021644	046101	042524	047122			
	021652	052101	047111	020107			
	021660	027461	020060	040520			
	021666	052124	051106	020116			
	021674	042524	052125	042040			
	021702	047117	000105				

2432	021706	024040	041520	020051	DH6:	.ASCIZ	'(PC)	DEVAOR	LINE	CURLPR	ICOUNT'
	021714	020040	042504	040526							
	021722	051104	020040	046040							
	021730	047111	020105	020040							
	021736	052503	046122	051120							

	021744	020040	041511	052517	
	021752	052116	000		
2433		021756			.EVEN
2434	021756	001116	017436	017676	DT7: .WORD SERRPC,DHADR,LINE,CURLPR,SREGO,0
	021764	017512	001160	000000	
2435					;INFORMATION FOR MESSAGE 20
2436					EM20: .ASCIZ 'BINARY UP COUNT PATTERN TEST DONE'
2437	021772	044502	040516	054522	
2438	022000	052440	020120	047503	
	022006	047125	020124	040520	
	022014	052124	051105	020116	
	022022	042524	052123	042040	
	022030	047117	000105		
2439					;INFORMATION FOR MESSAGE 21
2440					EM21: .ASCIZ 'BINARY DOWN COUNT PATTERN TEST DONE'
2441	022034	044502	040516	054522	
2442	022042	042040	053517	020116	
	022050	047503	047125	020124	
	022056	040520	052124	051105	
	022064	020116	042524	052123	
	022072	042040	047117	000105	
2443					;INFORMATION FOR MESSAGE 22
2444					EM22: .ASCIZ 'RANDOM DATA PATTERN TEST DONE'
2445	022100	040522	042116	046517	
2446	022106	042040	052101	020101	
	022114	040520	052124	051105	
	022122	020116	042524	052123	
	022130	042040	047117	000105	
2447					;INFORMATION FOR MESSAGE 23
2448					EM23: .ASCIZ 'SINGLE CHAR PATTERN TEST DONE'
2449	022136	044523	043516	042514	
2450	022144	041440	040510	020122	
	022152	040520	052124	051105	
	022160	020116	042524	052123	
	022166	042040	047117	000105	
2451					;INFORMATION FOR MESSAGE 24
2452					EM24: .ASCIZ 'TYPED BUFFER PATTERN TEST DONE'
2453	022174	054524	042520	020104	
2454	022202	052502	043106	051105	
	022210	050040	052101	042524	
	022216	047122	052040	051505	
	022224	020124	047504	042516	
	022232	000			
2455					;INFORMATION FOR MESSAGE 25
2456					EM25: .ASCIZ 'DATA PATTERNS TEST TIMEOUT'
2457	022233	104	052101	020101	
2458	022240	040520	052124	051105	

	022246	051516	052040	051505								
	022254	020124	044524	042515								
	022262	052517	000124									
2460	022266	024040	041520	020051	DH7:	.ASCIZ	' (PC)	DEVADR	LINE	CURLPR	ICOUNT	PATCDE'
	022274	020040	042504	040526								
	022302	051104	020040	046040								
	022310	047111	020105	020040								
	022316	052503	046122	051120								
	022324	020040	041511	052517								
	022332	052116	020040	040520								
	022340	041524	042504	000								
2461		022346			.EVEN							
2462	022346	001116	017436	017676	DT10:	.WORD	SERRPC,	DHADR,	LINE,	CURLPR,	SREG0,	SREG1,0
	022354	17512	001160	001162								
	022362	000000										
2463												
2464					.EVEN							

2467
2468
2469
2470
2471
2472

.SBTTL MISCELLANEOUS TABLES AND MESSAGE AND DATA BUFFERS
:*****
:MISCELLANEOUS MESSAGES
:*****

TITLE: .ASCIZ <15><12>'MAINDEC-11-DZDHN-A DH11 DATA RELIABILITY TEST'<15><12>

2473

TITLE2: .ASCIZ <15><12>'TESTING DH11 # ' <15><12>

2474

INMSG1: .ASCIZ <15><12>'TYPE SCR ADDRESS FOR FIRST DH11'<15><12>

2475

INMSG2: .ASCIZ <15><12>'TYPE VECTOR ADDRESS FOR FIRST DH11'<15><12>

2476

INMSG3: .ASCIZ <15><12>'TYPE DH11 DEVICE SELECTION PARAMETER'<15><12>

2477

INMSG4: .ASCIZ <15><12>'INVALID DH11 SCR ADDRESS - TRY AGAIN'<15><12>

2478

INMSG5: .ASCIZ <15><12>'INVALID DH11 VECTOR ADDRESS - TRY AGAIN'<15><12>

022364 005015 040515 047111
022372 042504 026503 030461
022400 042055 042132 047110
022406 040455 042040 030510
022414 020061 040504 040524
022422 051040 046105 040511
022430 044500 044514 054524
022438 052040 051505 006524
022446 000012
022454 005015 042524 052123
022462 047111 020107 044104
022470 030461 021440 020040
022478 005015 000
022486 015520 052012 050131
022494 040100 041523 040122
022502 020100 051104 051505
022510 044500 047506 020122
022518 041046 030461 005015
022526 005015 052012 050131
022534 020100 042526 052103
022542 051117 040440 042104
022550 052117 051523 043040
022558 052117 040440 051111
022566 052123 042040 030510
022574 005015 000012 042520
022582 042040 030510 020061
022590 042504 044500 042503
022598 051440 046105 041505
022606 047111 047111 050440
022614 011010 005015 021040
022622 015520 044500 000
022630 015520 044500 053116
022638 005015 044500 042040
022646 015520 042107 051104
022654 005015 020107 051104
022662 044500 047111 051104
022670 044500 047111 051104
022678 044500 047111 051104
022686 044500 047111 051104
022694 044500 047111 051104
022702 044500 047111 051104
022710 044500 047111 051104
022718 044500 047111 051104
022726 044500 047111 051104
022734 044500 047111 051104
022742 044500 047111 051104
023002 000012

2479	023004 023012 023020 023028 023034 023042 023050	005015 052515 046105 052101 052123 042040 000011	047531 052123 041505 046040 047440 030510 000011	020125 051440 020124 040505 042516 006461 000011	INMSG6: .ASCIZ <15><12>'YOU MUST SELECT AT LEAST ONE DH11'<15><12>
2480	023058 023060 023066 023074 023102 023110 023116	005015 051505 047117 021105 052123 042524 006507	042504 020123 044524 052040 051101 052123 000012	051120 041442 052516 020117 020124 047111 000012	INMSG7: .ASCIZ <15><12>'DEPRESS "CONTINUE" TO START TESTING'<15><12>
2481	023122 023128 023134 023140 023150 023160	005015 046040 047524 047514 040516 006522	054524 047111 042514 020116 040524 000012	042520 020109 052103 040520 042524 000012	INMSG8: .ASCIZ <15><12>'TYPE LINE SELECTION PARAMETER'<15><12>
2482	023164 023170 023200 023206 023212 023218 023224 023230 023236 023242 023248 023254 023260 023266 023272 023278	005015 047040 020109 020109 046101 046101 046101 020123 020123 020123 020123 020123 020123 020123 020123 020123	054524 027117 047527 047450 020051 042505 052103 032120 030061	042520 047440 042122 052103 042502 020116 051117 047440 006451	VCMC: .ASCIZ <15><12>'TYPE NO. OF WORDS (OCTAL) BETWEEN VECTORS (4 OR 10)'<15><12>
2484	023280 023286 023292 023298 023304	005015 020040 044524 051223 000011	044104 051440 051123 000000 000011	021440 040524 041511 000000 000011	STMSG1: .ASCIZ <15><12>'DH # STATISTICS:'
2485	023308 023314 023320 023326 023332	0015 044524 047111 006440 005015	052012 043516 020105 000012 044514	051505 046040 020043 053440 042516	STMSG2: .ASCIZ <15><12>'TESTING LINE # '<15><12>
2486	023338 023344 023350 023356 023362 023368 023374	000011 051501 050120 000011 000011 000011 000011	042040 042040 042105 000011 000011 000011 000011	042516 053440 047522 005015 000011 000011 000011	STMSG3: .ASCIZ <15><12>'LINE # WAS DROPPED'<15><12>
2487	023380 023386 023392 023398 023404 023410 023416 023422 023428 023434 023440	000011 020105 047524 054040 020114 051105 051101 043040 020122 051105 006522	046012 020043 040524 047524 042040 020122 051105 046522 047440 051126 000012	047111 051040 020114 040524 052101 050040 020122 051105 051126 000012	STMSG4: .ASCIZ <15><12>'LINE # RTOTAL XTOTAL DATERR PARERR FRMERR OVRERR'<15><12>

;MESSAGES FOR INPUTTING PARAMETERS TO ECHO TESTS

2496
2497
2498
2499
2500
2501
2502

0233452	020040	006440	000012
0233453	000040		
0233454	020040	005015	000
0233455	015	051411	047111
0233456	04	020107	044514
0233457	04	041440	044103
0233458	04	041440	047117
0233459	04	052040	052040
0233460	04	040516	040516
0233461	04	042040	042040
0233462	04	042524	042524
0233463	04	046040	047111
0233464	000012		
023566	005015	054524	042520
023574	046040	047111	020105
023602	020043	030050	020060
023610	020055	033461	030040
023616	052103	046101	000051
023624	005015	042524	052123
023632	047111	020107	044514
023640	042516	021440	020040
023646	026440	043440	020117
023654	054524	042520	044440
023662	020116	047117	052040
023670	051505	020124	044514
023676	042516	005015	000
023703	015	052012	050131
023710	03	053440	047503
023716	020116	047524	026514
023724	020103	047524	042440
023732	044520	051440	051440
023740	047503	052116	047522
023746	026514	020106	047524
023754	042440	044103	020117
023762	052502	043106	051105
023770	005015	000012	
024002	005015	047504	054440
024008	052517	053440	047101
024010	020124	047524	041440
024016	040510	043516	020105
024024	046042	051120	020042
024032	054450	047440	020122
024040	024516	037440	000040
024046	005015	051124	047101
024054	046523	052111	042524
024062	020122	050123	042505
024070	020104	000077	

EC: .ASCIZ ' (15)<(12)>
 EC2: .ASCIZ ' , ' (15)<(12)>
 EC3: .ASCIZ ' (15)<(12)>
 ECMSG1: .ASCIZ <(15)<(12)>'SINGLE LINE ECHO TEST - CONNECT TERMINAL TO DH11 TEST LINE'<(15)>

ECMSG2: .ASCIZ <(15)<(12)>'TYPE LINE # (00 - 17 OCTAL)'

ECMSG3: .ASCIZ <(15)<(12)>'TESTING LINE # - GO TYPE IN ON TEST LINE'<(15)<(12)>

ECMSG4: .ASCIZ <(15)<(12)>'TYPE: [CONTROL-C TO EXIT] [CONTROL-E TO ECHO BUFFER]'<(15)<(12)>

LPMSG: .ASCIZ <(15)<(12)>'DO YOU WANT TO CHANGE "LPR" (Y OR N) ? '

XMSG1: .ASCIZ <(15)<(12)>'TRANSMITTER SPEED ?'

2503					
2504	024074	005015	047111	040526	XMSG2: .ASCIZ <15><12>'INVALID XMIT SPEED - TRY AGAIN'<15><12>
	024102	044514	020104	046530	
	024110	052111	051440	042520	
	024116	042105	026440	052040	
	024124	054522	040440	040507	
	024132	047111	005015	000	
2505					
2506	024137	015	051012	041505	RSMSG1: .ASCIZ <15><12>'RECEIVER SPEED ?'
	024144	044506	042526	020122	
	024150	050123	042505	020104	
	024156	000077			
2507					
2508	024162	005015	047111	040526	RSMSG2: .ASCIZ <15><12>'INVALID RCVR SPEED - TRY AGAIN'<15><12>
	024170	044514	020104	041522	
	024176	051126	051440	042520	
	024184	042105	026440	052040	
	024192	054522	040440	040507	
	024200	047111	005015	000	
2509					
2510	024226	015	041412	040510	CLMSG1: .ASCIZ <15><12>'CHAR LENGTH (6, 7, OR 8) ?'
	024232	020122	042514	043516	
	024240	044124	024040	026056	
	024246	033440	020054	051117	
	024254	034040	020051	020077	
	024262	000			
2511					
2512	024263	015	044412	053116	CLMSG2: .ASCIZ <15><12>'INVALID CHAR LENGTH - TRY AGAIN'<15><12>
	024270	046101	042111	041440	
	024276	040510	020122	042514	
	024284	043516	044124	026440	
	024292	052040	054522	040440	
	024300	040507	047111	005015	
	024308	000			
2513					
2514	024327	015	047012	027117	SBMSG1: .ASCIZ <15><12>'NO. OF STOP BITS (1 OR 2) ?'
	024334	047440	020106	052123	
	024342	050117	041040	052111	
	024350	020123	030450	047440	
	024358	020123	024462	037440	
	024364	000040			
2515					
2516	024366	005015	047111	040526	SBMSG2: .ASCIZ <15><12>'INVALID NO. STOP BITS - TRY AGAIN'<15><12>
	024374	044514	020104	047516	
	024402	020056	052123	050117	
	024410	041040	052111	020123	
	024416	020055	051124	020131	
	024424	043501	044501	006516	
	024432	000012			
2517					
2518	024434	005015	040520	044522	PBMSG1: .ASCIZ <15><12>'PARITY SELECTION (E, O, OR <CR>) ?'
	024442	054524	051440	046105	
	024450	041505	044524	047117	
	024456	024040	026105	047440	

	024464	020054	051117	036040	
	024472	051103	024476	037440	
	024500	000040			
2519					
2520	024502	005015	047111	040526	PBMSG2: .ASCIZ <15><12>'INVALID PARITY - TRY AGAIN'<15><12>
	024510	044514	020104	040520	
	024516	044522	054524	026440	
	024524	052040	054522	040440	
	024530	040507	047111	005015	
	024540	000			
2521					
2522	024541	015	043012	046111	FILC1: .ASCIZ <15><12>'FILLER CHARACTER ? '
	024546	042514	020122	044103	
	024554	051101	041501	042524	
	024562	020122	020077	000	
2523					
2524	024567	015	043012	046111	FILC2: .ASCIZ <15><12>'FILLER COUNT ?'
	024574	042514	020122	047503	
	024582	047122	020122	000077	
2525	024590	005015	042524	042116	SNMSG1: .ASCIZ <15><12>'SEND MODE -(Y OR N) '
	024596	046440	042117	020105	
	024604	024050	020131	051117	
	024612	047040	020051	000040	
2526	024640	005015	054524	042520	SNMSG2: .ASCIZ <15><12>'TYPE SEND BUFFER - TERMINATE WITH CONTROL-C'<15><12>
	024646	051440	047105	020104	
	024654	052502	043106	051105	
	024662	026440	052040	051105	
	024670	044515	040516	042524	
	024676	053440	052111	020110	
	024704	047503	052116	047522	
2527	024712	026514	006503	000012	SNMSG3: .ASCIZ <15><12>'CHANGE PARAMETERS (Y OR N)? '
	024720	005015	044103	047101	
	024726	042507	050040	051101	
	024734	046501	052105	051105	
	024742	020123	054450	047440	
	024750	020122	024516	020077	
	024756	000			
2528					

2531 :MESSAGES FO INPUTTING PATTERNS TEST PARAMETERS
2532 DPMSG1: .ASCIZ <15><12>'DATA PATTERNS TESTS - CONNECT TEST JUMPER'<15><12>

024757 015 042012 052101
024764 020101 040520 052124
024772 051516 052140
024780 051524 052140
024788 047117 042516
024796 052040 051505
024804 052040 050115
024812 052040 000
024820 041011 043123
024828 020123 044523
024836 037440 024040
024844 020050 030465
024852 000

2533 DPMSG2: .ASCIZ <15><12>'BUFFER SIZE ? (1 - 512)'

024860 044412 053116
024868 042111 051440
024876 020105 020055
024884 020131 043501
024892 006516 000012
024900 040520 052124
024908 020116 054524
024916 037440 024040
024924 026123 026104
024932 026123 020102
024940 036040 051103
024948 006440 000012
024956 042523 020124
024964 033460 030475
024972 020117 047514
024980 047440 020116
024988 052124 051105
024996 000012

2534 DPMSG3: .ASCIZ <15><12>'INVALID SIZE - TRY AGAIN'<15><12>

025004 044501 000012
025012 051105 052124
025020 042520 024040
025028 026101 026104
025036 026122 020102
025044 051117 051103
025052 044717 000012
025060 005015 042523
025068 051123 033460
025076 042503 020117
025084 047503 047440
025092 047503 020116
025100 052124 051105
025108 000012

2535 DPMSG4: .ASCIZ <15><12>'PATTERN TYPE ? (A,U,D,R,S,B OR <CR>) '<15><12>

025116 047111 040526
025124 020104 040520
025132 051105 020116
025140 051124 020131
025148 044501 006516
025156 000012

2536 DPMSG5: .ASCIZ <15><12>'SET SR07=1 TO LOCK ON PATTERN'<15><12>

025164 005015 054524 042520
025172 051440 047111 046107
025180 020105 042524 052123
025188 041440 040510 020122
025196 000

2537 DPMSG6: .ASCIZ <15><12>'INVALID PATTERN - TRY AGAIN'<15><12>

025204 052012 050131
025212 047111 052040
025220 020124 052502
025228 051105 026440
025236 052040 051105 044515
025244 040516 042524 053440
025252 052111 020110 047503
025260 052116 047522 026514
025268 006503 000012

2538 DPMSG7: .ASCIZ <15><12>'TYPE SINGLE TEST CHAR '<15><12>

025276 000012
025284 000012
025292 000012
025300 000012
025308 000012
025316 000012
025324 000012
025332 000012
025340 000012
025348 000012
025356 000012
025364 000012
025372 000012
025380 000012
025388 000012
025396 000012
025404 000012
025412 000012

2539 DPMSG8: .ASCIZ <15><12>'TYPE IN TEST BUFFER - TERMINATE WITH CONTROL-C'<15><12>

025420 000012
025428 000012
025436 000012
025444 000012
025452 000012
025460 000012
025468 000012
025476 000012
025484 000012
025492 000012
025500 000012
025508 000012
025516 000012
025524 000012
025532 000012
025540 000012
025548 000012
025556 000012
025564 000012
025572 000012
025580 000012
025588 000012
025596 000012
025604 000012
025612 000012

2540 .EVEN
2541 ;ERROR STATISTICS TABLES

```

02543 025416 000020 RTOTAL: .BLKW 16. ;TOTAL CHARS RCVD PER LINE
02544 025456 000020 XTOTAL: .BLKW 16. ;TOTAL CHARS XMITTED PER LINE
02545 025516 000020 DATERR: .BLKW 16. ;TOTAL DATA COMPARE ERRORS PER LINE
02546 025556 000020 PARERR: .BLKW 16. ;TOTAL PARITY ERRORS PER LINE
02547 025616 000020 OVRERR: .BLKW 16. ;TOTAL OVERRUN ERRORS PER LINE
02548 025656 000020 FRMERR: .BLKW 16. ;TOTAL FRAMING ERRORS PER LINE

;STATISTICS TABLES POINTERS
025716 000000 DEPTR: 0 ;CONTAINS POINTERS TO STAT TABLES
025720 000000 PEPTR: 0
025722 000000 ORPTR: 0
025724 000000 FRPTR: 0

025726 000000 RBFPTR: 0 ;CONTAINS INPUT BUFFER POINTER
025730 000000 TBFPTR: 0 ;CONTAINS OUTPUT BUFFER POINTER

;600. WORD RECEIVER INPUT BUFFER
025732 001130 RBUF: .BLKW 600.

;600(10) BYTE TRANSMITTER OUTPUT DATA BUFFER
02568 030212 001130 .EVEN
02569 030212 001130 TBUF: .BLKB 600.

02571 031342 000000 ENBUFS: 0 ;MARK END OF BUFFERS
02572 000001 .END
02573

```


BCR = 000010	368	372	376	735*	765*	776*	787*	1197	1201	1205	1403*	1773*	1795*
BEGIN 001624	2138#												
BEGINA 001642	18	176#											
BIT0 = 000001	180#	630	855	1492	1497								
BIT00 = 000001	23#	414											
BIT01 = 000002	23#												
BIT02 = 000004	23#												
BIT03 = 000010	23#												
BIT04 = 000020	23#												
BIT05 = 000040	23#												
BIT06 = 000100	23#	693	748	757									
BIT07 = 000200	23#	445	1266										
BIT08 = 000400	23#												
BIT09 = 001000	23#	1351	1352										
BIT1 = 000002	23#	485											
BIT10 = 002000	23#	311	1140										
BIT11 = 004000	23#	288	303	316	334	354	373	425	444	462	684	713	1121
	1130	1145	1163	1183	1202	1246	1265	1283	1351	1391	1769		
BIT12 = 010000	23#												
BIT13 = 020000	23#	723	738	790	1352								
BIT14 = 040000	23#	420	1241	1351									
BIT15 = 100000	23#	335	511	723	1164	1318							
BIT2 = 000004	23#												
BIT3 = 000010	23#												
BIT4 = 000020	23#												
BIT5 = 000040	23#												
BIT6 = 000100	23#												
BIT7 = 000200	23#												
BIT8 = 000400	23#												
BIT9 = 001000	23#												
BKR = 000014	2140#												
BPTVEC = 000014	23#												
BRLVL 017632	213	2235#											
BRPTR 020060	213#	220#	228	613*	2323#								
BUSER 014750	185	1598#											
CAR = 000006	387	396	736*	788*	1216	1225	1404*	1774*	2137#				
CEXIT 020070	680#	706	742*	2329#									
CHKADR 014526	1504	1542#											
CHKVCT 014640	1512	1569#											
CHPS1 017342	277	702	1112	1605	1615	1763	2101#						
CHPS2 017356	315	333	352	371	394	423	441	461	679	1144	1162	1181	1200
	1223	1244	1262	1282	1392	2109#							
CHRCNT 017516	264#	388	408	480	809	879#	1094*	1217	1399	1403	1693*	1696*	1710
	2021	2031	2043	2055	2072	2180#							
CKER 004010	304	493#											
CKEROP 010476	1131	1310#											
CKRST1 013670	1362	1372#											
CKRST2 013720	1380#	1606	1616										
CLALL 017324	2020	2030	2042	2054	2071	2092#							
CLMSG1 024225	1895	2510#											
CLMSG2 024263	1919	2512#											
CLMSK 020114	1319	2082#	2086#	2339#									
CLSEL 017520	265#	1686#	1687	1692*	1695	2182#							

CLSTAT	015124	238	1647#											
CURLPR	017512	694	1402	1662	1664*	1667	1694*	1695*	1732*	1733*	1775	1803*	1810*	1822*
		1832*	1839*	1851*	1878	1880*	1896*	1910*	1914*	1921*	1942*	1949*	1957*	1970*
		1974*	2081	2177*	2359	2392	2403	2426	2434	2462				
DATCNT	020074	942*	945*	946	949	950*	960*	963*	964	967	968*	978*	981*	982
		985	986*	996*	999*	1000	1003	1004*	1027*	1045*	1046	1049	1050*	1059*
		1096*	1097	1100	1101*	1122	2331#							
DATERR	025516	495	593	2545#										
DATPAT	020076	899*	912	916	920	924	928	932	1014*	1016*	1018*	1020*	1123	2332#
DDISP	= 177570	23#	24											
DEPTR	025716	495*	501*	545*	2552#									
DF1	021326	101	2408#											
DF2	020330	31	38	45	52	59	66	73	80	87	94	108	115	122
		129	136	143	150	157	164	171	2360#					
DHADR	017436	226*	249	288*	632*	683	857*	892	1121*	1431*	1553*	1557*	1768	2143#
		2434	2462											
DHADTB	017530	211	1559	1562	2191#									
DHFILL	020160	1987*	1999*	2000	2011*	2013*	2015*	2016	2342#					
DHNUM	017674	214*	217*	230	567	610*	2253#							
DHRLVL	017672	228*	245	682*	687	881*	884	2251#						
DHSEL	017444	221	617	1526*	2146#									
DHSET1	013750	275	1110	1391#										
DHST1	002576	275#												
DHST2	007422	944	962	980	998	1044	1095	1110#						
DHTLVL	017673	248	690	887	2252#									
DHVCT	017440	227*	243	633*	685	858*	882	1580*	1584*	1621	2144#			
DHVCTB	017570	212	1586	1589	2211#									
DHMAIT	005150	704#												
DH1	020235	29	36	43	50	57	64	71	2357#					
DH2	021031	78	85	2390#										
DH3	021224	92	2401#											
DH4	021363	106	113	2413#										
DH5	021546	120	2424#											
DH6	021706	127	134	141	148	155	162	2432#						
DH7	022266	169	2460#											
DISPLA	001140	24#	182*	1351*	1352*									
DISPRE	000174	16#	182											
DPATA	006370	914	942#	1015										
DPATB	007202	934	1059#											
DPATCR	006760	910	1014#	1023										
DPATD	006564	922	978#	1019										
DPATR	006662	926	996#	1021										
DPATS	007042	930	1026#											
DPATU	006466	918	960#	1017										
DPFLG	020072	178*	201	628*	654	852*	1372	1380	1487*	1495*	1518	1555	1582	2330#
DPMSG	025333	1061	2539#											
DPMSG1	024757	860	2532#											
DPMSG2	025035	866	2533#											
DPMSG3	025067	874	2534#											
DPMSG4	025124	894	2535#											
DPMSG5	025176	906	2536#											
DPMSG6	025240	937	2537#											
DPMSG7	025300	1029	2538#											
DRFLIN	017452	241*	298*	327*	346*	365*	384*	404*	435*	455*	472*	574	2149#	

XSPTR	017700	1802*	1807	1809*	1810	1814*	1815	2262#
XTOTAL	025456	202*	412*	592	2544#			
SAPTHD	000000	15#						
SASTAT=	##### U	11#						
SATYC	012546	11#						
SATY1	012522	11#						
SATY3	012530	11#	1357#					
SATY4	012540	11#	1357#					
SBASE	001304	T						
SBDADR	001122	T						
SBDAT	001126	T	182					
SCM1	001310	T						
SCM2	001312	T						
SCHARC	012516	13#						
SCHTAG	001100	T	182					
SCH1	000010	T						
SCH2	000020	T						
SCH3	000010	T						
SCH4	000010	T						
SCPUOP	001255	T						
SCRLF	001255	T	1352	1353	1356	1358	1359	1360
SDBLK	012234	13#						
SDDM0	001314	T						
SDDM1	001316	T						
SDDM10	001340	T						
SDDM11	001342	T						
SDDM12	001344	T						
SDDM13	001346	T						
SDDM14	001348	T						
SDDM15	001350	T						
SDDM2	001320	T						
SDDM3	001322	T						
SDDM4	001324	T						
SDDM5	001326	T						
SDDM6	001330	T						
SDDM7	001332	T						
SDDM8	001334	T						
SDDM9	001336	T						
SDEVCT	001240	T						
SDEVH	001306	T						
SDDAGN	010762	13#						
SOTBL	012224	11#						
SENORD	010752	11#						
SENOCT	010720	11#						
SENOHC	010766	11#						
SENULL	011003	11#						
SENV	001250	T	1352	1356	1357			
SENVH	001251	T	1802	1356	1357			
SEOP	010664	13#	616	1350#				
SEOPCT	010712	13#						
SEFLG	001103	T	1351#	1352#				
SEMAX	001115	T	182#	1351#				
SERROR	011250	13#	1352#					
SERRPC	001116	13#	1352#	1353	2359	2392	2403	2415
					2426	2434	2462	

.STYPO 6# 1354

ADC	2060	2062													
ADD	218	219	220	357	376	390	396	412	477	484	501	502	503	504	537
	611	612	613	1186	1205	1219	1225	1297	1300	1340	1343	1353	1354	1355	1356
	1357	1359	1360	1401	1428	1453	1456	1550	1561	1577	1588	1623	1626	1640	1672
	1673	1698	1745	1809	1814	1838	1843	2059	2061						
ASL	223	290	411	483	500	548	551	554	586	600	614	650	1353	1359	1360
	1361	1424	1696												
ASLB	1355	2086													
ASR	1357	1449	1450	1451											
BCC	1355														
BEQ	182	195	199	202	205	224	283	312	350	369	392	421	514	532	544
	575	588	601	618	649	655	664	670	672	725	727	729	807	831	838
	840	869	898	1033	1070	1073	1141	1179	1198	1221	1242	1299	1322	1342	1350
	1351	1352	1353	1354	1356	1357	1359	1360	1425	1427	1478	1480	1482	1503	1511
	1516	1519	1525	1532	1552	1556	1579	1583	1663	1668	1688	1690	1728	1730	1759
	1806	1835	1865	1872	1874	1900	1906	1917	1932	1938	1945	1961	1967	1991	1998
	2010	2085													
BGE	1351	1356	1543	1570											
BGT	810	1350	1354	1355	1359	1360									
BHT	1351														
BIC	318	337	426	443	644	723	757	759	1147	1166	1247	1264	1319	1350	1354
	1358	1359	1452	1455	1694	1732	1803	1832	1896	1949	1957	2083			
BICB	1431	1907	1939	2015											
BIS	276	288	298	327	346	365	384	404	414	435	455	472	485	511	693
	713	738	748	790	1111	1130	1303	1318	1354	1355	1695	1733	1810	1822	1839
	1851	1910	1914	1921	1942	1970	1974								
BISB	336	446	509	1165	1267	1316	1353	1395							
BIT	221	311	420	531	574	587	617	1140	1241	1351	1352	1360	1426	1548	1575
BITB	182	1356	1357												
BLOS	1358														
BLT	731	872	1354	1355	1356	1359	1360	1546	1573						
BMI	331	439	459	957	975	993	1011	1023	1054	1105	1160	1260	1280	1355	
BNE	182	190	207	222	479	539	615	674	707	761	770	781	812	818	842
	908	913	917	921	925	929	933	947	954	965	972	983	990	1001	1008
	1041	1047	1076	1082	1098	1117	1302	1345	1351	1352	1353	1354	1355	1356	1357
	1358	1360	1362	1373	1375	1381	1383	1418	1549	1563	1576	1590	1637	1639	1650
	1685	1715	1735	1739	1743	1791	1808	1816	1837	1845	1876	1879	1909	1913	1941
	1969	1973	2025	2037	2049	2066	2076	2095							
BPL	549	552	555	1352	1354	1355	1356	1358	1535	1666	1779				
BR	182	225	252	253	263	269	273	285	300	305	583	602	651	668	709
	740	743	767	778	783	819	822	835	875	903	938	1038	1083	1086	1119
	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1362	1422	1429	1483	1485
	1505	1513	1544	1547	1554	1558	1564	1571	1574	1581	1585	1591	1670	1737	1741
	1812	1820	1841	1849	1869	1877	1881	1904	1911	1915	1920	1936	1943	1948	1965
	1971	1975	1978	1995	2012	2088									
BVS	1360														
CLR	176	177	178	179	180	182	197	240	241	242	267	270	281	355	374
	564	628	629	680	681	705	746	853	854	890	942	950	960	968	978
	986	996	1004	1027	1050	1059	1101	1115	1184	1203	1350	1351	1353	1354	1355
	1359	1360	1362	1396	1430	1487	1488	1494	1495	1496	1624	1627	1648	1671	1692
	1709	1984	1987	2005	2033	2093	2117								
CLRB	508	608	658	825	891	1026	1315	1351	1355	1356	1357	1358	1359	1360	1421
	1859	1893	1926	1954	1983										
CMP	182	206	282	326	345	364	383	391	403	434	454	471	478	513	538

	724	726	728	730	760	769	780	806	809	811	871	912	916	920	924
	928	932	946	953	964	971	982	989	1000	1007	1046	1069	1072	1075	1097
	1155	1174	1193	1212	1220	1232	1255	1275	1292	1298	1301	1321	1341	1344	1351
	1355	1358	1479	1481	1542	1545	1562	1569	1572	1589	1649	1662	1667	1687	1727
	1738	1742	1807	1815	1836	1844	2094								
CMPB	543	663	671	673	830	839	841	897	1032	1351	1352	1356	1357	1358	1359
	1360	1758	1864	1873	1875	1899	1908	1912	1916	1931	1940	1944	1960	1968	1972
	1990														
COM	193														
COMB	1419														
DEC	648	817	1081	1350	1353	1638	1790	2087							
DEC8	1354	1356	2047												
ENT	23														
HALT	16	1352	1356	1362	1538										
INC	217	536	545	550	553	556	599	610	646	945	963	981	999	1045	1096
	1339	1350	1351	1352	1354	1355	1357	1362	1636	1686	1691	1713	1714	1793	2024
	2036	2048	2065	2075											
INCB	1351	1352	1356	1423	2035										
IOT	23														
JMP	16	18	19	20	21	200	203	208	209	254	328	347	366	385	405
	436	456	473	616	619	630	637	675	716	843	855	861	910	914	918
	922	926	930	934	958	976	994	1012	1024	1055	1106	1128	1157	1176	1195
	1214	1234	1257	1277	1294	1350	1376	1377	1378	1384	1385	1386	1492	1497	1517
	1520	1539	1606	1616	1764										
JSR	196	229	238	239	251	255	262	268	272	275	277	284	293	304	315
	320	321	333	339	340	352	358	359	371	377	378	394	397	398	423
	428	429	441	448	449	461	465	466	522	523	524	534	565	578	652
	679	695	702	708	782	824	863	864	943	944	961	962	979	980	997
	998	1015	1017	1019	1021	1043	1044	1095	1110	1112	1118	1131	1144	1149	1150
	1162	1168	1169	1181	1187	1188	1200	1206	1207	1223	1226	1227	1244	1249	1250
	1262	1269	1270	1282	1286	1287	1330	1331	1350	1352	1356	1357	1392	1504	1512
	1605	1615	1697	1763	1882	1883	1884	1885	1886	1887	1888	2020	2030	2042	2054
	2071														
MOV	182	185	186	187	188	211	212	213	214	215	226	227	228	236	237
	243	244	247	249	260	264	265	280	291	292	296	299	303	314	316
	319	324	330	334	335	338	343	353	354	356	362	372	373	375	381
	387	388	395	401	407	408	413	424	425	432	442	444	445	447	452
	458	462	464	469	475	476	480	486	493	494	495	496	497	498	512
	516	517	518	519	520	527	530	563	577	585	589	591	592	593	594
	595	596	597	609	627	632	633	643	645	647	662	682	683	684	685
	686	689	691	692	694	704	711	712	714	732	735	736	737	739	742
	745	747	758	765	766	776	777	787	788	789	791	800	802	829	852
	857	858	868	877	879	881	882	883	886	892	896	899	909	949	951
	967	969	985	987	1003	1005	1014	1016	1018	1020	1031	1034	1042	1049	1051
	1062	1064	1078	1094	1100	1102	1114	1121	1122	1123	1124	1143	1145	1148	1153
	1159	1163	1164	1167	1172	1182	1183	1185	1191	1201	1202	1204	1210	1216	1217
	1224	1230	1245	1246	1253	1263	1265	1266	1268	1273	1279	1283	1285	1290	1296
	1310	1311	1320	1324	1325	1326	1327	1328	1334	1337	1350	1351	1352	1353	1354
	1355	1356	1357	1358	1359	1360	1361	1362	1391	1393	1397	1398	1399	1402	1403
	1404	1420	1444	1445	1446	1448	1458	1464	1465	1466	1467	1468	1477	1484	1486
	1491	1493	1502	1510	1524	1526	1527	1531	1533	1553	1557	1559	1560	1580	1584
	1586	1587	1598	1599	1600	1601	1602	1608	1609	1610	1611	1612	1621	1622	1625
	1647	1664	1669	1693	1708	1710	1711	1716	1731	1736	1740	1744	1750	1751	1752
	1762	1768	1769	1770	1773	1774	1775	1776	1785	1795	1802	1831	1863	1880	1898

.LIST	2	16	23	24	181	182	237	1350	1351	1352	1358	1361			
.MACRO	12	24	1361												
.MCALL	4	5	6	7	8	23	24	182							
.NLIST	1	16	23	24	181	182	237	1350	1351	1352	1358	1361			
.PAGE	22	24	184	235	307	417	489	606	620	678	720	754	797	845	889
	941	1058	1109	1136	1238	1306	1349	1364	2348	2465	2466	2530			
.REPT	16	24													
.SBTTL	12	14	15	16	23	24	237	621	846	1350	1351	1352	1353	1354	1355
	1356	1357	1358	1359	1360	1361	1362	2130	2349	2467					
.TITLE	11														
.WORD	14	15	16	24	1350	1353	1354	1356	1357	1359	1360	1362	2359	2392	2403
	2407	2415	2426	2434	2462										

ERRORS DETECTED: 0

*DZDHN.A, DZDHN.A/CRF=DZDHN.A
RUN-TIME: 69 36 7 SECONDS
CORE USED: 23K

