DH11

DIAGNOSTIC

MD-11-DZDHM-B

EP-DZDHM-B-DL-A NOV 1976

FICHE 1 OF 2 MADE IN U.S

DH11

DIAGNOSTIC

MD-11-DZDHM-B

EP-DZDHM-B-DL-A NOV 1976

FICHE 2 OF 2 MADE IN USA

.REM *

SEQ 0001

PRODUCT CODE: MAINDEC-11-DZDHM-B
PRODUCT NAME: DH11 DIAGNOSTIC
DATE: 21 SEPTEMBER-1976
AUTHOR: E. CROWLEY
MAINTAINED BY: DIAGNOSTIC ENGINEERING

TABLE OF CONTENTS

1.0 GENERAL PROGRAM DESCRIPTION

1.1 PROGRAM PURPOSE

"MD-11-DZDHM" IS A COMPREHENSIVE DIAGNOSTIC TEST PROGRAM DESIGNED TO AID IN THE ACCEPTANCE TESTING, INSTALLATION CHECKOUT, AND CORRECTIVE MAINTENANCE OF THE DH11 16. LINE ASYNCHRONOUS SERIAL LINE MULTIPLEXOR. IT CONSISTS OF 44. LOGICALLY SEQUENCED DIAGNOSTIC TESTS DESIGNED TO TEST AND VERIFY THAT THE DH11 IS OPERATING IN ACCORDANCE WITH ITS DESIGN SPECIFICATIONS.

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

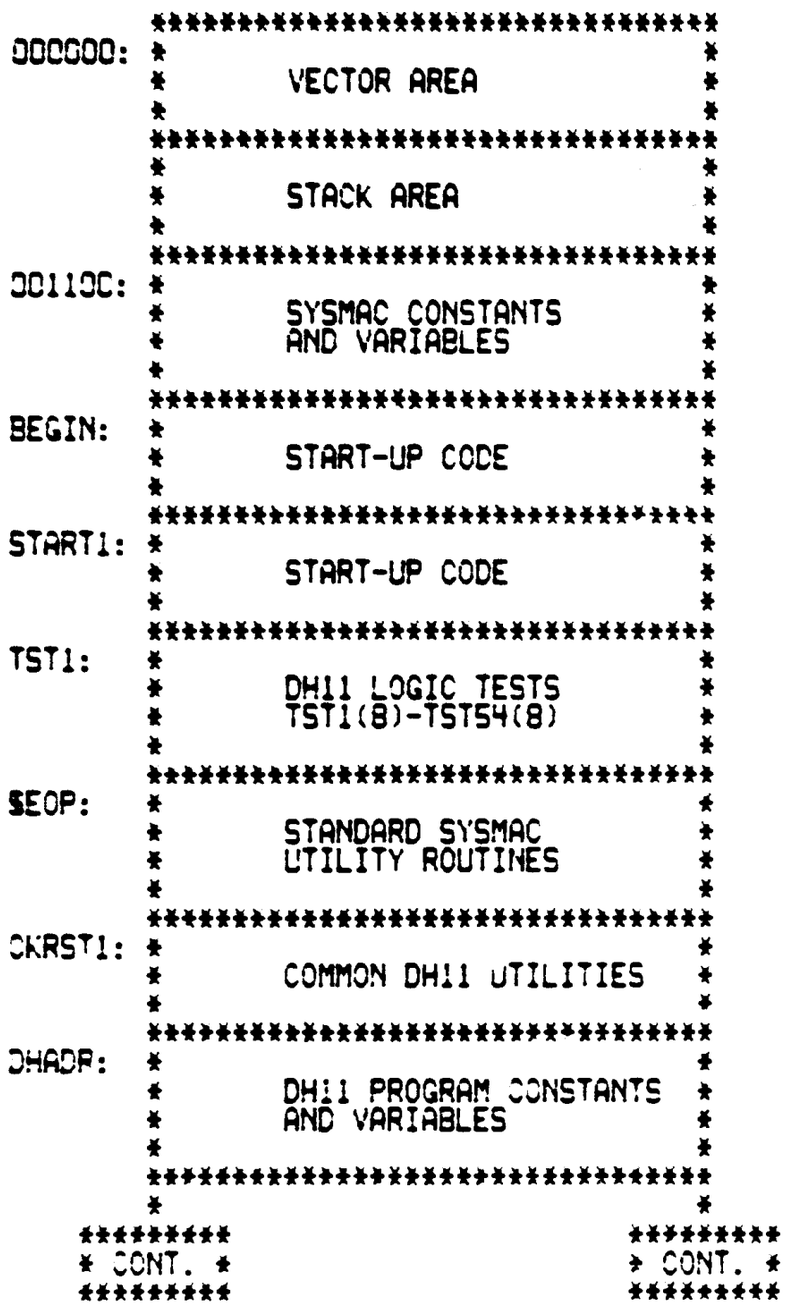
IN ORDER TO FACILITATE INSTALLATION CHECKOUT, TESTS 101, AND 105 THROUGH 107 (TEST GROUP 1) OF THE DM11-BB MODEM CONTROL DIAGNOSTIC, DZDHK, HAVE BEEN INCLUDED IN THIS PROGRAM. IN THIS WAY ALL THE LEVEL CONVERTERS AND CABLES CAN BE CHECKED WITH JUST ONE PROGRAM USING THE H315 TURNAROUND CONNECTOR. HOWEVER, RUNNING THESE TESTS REQUIRE OPREATOR INTERVENTION (SR10=1) OTHERWISE, THE TESTS ARE BYPASSED. (REFER TO SECTION 6.3)

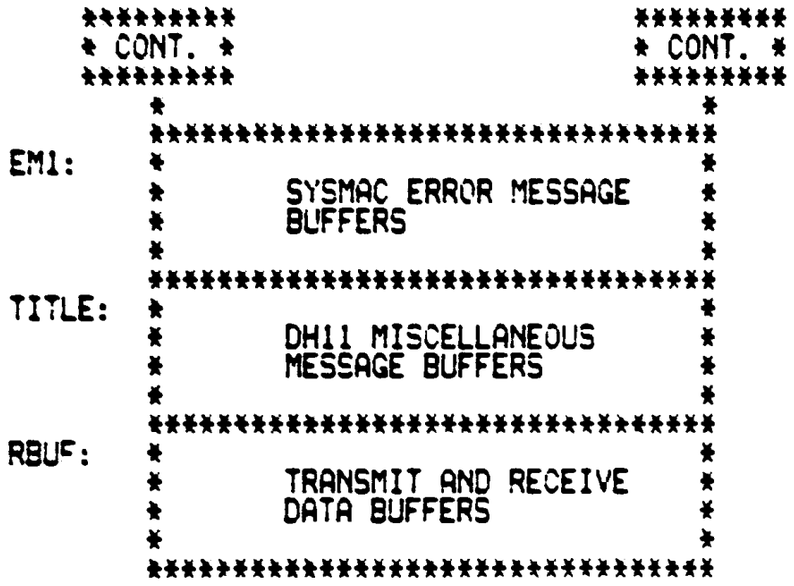
1.1.1 LOGIC TEST SUMMARY

SEQ 0005

```
T1      CHECK SSYN RESPONSE FROM ALL DH11 REGISTERS
T2      TEST THAT "MASTER CLR" CAN CLEAR THE "SCR","LPR","BKR",AND "SSR" REGS
T3      TEST "SCR" REG R W BITS CAN SET/CLR (NORMAL MODE)
T4      TEST "SCR" REG. READ ONLY BITS (NORMAL MODE)
T5      TEST "SCR" REG. BITS THAT CAN BE SET/CLR IN MAINT. MODE
T6      TEST THAT ALL R W BITS IN "LPR" CAN BE SET/CLR
T7      TEST THAT ALL R W BITS IN "BKR" CAN BE SET/CLR
T10     TEST THAT ALL R/W BITS IN "SSR" CAN BE SET/CLR
T11     TEST THAT CLR/SET OF BIT "N" IN "LPR" DOES NOT CLEAR ANY OTHER BITS
T12     TEST THAT CLR/SET OF BIT "N" IN "BKR" DOES NOT CLEAR ANY OTHER BITS
T13     TEST THAT CLR/SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER BITS
T14     "CAR" MEMORY ADDRESSING TEST
T15     "BCR" MEMORY ADDRESSING TEST
T16     "CAR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
T17     "BCR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
T20     "CAR" MEMORY PATTERNS TEST / 0'S DISTURB
T21     "BCR" MEMORY PATTERNS TEST / 0'S DISTURB
T22     "CAR" MEMORY PATTERNS TEST / 1'S DISTURB
T23     "BCR" MEMORY PATTERNS TEST / 1'S DISTURB
T24     TEST THAT "CAR" MEMORY EXT BITS SET/CLR PROPERLY
T25     TEST INTR. ENAB. BITS - INTR. CONDITION DISABLED
T26     TEST CHAR. AVAIL. I.E.  WITH INTR. CONDITION ACTIVE
T27     TEST SILO OVFLW.  I.E.  WITH INTR. CONDITION ACTIVE
T30     TEST NON EX MEM I.E.  WITH INTR. CONDITION ACTIVE
T31     TEST XMITTR DONE I.E.  WITH INTR. CONDITION ACTIVE
T32     BASIC TRANSMITTER "NPR" LOGIC TEST 1
T33     TRANSMITTR NPR LOGIC TEST 2
T34     TEST THAT CHARACTER AVAILABLE CAN CAUSE RCVR INTERRUPT
T35     TEST THAT THE SILO STATUS REG COUNTS UP CORRECTLY
T36     TEST THAT SILO STATUS REGISTER DOWN COUNTS CORRECTLY
T37     TEST SILO ALARM LEVEL FOR  COUNTS 0,1,2,4,8,16, AND 32
T40     TRANSMITTER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS
T41     RECEIVER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS
T42     VERIFY STORAGE OVERFLOW - NON MAINT MODE - ALL SELECTED LINES
T43     BASIC DATA TEST - ALL SELECTED LINES/ALL CHAR LENGTHS
T44     SINGLE LINE DATA TEST - ALL SELECTED LINES
T45     BASIC PARITY LOGIC TEST - ALL SELECTED LINES - ODD PARITY
T46     MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES
T47     AUTO ECHO TEST 1 - ALL SELECTED LINES
T50     AUTO ECHO TEST 2 - ALL SELECTED LINES
T51     AUTO ECHO TEST 3 - ALL SELECTED LINES
T52     BREAK BIT TEST - ALL SELECTED LINES
T53     HALF DUPLEX TEST - ALL SELECTED LINES
T54     VERIFY THAT OVERRUN CAN SET PROPERLY - ALL SELECTED LINES
T55     ABBREVIATED DM11-BB DIAGNOSTIC. (DZDHK T101)
T56     DM11-BB DIAGNOSTIC CONTINUED (DZDHK T105)
T57     DM11-BB DIAGNOSTIC CONTINUED (DZDHK T106)
T60     DM11-BB DIAGNOSTIC CONTINUED (DZDHK T107)
```

1.1.2 MD-11-DZDHM CORE MEMORY MAP




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

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

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

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

B. A DH11 16. LINE ASYNCHRONOUS SERIAL LINE MULTIPLEXOR

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

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

1.2.2 SOFTWARE REQUIREMENTS

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

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

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

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

A. CPU/CORE MEMORY DIAGNOSTICS

1.5 FAILURE ASSUMPTIONS

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

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

2.1.1 LOADING PROCEDURES

A. PAPER TAPE SYSTEMS

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

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

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

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

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

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

2.1.2 STARTING PROCEDURES

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

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

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

   SET THE SR=004000 (QUICK PASS)

   SET THE SR=002000 (TO PERFORM AN ABBREVIATED DM11-BB TEST. REFER TO SECTIONS 1.1 AND 6.3)

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

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

   NOTE: THE CSR REGISTER ADDRESS OF THE DM11-BB('S) IS LOADED ONLY FROM THE AUTOSIZER, HOWEVER, AFTER INITIAL LOAD, THE PROGRAM CAN BE STARTED AT 210(8) TO CHANGE SELECTION PARAMETERS AS DESCRIBED IN SECTION 2.1.2 D.

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

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

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

   SET T E SR=004000 (QUICK PASS)

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

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

   NOTES: IF THE DM11-BB VECTORS ARE INTERLEAVED WITH THE DH11 VECTORS (2040 FRONT END) THE DISPLACEMENT IS 20(8) ADDRESSES.

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

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

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

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

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

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

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

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

EXAMPLES:

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

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

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

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

EXAMPLES:

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

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

****
NOTE
****

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

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

    "DEPRESS CONTINUE TO STAPT TESTING"

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

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

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

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

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

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

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

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

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

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

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

SET THE SR=004000 (QUICK PASS)

SET THE SR=002000 (TO PERFORM AN ABBREVIATED DM11-BB TEST.
THIS ASSUMES THE AUTOSIZER WAS PREVIOUSLY
USED TO LOAD THE DM11-BB CSR ADDRESSES.)

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

6. SET THE HALT/ENABLE SWITCH TO ENABLE
7. DEPRESS START - THE PROGRAM TYPES THE TITLE AND THEN ASKS FOR DEVICE SELECTION PARAMETER. PROCEED AS IN (B-10) ABOVE.
9. PROGRAM WILL ASK FOR LINE SELECTION PARAMETERS. PROCEED AS IN (B-11) ABOVE.
   NOTE: THE DEVICE SELECTION AND LINE SELECTION PARAMETERS APPLY TO BOTH THE DH11 AND THE DM11-BB. THAT IS, IF DH #7, LINE #3 IS CHOSEN THEN DM11-BB #7 LINE #3 WILL ALSO BE TESTED.
10. IF SR8=1, THE PROGRAM WILL HALT AND PRINT THE FOLLOWING MESSAGE:

    "DEPRESS CONTINUE TO START TESTING"

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

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

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

2.2 SPECIAL ENVIRONMENTS

2.2.1 ACT11 APT11

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

2.2.2 XXDP

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

2.2.3 SWITCHLESS FEATURE

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

CONTROL:

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

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

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

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

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

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

2.3 PROGRAM OPTIONS

2.3.1 CONSOLE SWITCH REGISTER

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

SWITCH REGISTER	START	TESTING
15 = 1	-----	HALT ON ERROR (AFTER TYPING ERROR MESSAGE)
14 = 1	-----	LOOP CONTINUOUSLY ON CURRENT TEST.
13 = 1	-----	INHIBIT ERROR TYPOUTS.
11 = 1	-----	INHIBIT SUB-TEST ITERATIONS (QUICK PASS)
10 = 1	-----	PERFORM DM11-BB ABBREVIATED TESTS.
9 = 1	-----	LOCK ON HARD ERRORS
8 = 1	HALTS AFTER CONFIGURATION TO PERMIT DUMPING PRE-CONFIGURED COPIES OF THE PROGRAM.	SEARCH FOR AND LOCK ON TEST SELECTED BY CONTENTS OF SR <07:00>
<07:00>		CONTAINS TEST NUMBER TO SEARCH FOR WHEN SR 08 = 1
1 = 1	TYPES DEVICE MAP GENERATED BY THE AUTOSIZER.	-----
0 = 1	ALLOWS THE USER TO INPUT DH PARAMETERS MANUALLY. (INHIBITS THE AUTOSIZER)	-----

2.3.2 CORE MEMORY LOCATIONS

SEQ 0016

A. DH11 CONFIGURATION TABLES AND VARIABLES

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

1. DHADTB: 16. WORD DEVICE ADDRESS TABLE

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

2. DHVCTB: 16. WORD DEVICE VECTOR ADDRESS TABLE

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

3. BRLVL: 16. WORD BR LEVEL TABLE

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

4. DHSEL: DEVICE SELECTION PARAMETER

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

5. LINSEL: LINE SELECTION PARAMETER

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

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

NOTE: IT IS RECOMMENDED THAT WHEN NON-STANDARD CONFIGURATIONS ARE ENCOUNTERED, THE DM11-BB MODEM CONTROL DIAGNOSTIC, DZDHK, SHOULD BE RUN, RATHER THAN ALTERING THE DM11-BB TABLES IN THIS PROGRAM.

E02

B. SUB-TEST ITERATION COUNT

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

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

2.3.3 REGISTER USAGE

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

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

2.4 EXECUTION TIMES

A. SR11 = 0 SUB-TEST ITERATIONS

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

B. SR11 = 1 INHIBIT ITERATIONS

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

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

3.0 ERROR INFORMATION

3.1 ERROR REPORTING PROCEDURES

3.1.1 STANDARD SYSMAC.SML ERROR REPORTING CONVENTIONS

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

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

EXAMPLE:

```
SYSTEM CONTROL REGISTER ERROR
 (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B
002720  000002  001074  000003  160020  160020  000000  000001
```

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

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

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

THERE ARE ONLY TWO MESSAGES IN THE DM11-BB PORTION OF THIS PROGRAM:
ONE INFORMS THE USER THAT NO DM11-BB'S WERE FOUND BY THE AUTOSIZER AND THE PROGRAM THEN CONTINUES TESTING THE DM11'S. THE OTHER INSTRUCTS THE USER TO RUN THE DM11-BB MODEM CONTROL DIAGNOSTIC, DZDHK, DUE TO AN ERROR. THE PROGRAM THEN HALTS.

3.1.2 ERROR MESSAGE TABLES

SEQ 0020

```
;ERROR TABLE ITEM FOR ERROR MESSAGE 1

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 2

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 3

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 4

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 5

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 6

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 7

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

```
;ERROR TABLE ITEM FOR ERROR MESSAGE 10

        EM10            ;"BYTE COUNTER REGISTER ERROR - LINE #XX"
        DH2             ;"  (PC)   (PS)   (SP)   TEST   DEVADR   REGADR   WAS   S/B "
        DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 11

        EM11            ;"UNEXPECTED DH11 RCVR INTERRUPT"
        DH2             ;"  (PC)   (PS)   (SP)   TEST   DEVADR   REGADR   WAS   S/B "
        DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 12

        EM12            ;"UNEXPECTED DH11 XMITTR INTERRUPT"
        DH2             ;"  (PC)   (PS)   (SP)   TEST   DEVADR   REGADR   WAS   S/B "
        DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 13

        EM13            ;"CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT"
        DH2             ;"  (PC)   (PS)   (SP)   TEST   DEVADR   REGADR   WAS   S/B "
        DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 14

        EM14            ;"TRANSMITTER NPR LOGIC ERROR - LINE #  "
        DH2             ;"  (PC)   (PS)   (SP)   TEST   DEVADR   REGADR   WAS   S/B "
        DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 15

        EM15            ;"XMITTR FAILED TO INTERRUPT - LINE #  "
        DH2             ;"  (PC)   (PS)   (SP)   TEST   DEVADR   REGADR   WAS   S/B "
        DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 16

        EM16            ;"RCVR FAILED TO INTERRUPT"
        DH2             ;"  (PC)   (PS)   (SP)   TEST   DEVADR   REGADR   WAS   S/B "
        DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL
```

```
;ERROR TABLE ITEM FOR ERROR MESSAGE 17

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 20

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 21

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 22

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 23

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


;ERROR TABLE ITEM FOR ERROR MESSAGE 24

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 25

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

```
;ERROR TABLE ITEM FOR ERROR MESSAGE 26

        EM26            ;"HALF-DUPLEX TEST ERROR - LINE #  "
        DH2             ;"  (PC)   (PS)   (SP)   TEST   DEVADR   REGADR   WAS   S/B "
        DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 27

        EM27            ;"UNEXPECTED BUS ERROR TRAP"
        DH3             ;  (PC)   (PS)   (SP)   TEST   TRPPC   TRPPS
        DT3             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 30

        EM30            ;"UNEXPECTED RSVD INSTR TRAP"
        DH3             ;  (PC)   (PS)   (SP)   TEST   TRPPC   TRPPS
        DT3             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2
        0               ;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 31

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 32

        EM32            ;"AUTO ECHO TEST TIMEOUT - LINE #  "
        DH5             ;"  (PC)   (LPRG)    TEST""
        DT4             ;$ERRPC,$TMP0,$TMP2
        0               ;PRINT ALL OCTAL


;ERROR TABLE ITEM FOR ERROR MESSAGE 33

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 34

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

```
;ERROR TABLE ITEM FOR ERROR MESSAGE 35

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 36

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 37

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 40

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 41

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 42

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 43

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

```
;ERROR TABLE ITEM FOR ERROR MESSAGE 44

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 45

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 46

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 47

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 50

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 51

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 52

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

```
;ERROR TABLE ITEM FOR ERROR MESSAGE 53

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 54

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 55

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 56

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

;ERROR TABLE ITEM FOR ERROR MESSAGE 57

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

3.1.3 DATA HEADER MNEUMONIC DEFINITIONS

ALL NUMBERS PRINTED AS ERROR DATA ARE IN OCTAL

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

3.2 POWER FAIL PRINTOUT

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

"POWER"

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

3.3 ERROR HALTS

A. SYSMAC ERROR SERVICE ROUTINE HALT

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

B. POWER FAIL HALT

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

C. TRAP CATCHER HALTS

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

VN / VN+2
VN+2/ HALT

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

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

(NONE PROVIDED)

4.2 PROGRESS REPORTS

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

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

WHERE "X" IS THE REVISION LEVEL LETTER DESIGNATION.

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

"TESTING DH11 #NN"

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

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

"END PASS #XXXXX"

WHERE "X" IS THE PASS COUNT IN DECIMAL.

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

"DEPRESS CONTINUE TO START TESTING"

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

5.0 DH11 DEVICE INFORMATION

SEQ 0030

5.1 ADDRESS AND VECTOR ASSIGNMENTS

THE DH11 USES FLOATING ADDRESSES AND IS LOCATED AFTER DJ11'S IN THE FLOATING ADDRESS SPACE THAT BEGINS AT LOCATION 760010. BECAUSE THE DH11 HAS EIGHT REGISTERS, IT MUST BE ASSIGNED AN ADDRESS THAT IS A MULTIPLE OF 20 (OCTAL). ALL DH11'S IN A SYSTEM SHOULD HAVE CONSECUTIVE ADDRESSES.

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

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

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

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

THE DH11 VECTORS (2) FOLLOW THOSE OF THE DJ11 IN THE FLOATING VECTOR SPACE THAT STARTS AT ADDRESS 300. THE VECTORS STARTING AT 300 ARE USED IN THE FOLLOWING ORDER: DC11; KL11/DL11-A, B; DP11; DM11-A; DN11; DM11-BB; DR11-A; DR11-C; PA611 READERS; PA611 PUNCHES; DT11; DX11; DL11-C, D, E; DH11.

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

5.2 REGISTER DEFINITION

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

5.2.1 THE SYSTEM CONTROL REGISTER - ADDRESS X00

SEQ 0031

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 S. 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 INTERPUPT REQUEST. THIS BIT IS READ/WRITE.

13 TRANSMITTER AND NON-EX-MEM INTERRUPT ENABLE

SEQ 0032

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

14 STORAGE INTERRUPT

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

15 TRANSMITTER INTERRUPT

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

5.2.2 NEXT RECEIVED CHARACTER REGISTER ADDRESS X02

SEQ 0033

BITS	DESCRIPTION
00-07	NEXT RECEIVED CHARACTER THESE BITS CONTAIN THE NEXT RECEIVED CHARACTER, RIGHT JUSTIFIED. THE LEAST SIGNIFICANT BIT IS BIT 00.
08-11	LINE NUMBER THESE BITS INDICATE THE LINE NUMBER ON WHICH THE NEXT RECEIVED CHARACTER WAS RECEIVED. BIT 8 IS THE LEAST SIGNIFICANT BIT.
12	PARITY ERROR THIS BIT IS SET IF THE PARITY OF THE RECEIVED CHARACTER DOES NOT AGREE WITH THAT DESIGNATED FOR THAT LINE.
13	FRAMING ERROR THIS BIT IS SET IF THE RECEIVER SAMPLES A LINE FOR THE FIRST STOP BIT, AND FINDS THE LINE IN A SPACING CONDITION (LOGICAL 0). THIS CONDITION USUALLY INDICATES THE RECEPTION OF A BREAK.
14	DATA OVERRUN THIS BIT IS SET WHEN THE RECEIVED CHARACTER WAS PRECEDED BY A CHARACTER THAT WAS LOST DUE TO THE INABILITY OF THE RECEIVER SCANNER TO SERVICE THE UART RECEIVER HOLDING BUFFER. REFER TO THE SECTION ON PROGRAMMING FOR FURTHER DETAILS ON DOUBLE-BUFFERED RECEPTION.
15	VALID DATA PRESENT THIS BIT INDICATES THAT THE DATA PRESENTED IN BITS 14-00 IS VALID. IT PERMITS A CHARACTER HANDLING PROGRAM TO TAKE CHARACTERS FROM THE SILO UNTIL IT IS EMPTY. THIS IS DONE BY READING THIS REGISTER AND CHECKING BIT 15 UNTIL A WORD IS OBTAINED FOR WHICH BIT 15 IS A ZERO. THE ENTIRE NEXT RECEIVED CHARACTER REGISTER IS READ-ONLY AND IS ADDRESSABLE ONLY ON A WORD BASIS.

5.2.3 LINE PARAMETER REGISTER ADDRESS X04

SEQ 0034

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

02 TWO STOP BITS

THIS BIT, WHEN SET, CONDITIONS A LINE TRANSMITTING WITH 6, 7, OR 8-BIT CODE TO TRANSMIT CHARACTERS HAVING TWO STOP MARKS. IF THE LINE IS TRANSMITTING 5-BIT CODE, ASSERTION OF THIS BIT CAUSES THE CHARACTERS TO BE TRANSMITTED WITH 1.5 STOP MARKS. IF THIS BIT IS NOT ASSERTED, 1 STOP MARK IS SENT.

03 NOT USED

04 PARITY ENABLED

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

05 ODD PARITY

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

06-09 RECEIVER SPEED

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

10-13 TRANSMITTER SPEED

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

SEQ 0035

SPEED TABLE FOR RECEIVER AND TRANSMITTER SPEEDS:

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

14 HALF DUPLEX/FULL DUPLEX

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

SEQ 0036

THIS REGISTER SHOULD BE LOADED ONLY AFTER THE SYSTEM CONTROL REGISTER (SCR) HAS HAD THE APPROPRIATE BITS SET TO SELECT THE DESIRED LINE NUMBER. WHEN THIS REGISTER IS LOADED, ADDRESS BITS 00-15 ARE TRANSFERRED INTO SEMICONDUCTOR MEMORIES IN THE DH11 FROM BITS 00-15 OF THIS REGISTER. ADDRESS BITS 16-17 ARE TRANSFERRED INTO SEMICONDUCTOR MEMORIES IN THE DH11 FROM BITS 4-5 OF THE SYSTEM CONTROL REGISTER.

INTERRUPTS MUST BE INHIBITED OR THE SCR SAVED BETWEEN THE SETTING OF THE SCR BITS 0-3 AND THE READ OR WRITE OF THE CURRENT ADDRESS REGISTER.

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

5.2.5 BYTE COUNT REGISTER ADDRESS X10

IN THE SAME FASHION AS THE LINE PARAMETER AND CURRENT ADDRESS REGISTERS, THIS REGISTER SHOULD NOT BE LOADED OR READ WITHOUT FIRST SELECTING A LINE NUMBER BY MEANS OF THE LOWER-ORDER FOUR BITS OF THE SYSTEM CONTROL REGISTER. THIS REGISTER SHOULD BE LOADED WITH THE TWO'S COMPLEMENT OF THE NUMBER OF CHARACTERS (BYTES) TO BE TRANSMITTED ON THAT LINE. THE BYTE COUNT REGISTER IS READ/WRITE.

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

5.2.6 BUFFER ACTIVE REGISTER (BAR) ADDRESS X12

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

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

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

5.2.7 BREAK CONTROL REGISTER ADDRESS X14

THIS REGISTER CONTAINS ONE BIT FOR EACH LINE. SETTING A BIT IN THIS REGISTER WILL IMMEDIATELY GENERATE A BREAK CONDITION ON THE LINE CORRESPONDING TO THAT BIT NUMBER. CLEARING THE BIT WILL TERMINATE THE BREAK CONDITION. THE BREAK CONDITION MAY BE TIMED BY SENDING CHARACTERS DURING THE BREAK INTERVAL, SINCE THESE CHARACTERS WILL NEVER ACTUALLY REACH THE LINE. FURTHER COMMENTS CONCERNING THE TRANSMISSION OF BREAK SIGNALS MAY BE FOUND IN THE BREAK SIGNALS SECTION.

5.2.8 SILO STATUS REGISTER ADDRESS X16

SEQ 0037

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 FUNCTIONAL LOGIC PARTITIONING

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

M7277 CURRENT ADDR REG MEMORY AND ADDR SELECT

SH3 CONTROL STROBE MUX FOR THE "LPR" REGISTER

TRANSMITTER DATA MUX WITH AUTO-ECHO CONTROL LOGIC SELECTION

MSYN / SSYN TIMING CHAIN

DH11 MASTER CLEAR LOGIC

SH4: UNIBUS ADDRESS SELECTION LOGIC WITH JUMPERS

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

SH5: BYTE COUNT AND CURRENT ADDRESS MEMORY WRITE TIMING LOGIC

CURRENT ADDRESS MEMORY LOGIC BITS<17:08>

UNIBUS ADDRESS DRIVERS BITS <17:08>

SH6: BYTE COUNT AND CURRENT ADDRESS MEMORY ADDRESS SELECT MULTIPLEXOR

CURRENT ADDRESS MEMORY LOGIC FOR BITS <07:00>

UNIBUS ADDRESS DRIVERS FOR BITS <07:00>

TRANSMITTER EVEN/ODD BYTE DATA MULTIPLEXOR

M7279 FIFO BUFFER

SH1: INPUT DATA MULTIPLEXOR FOR SILO MEMORY

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

SILO MEMORY READ/WRITE TIMING LOGIC

"SSR" REGISTER BITS <13:08>

SILO ALARM LEVEL COMPARATOR

RECEIVED "DATA READY" STATUS FLAG

M7288 LINE PARAMETER CONTROL
****

SEQ 0039

SH3: CLOCK TIMING SIGNAL BUFFERS

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

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

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

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

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

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

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

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

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

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

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

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

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

SH2: UART CHIPS BIT<1:0>

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

SH3: UART CHIPS BIT<3:2>

SH4: UART CHIPS BIT<5:4>

RECEIVER SCAN MULTIPLEEXORS (MASTER FE AND MASTER PE)

SH5: UART CHIPS BIT<7:6>

TRANSMITTER SCAN MULTIPLEXOR

-12 VOLT DC REGULATOR

M7288 #2 MULTIPLE UART CARD FOR LINES (8-15)
****

SEQ 0040

SH2: UART CHIPS BIT(9:8)

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

SH3: UART CHIPS BIT(11:10)

SH4: UART CHIPS BIT(13:12)

RECEIVER SCAN MULTIPLEEXORS (MASTER FE AND MASTER PE)

SH5: UART CHIPS BIT(15:14)

TRANSMITTER SCAN MULTIPLEXOR

-12 VOLT DC REGULATOR

M7289 SYSTEM CONTROL AND RCVR SCAN
****

SH3: TRANSMITTER AND RECEIVER SCAN LOGIC

SH4: TRANSMITTER AND RECEIVER SCAN TIMING

SH5: HALF-DUPLEX CONTROL LOGIC

UART DATA MULTIPLEXORS

SH6: SYSTEM CONTROL REGISTER

M9540 DC11 - DH11 CLOCK
****

SH1: CRYSTAL OSCILLATOR AND FREQUENCY DIVIDERS

M796 UNIBUS MASTER CONTROL
****

SH1: NPR CONTROL LOGIC

M7281 #1 INTERRUPT CONTROL
****

"A" SECTION: RECEIVER INTERRUPT CONTROL LOGIC

"B" SECTION: TRANSMITTER INTERRUPT CONTROL LOGIC

M7281 #2 NPR CONTROL
****

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

"B" SECTION: (NOT USED)

M727S REGISTER AND BYTE CONTROL
****

SH3: CLEAR "BAR" REG MULTIPLEXOR

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

UNIBUS RECEIVERS BIT<15:08>

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

UNIBUS DATA RECEIVERS BIT<07:0C>

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

SH5:	BIT<15:12>	SH7:	BIT<07:04>
SH6:	BIT<11:09>	SH8:	BIT<03:00>

5.4 DH11 MODULE ALLOCATION CHART

SEQ 0042

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 9	UNIBUS CONNECTOR (NOTE #3)	NPR CNTL	REG 8 BYTE CNT	CURRENT ADDRS 8 ADDRS	SYSTEM CNTL 8 RCV SCAN	INTR CNTL	PRIORITY SELECTOR (NOTE #9)	LINE PARAMETER CNTL	UNIBUS CONNECTOR (NOTES #1, 8 #2)
		M796				M405	M971		
							CABLE		
8		UNIBUS MASTER CNTL				EXTERNAL B CLOCK (NOTE #5)	DATA CABLE (NOTES #6 8 #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 #8							EXTERNAL A CLOCK (NOTE #5)
	M7821								M4540
F	* INTR CNTL (NOTE #7)								DH11 DC11 CLOCK

FIGURE 2-4 DH11 MODULE UTILIZATION DIAGRAM
PAGE 2

E04

SEQ 0043

NOTES:

1. IF END OF BUS, REPLACE M920 WITH M930.
2. IF LAST UNIT IN BASIC BOX, REPLACE M920 WITH BC11A CABLE WHEN EXPANDING TO PERIPHERAL BOX.
3. IF FIRST UNIT IN EXPANDER BOX, REPLACE M920 WITH BC11A CABLE.
4. E02 MUST BE G727 GRANT CONTINUITY IF MODEM CONTROL MODULE SET IS NOT INSTALLED. * DENOTES DM11-BB MODEM CONTROL OPTION, WITH DH11-AA OR AC.
5. MODULE SLOTS PROVIDE FOR ADDITIONAL CLOCK RATES.
6. FOR DIAGNOSTIC CHECKOUT OF DH11-AA, AB, OR AC, REPLACES M971 WITH M974.
7. THIS SLOT CONTAINS MODEM CONTROL MODULE M7807 WITH DH11-AD.
8. THIS SLOT CONTAINS MODEM CONTROL MODULE M7808 WITH DH11-AD.
9. THIS SLOT CONTAINS EIA CONVERTER AND PRIORITY MODULE M5906 FOR DH11-AD OR AE.

6.0 MAINTENANCE PROCEDURES

6.1 INTRODUCTION

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

6.2 PRELIMINARY CHECKS

A. VISUAL INSPECTION

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

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

B. POWER CHECKS

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

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

NOTES: USE THE BLACK WIRE FOR GROUND REFERENCE

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

6.3 MAINTENANCE CONNECTORS

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

A. DH11-AA, AB, OR AC CONFIGURATIONS

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

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

2) TESTING ALL 16. LINE[illegible]CLUDING DATA CABLES WHICH CONNECT TO DIS[illegible]BUTION PANEL. DOES NOT TEST LEVEL CONVERTER CIRCUITS LOCATED IN DISTRIBUTION PANEL.

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

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

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

B. DH11-AD CONFIGURATION

1. TESTING ALL 16. LINES WITHOUT DATA CABLES

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

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

2. TESTING ONE OR MORE SINGLE LINES INCLUDING DATA CABLES

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

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

NOTE: TO TEST THE DEVICE CABLE AS WELL, INSTALL THE H315 TEST CONNECTOR AT THE END OF THE DEVICE CABLE AND LEAVE THE DEVICE CABLE CONNECTED TO THE DISTRIBUTION PANEL. WHEN RUN IN THIS CONFIGURATION THE OPERATOR CAN SET THE SR=002000, THEREBY ENABLING THE PROGRAM TO PERFORM AN ABBREVIATED DM11-BB TEST. IN THIS WAY ALL THE LEVEL CONVERTERS AND CABLES CAN BE CHECKED WITH JUST ONE PROGRAM.

6.4 COMPLETE DH11 SUBSYSTEM CHECKOUT

SEQ 0046

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

A. MD-11-DZDHM DH11 DIAGNOSTIC

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

B. MD-11-DZDHN DATA RELIABILITY TEST

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

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

C. SYSTEMS EXERCISER "DHAX" EXERCISER MODULE

WHERE "X" DESIGNATES THE REVISION LEVEL IN USE.

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

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

6.5 MAINTENANCE HEADER DESCRIPTION

SEQ 0047

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

TEST ABSTRACT:	THIS IS A CAPSULE SUMMARY OF WHAT THE TEST IS DESIGNED TO TEST AND HOW IT OPERATES.
ERRORS:	LISTS THE PARTICULAR ERROR CALLS INVOKED BY THE TEST WHEN A FAULT IS DETECTED. (REFER TO PARA 3.1.2 AND 3.1.3 FOR A DETAILED DESCRIPTION OF THE ERROR CALL INFORMATION)
SYNC:	LISTS ONE OR MORE SIGNALS THAT MAY BE USED TO SYNCHRONIZE THE OSCILLOSCOPE WHEN AN ERROR LOOP IS ESTABLISHED (SR09=1). FOR (H) SIGNALS USE (-) SLOPE TO TRIGGER ON THE TRAILING EDGE OF THE SIGNAL AND FOR (L) SIGNALS USE (+) SLOPE TO TRIGGER ON THE TRAILING EDGE.
DEBUG:	CONTAINS SUGGESTIONS OF THINGS TO CHECK AND WHERE POSSIBLE THE MOST PROBABLE MODULE IS GIVEN.
KEY LOGIC:	CONTAINS A LIST OF LOGIC SIGNALS AND/OR LOGIC COMPONENTS WITH MODULE NAMES AND PRINT NUMBERS TO RELATE THE TEST ROUTINE FUNCTION TO THE FUNCTIONAL AREAS OF LOGIC WITHIN THE PRINTS. WHERE POSSIBLE SIGNAL PIN NOS. ARE LISTED.

EXAMPLE: MAINTENANCE HEADER FOR TEST 1
****

TEST ABSTRACT:
****

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

ERRORS:
****

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

SYNC: (NONE)
*****

DEBUG:
******

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

KEY LOGIC:
****

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

!

FLOW CHART
****

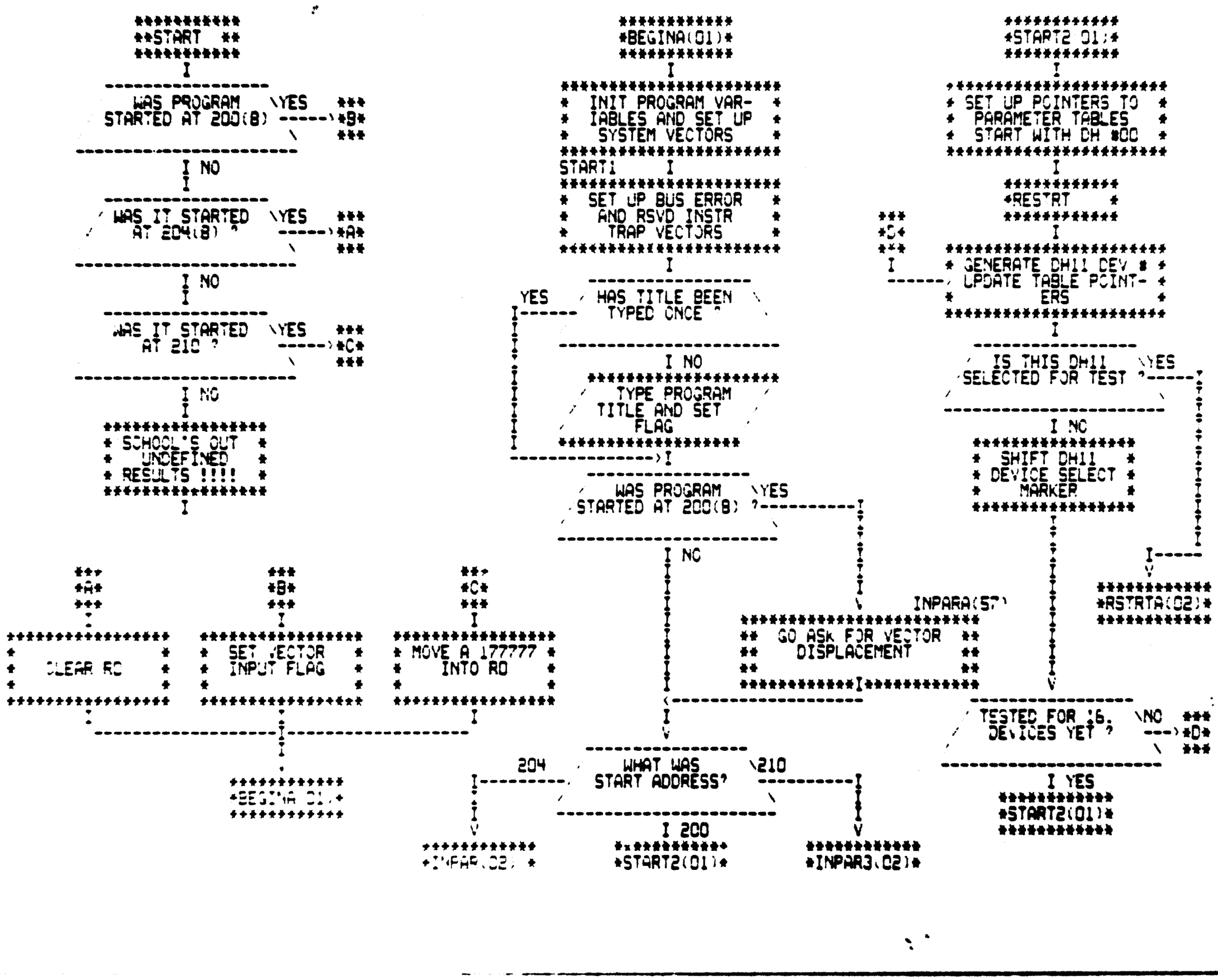
MD-11-DZDHM-B DH11 DIAGNOSTIC

****

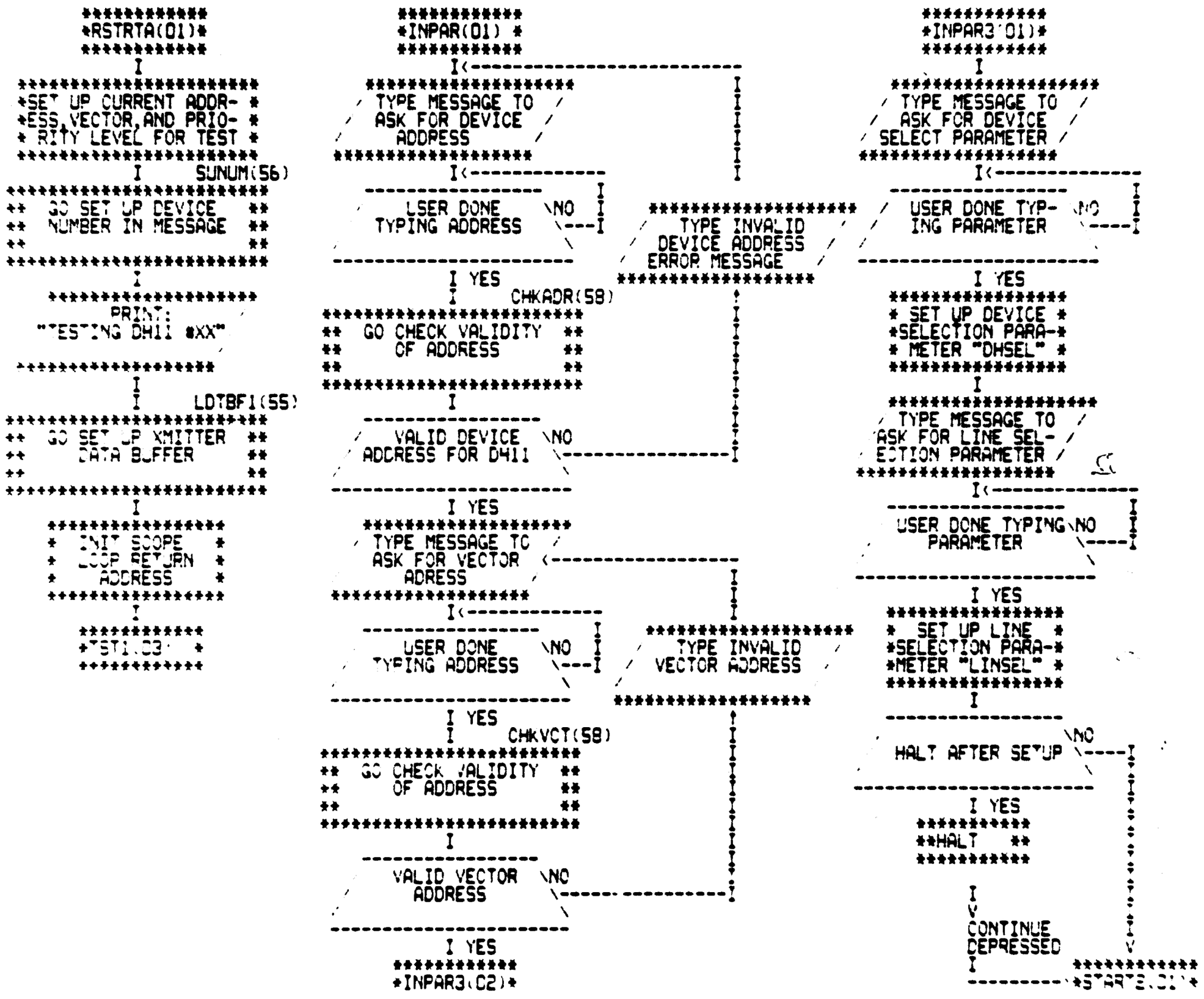
TABLE OF CONTENTS
****

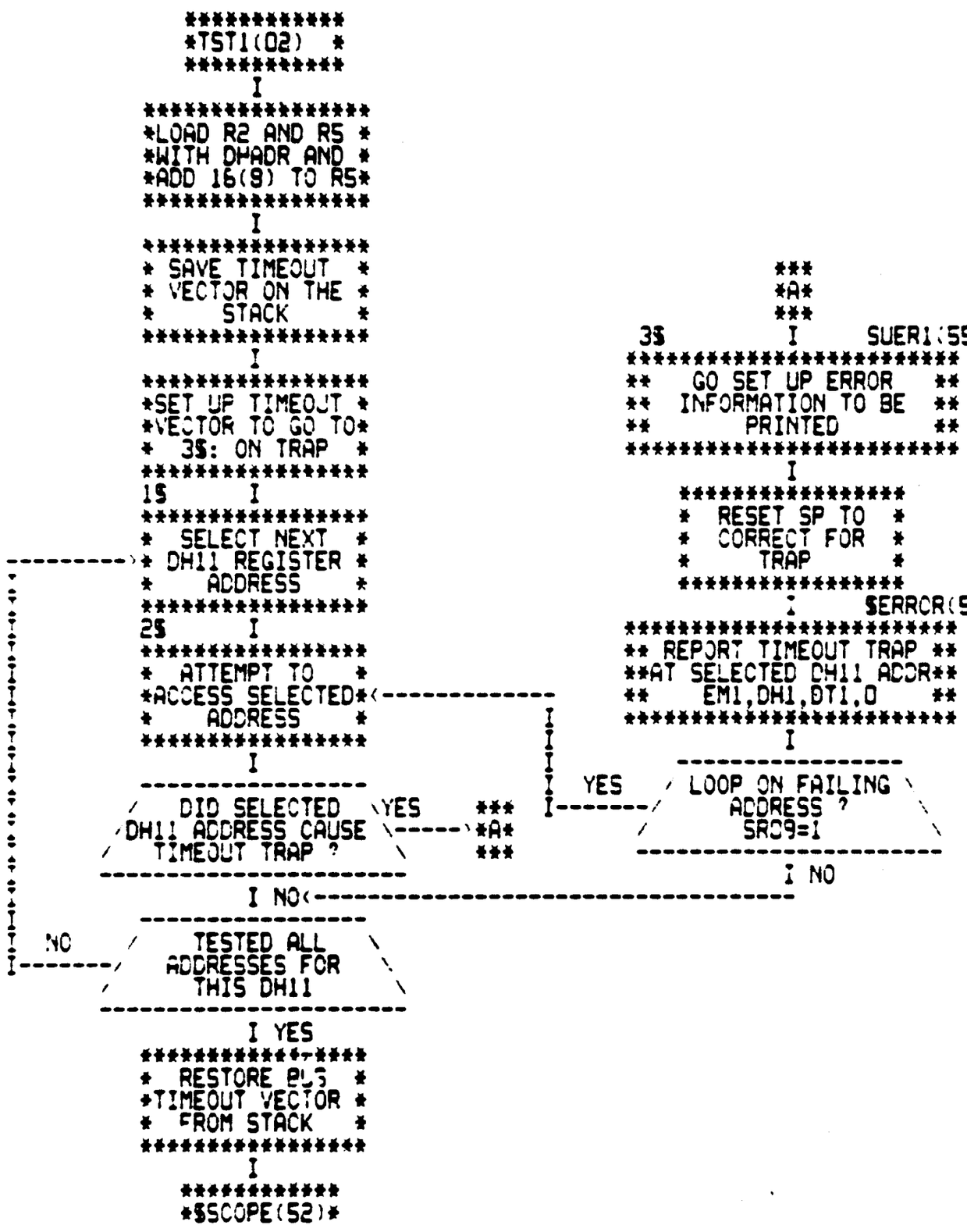
TABLE OF CONTENTS
****

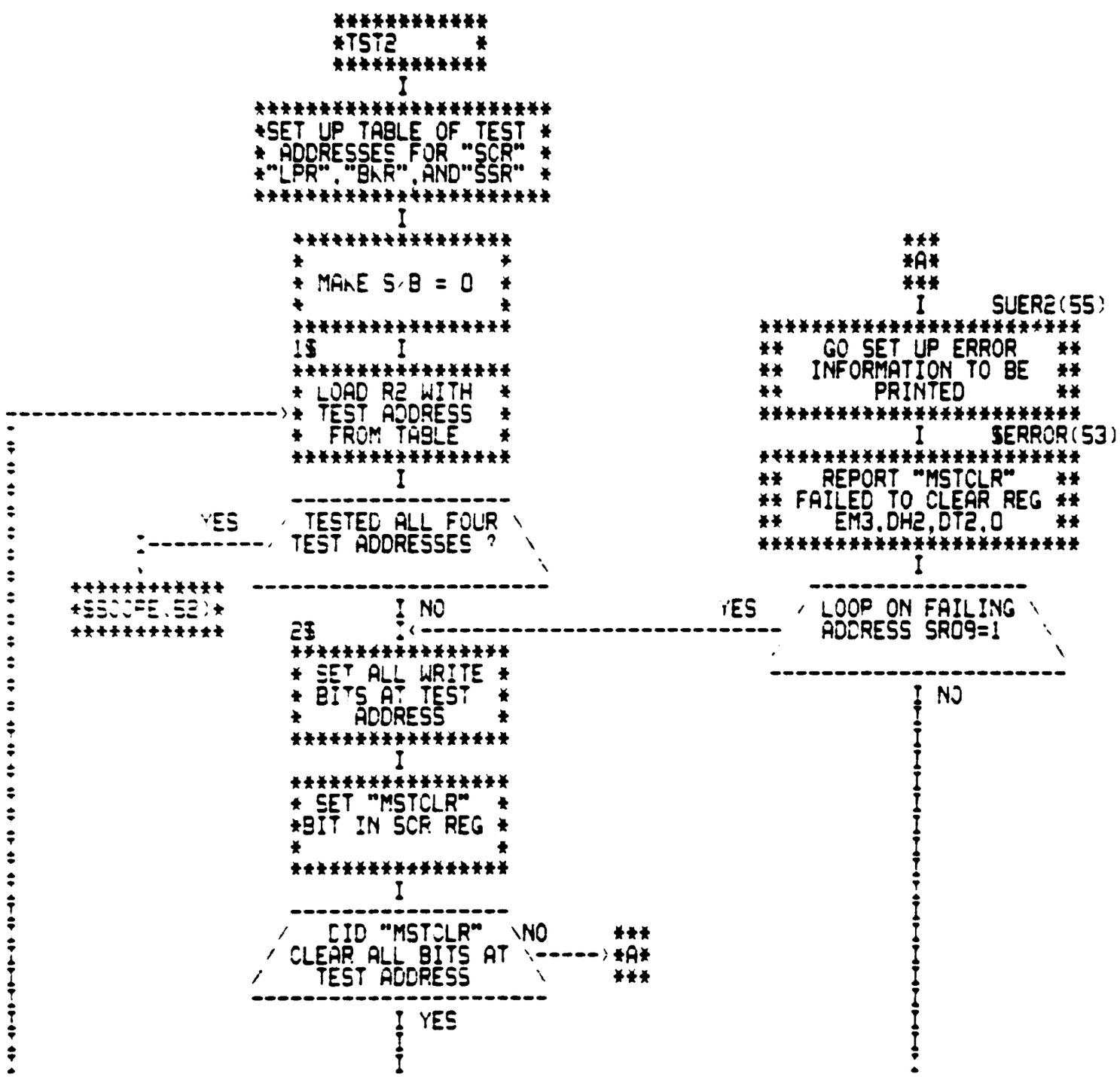
TABLE OF CONTENTS
****

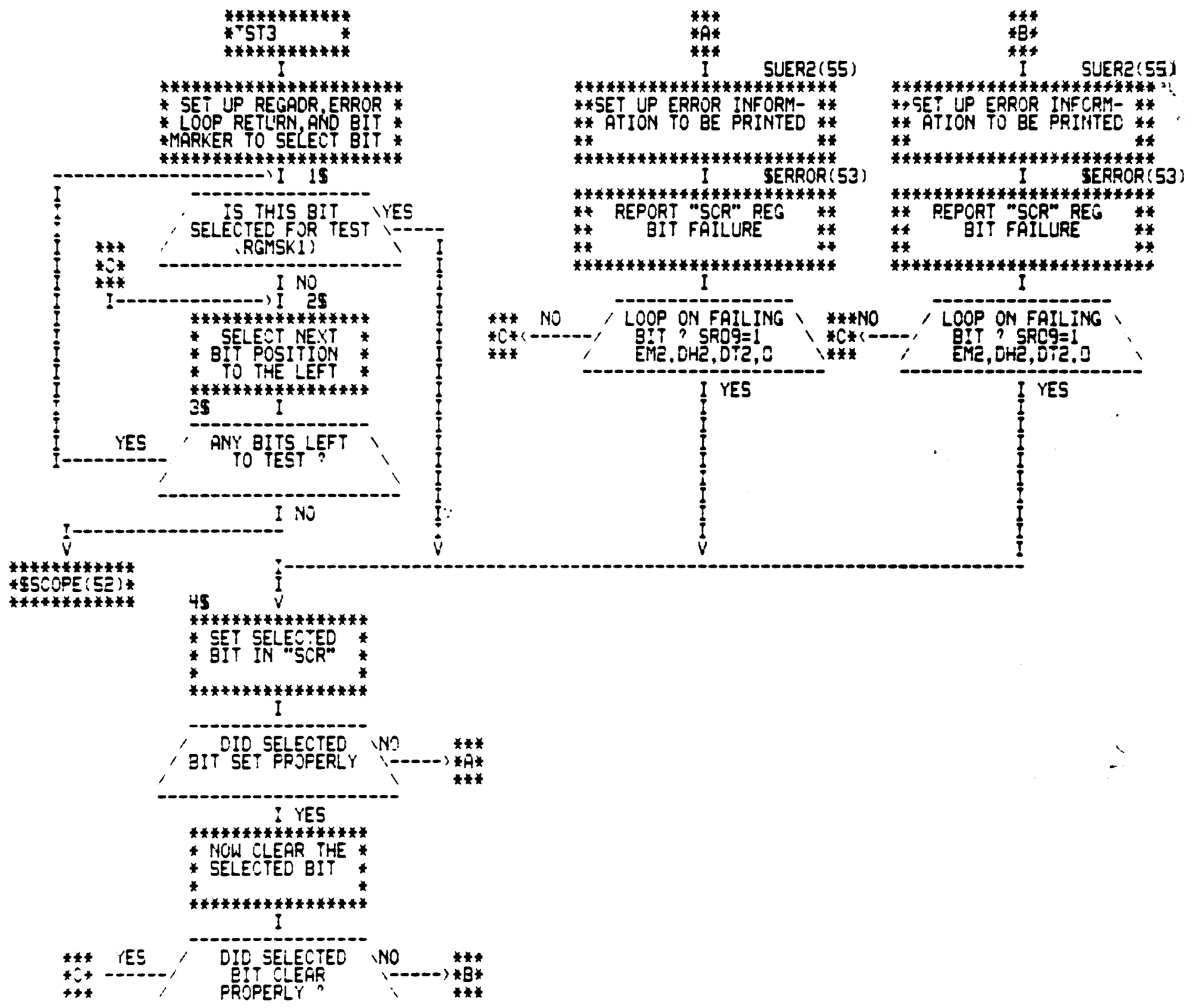

**START **
WAS PROGRAM STARTED AT 200(8)
YES
B
NO
WAS IT STARTED AT 204(8) ?
YES
A
NO
WAS IT STARTED AT 210 ?
YES
C
NO
SCHOOL'S OUT
UNDEFINED
RESULTS !!!!
A
CLEAR R0
B
SET VECTOR INPUT FLAG
C
MOVE A 177777 INTO R0
BEGINA(01)
INIT PROGRAM VARIABLES AND SET UP SYSTEM VECTORS
START1
SET UP BUS ERROR AND RSVD INSTR TRAP VECTORS
HAS TITLE BEEN TYPED ONCE ?
YES
NO
TYPE PROGRAM TITLE AND SET FLAG
WAS PROGRAM STARTED AT 200(8) ?
YES
NO
WHAT WAS START ADDRESS?
204
210
200
INPAR(02)
BEGINA(01)
START2(01)
INPARA(57)
GO ASK FOR VECTOR DISPLACEMENT
INPAR3(02)
START2 01:
SET UP POINTERS TO PARAMETER TABLES START WITH DH #00
RESTRT
GENERATE DH11 DEV # & UPDATE TABLE POINTERS
IS THIS DH11 SELECTED FOR TEST ?
YES
NO
SHIFT DH11 DEVICE SELECT MARKER
RSTRTA(02)
TESTED FOR 16. DEVICES YET ?
NO
D
YES
START2(01)

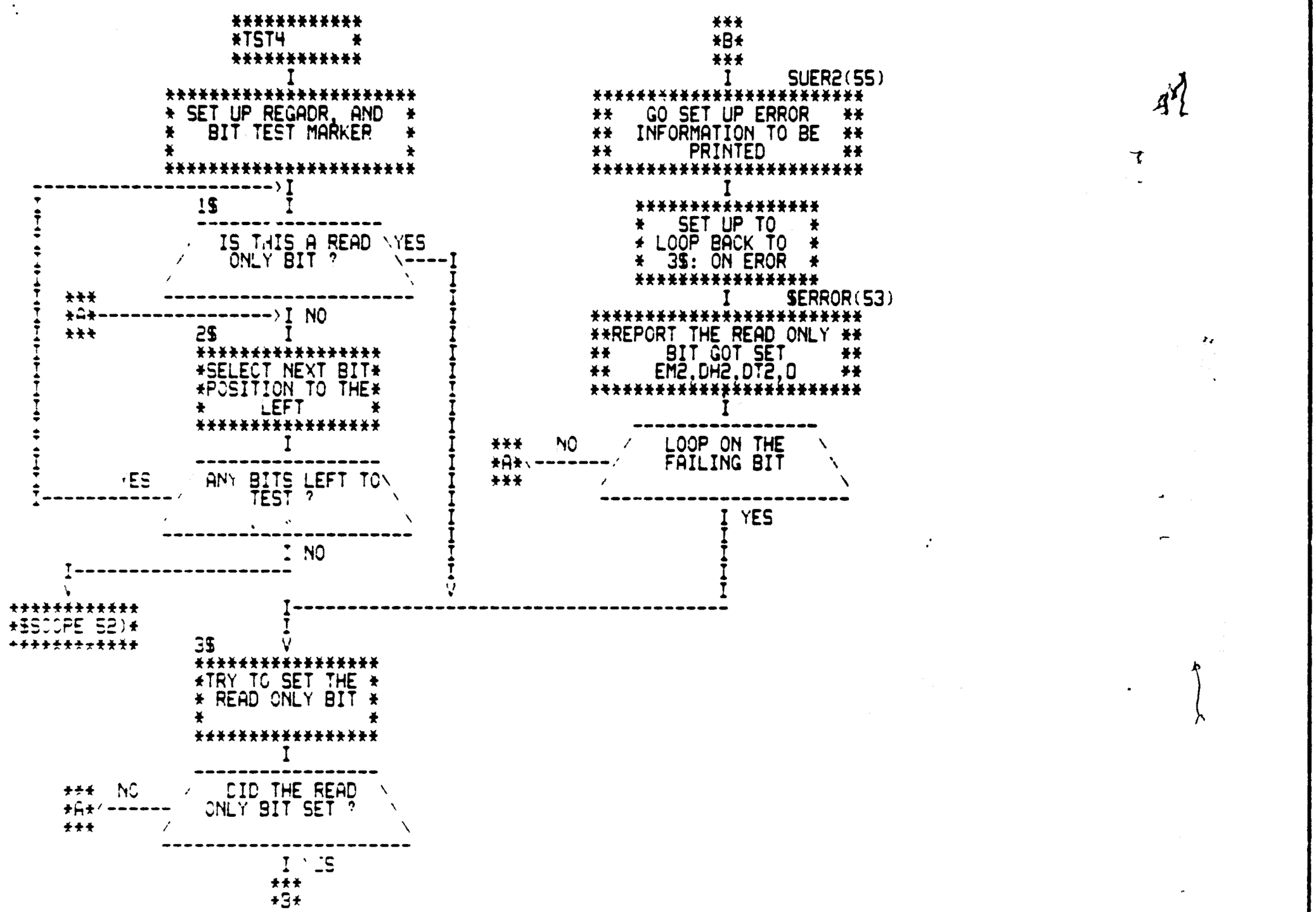
C05


RSTRTA(01)
SET UP CURRENT ADDR- ESS,VECTOR,AND PRIO- RITY LEVEL FOR TEST
SUNUM(56)
GO SET UP DEVICE NUMBER IN MESSAGE
PRINT: "TESTING DH11 #XX"
LDTBF1(55)
GO SET UP XMITTER DATA BUFFER
INIT SCOPE LOOP RETURN ADDRESS
TST1.03
*INPAR(01) *
TYPE MESSAGE TO ASK FOR DEVICE ADDRESS
USER DONE TYPING ADDRESS
NO
YES
CHKADR(58)
GO CHECK VALIDITY OF ADDRESS
VALID DEVICE ADDRESS FOR DH11
NO
YES
TYPE MESSAGE TO ASK FOR VECTOR ADRESS
USER DONE TYPING ADDRESS
NO
YES
CHKVCT(58)
GO CHECK VALIDITY OF ADDRESS
VALID VECTOR ADDRESS
NO
YES
INPAR3(02)
TYPE INVALID DEVICE ADDRESS ERROR MESSAGE
TYPE INVALID VECTOR ADDRESS
INPAR3(01)
TYPE MESSAGE TO ASK FOR DEVICE SELECT PARAMETER
USER DONE TYPING PARAMETER
NO
YES
SET UP DEVICE *SELECTION PARA-METER "DHSEL"
TYPE MESSAGE TO ASK FOR LINE SEL- ECTION PARAMETER
USER DONE TYPING PARAMETER
NO
YES
SET UP LINE *SELECTION PARA-METER "LINSEL"
HALT AFTER SETUP
NO
YES
HALT
CONTINUE DEPRESSED
START3.01

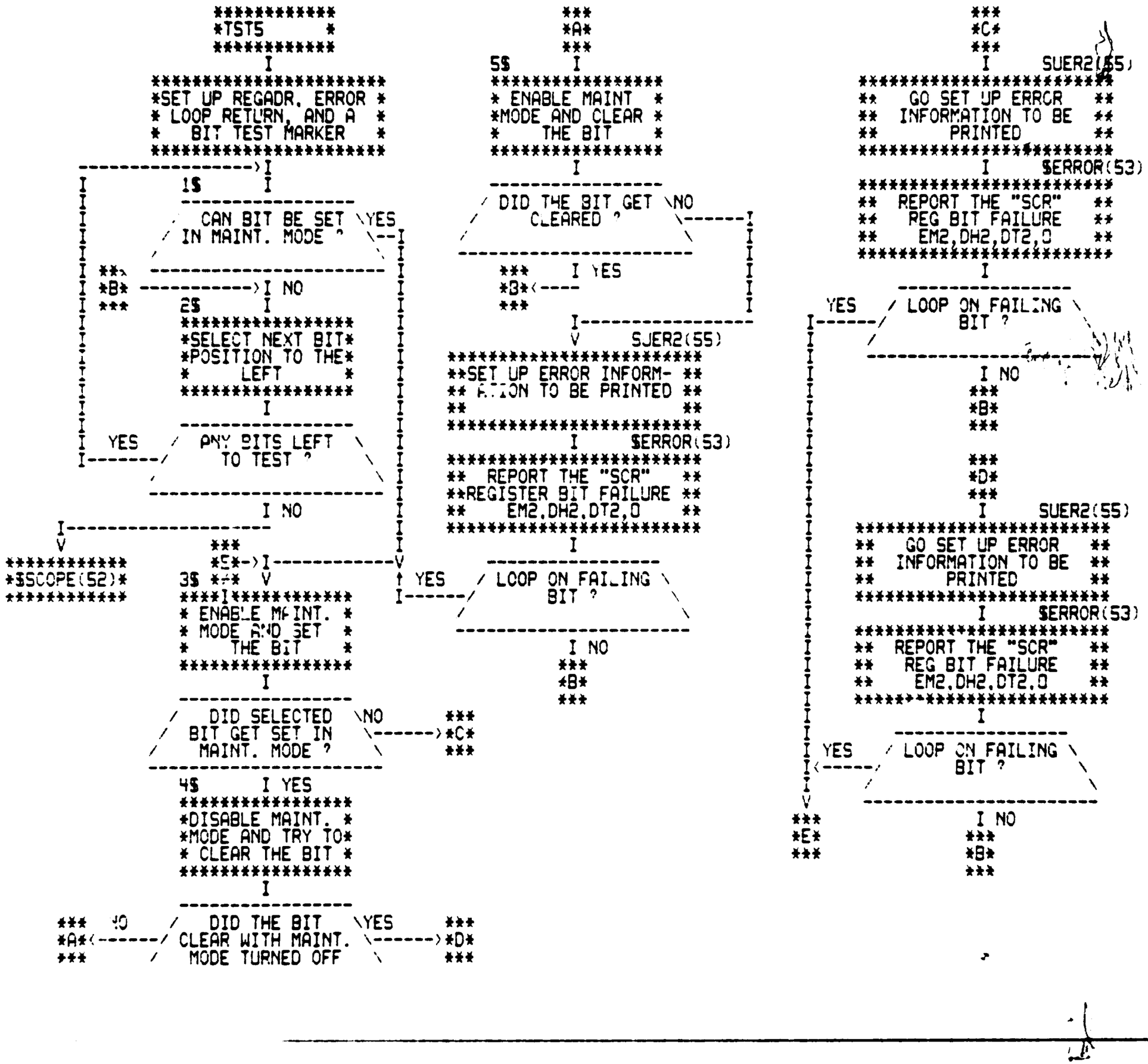

TST1(02)
LOAD R2 AND R5 WITH DHADR AND ADD 16(9) TO R5
SAVE TIMEOUT VECTOR ON THE STACK
SET UP TIMEOUT VECTOR TO GO TO 3$: ON TRAP
1$
SELECT NEXT DH11 REGISTER ADDRESS
2$
ATTEMPT TO ACCESS SELECTED ADDRESS
DID SELECTED DH11 ADDRESS CAUSE TIMEOUT TRAP ?
YES
A
NO
TESTED ALL ADDRESSES FOR THIS DH11
NO
YES
RESTORE BUS TIMEOUT VECTOR FROM STACK
$SCOPE(52)
A
3$
SUER1(55)
GO SET UP ERROR INFORMATION TO BE PRINTED
RESET SP TO CORRECT FOR TRAP
$ERROR(53)
REPORT TIMEOUT TRAP AT SELECTED DH11 ADDR
EM1,DH1,DT1,0
LOOP ON FAILING ADDRESS ? SR09=1
YES
NO

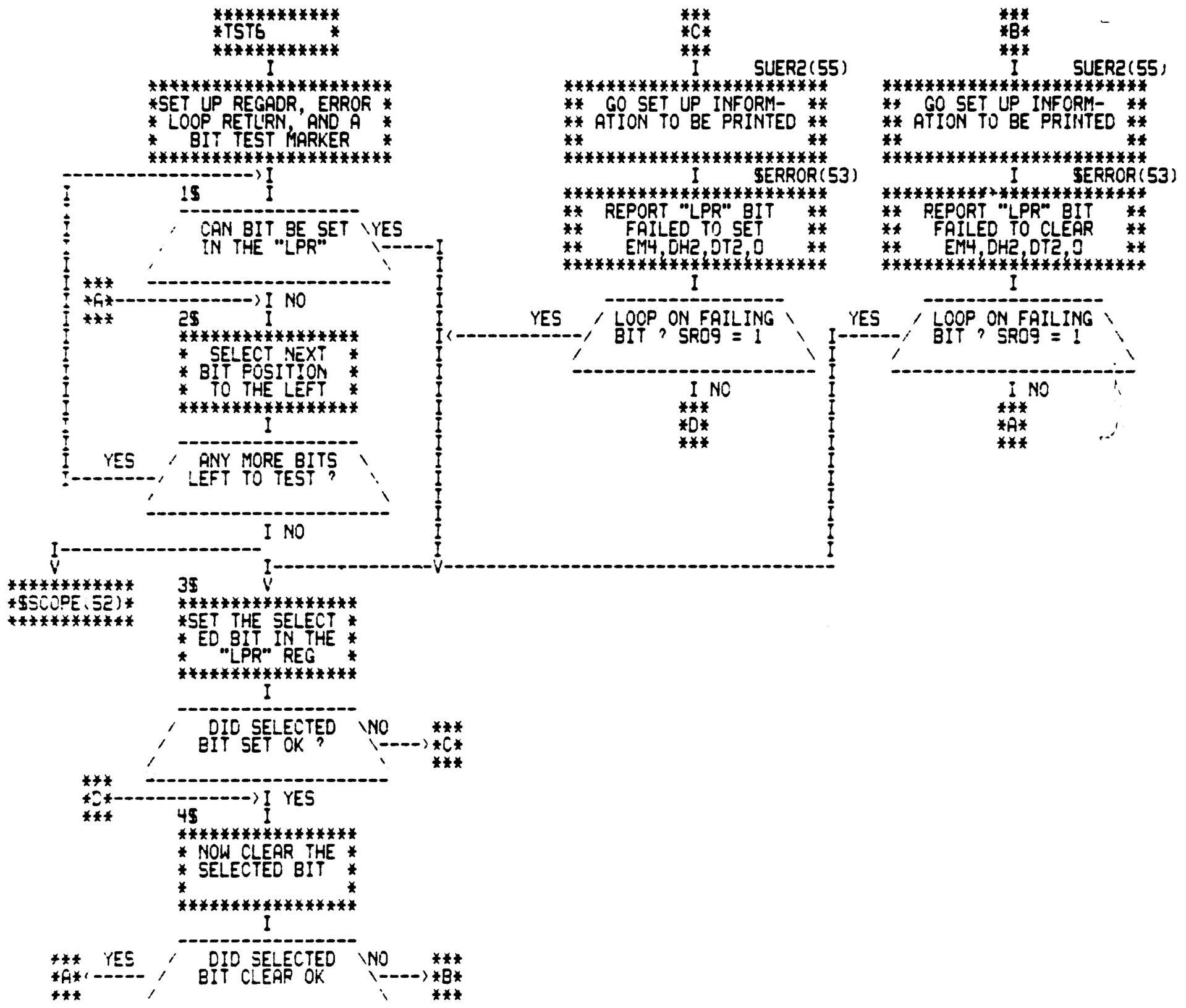

*TST2
SET UP TABLE OF TEST ADDRESSES FOR "SCR" "LPR","BKR",AND"SSR"
MAKE S/B = 0
1$
LOAD R2 WITH TEST ADDRESS FROM TABLE
TESTED ALL FOUR TEST ADDRESSES ?
YES
NO
2$
SET ALL WRITE BITS AT TEST ADDRESS
SET "MSTCLR" BIT IN SCR REG
DID "MSTCLR" CLEAR ALL BITS AT TEST ADDRESS
NO
YES
A
SUER2(55)
GO SET UP ERROR INFORMATION TO BE PRINTED
$ERROR(53)
REPORT "MSTCLR" FAILED TO CLEAR REG EM3.DH2,DT2.0
LOOP ON FAILING ADDRESS SR09=1
YES
NO

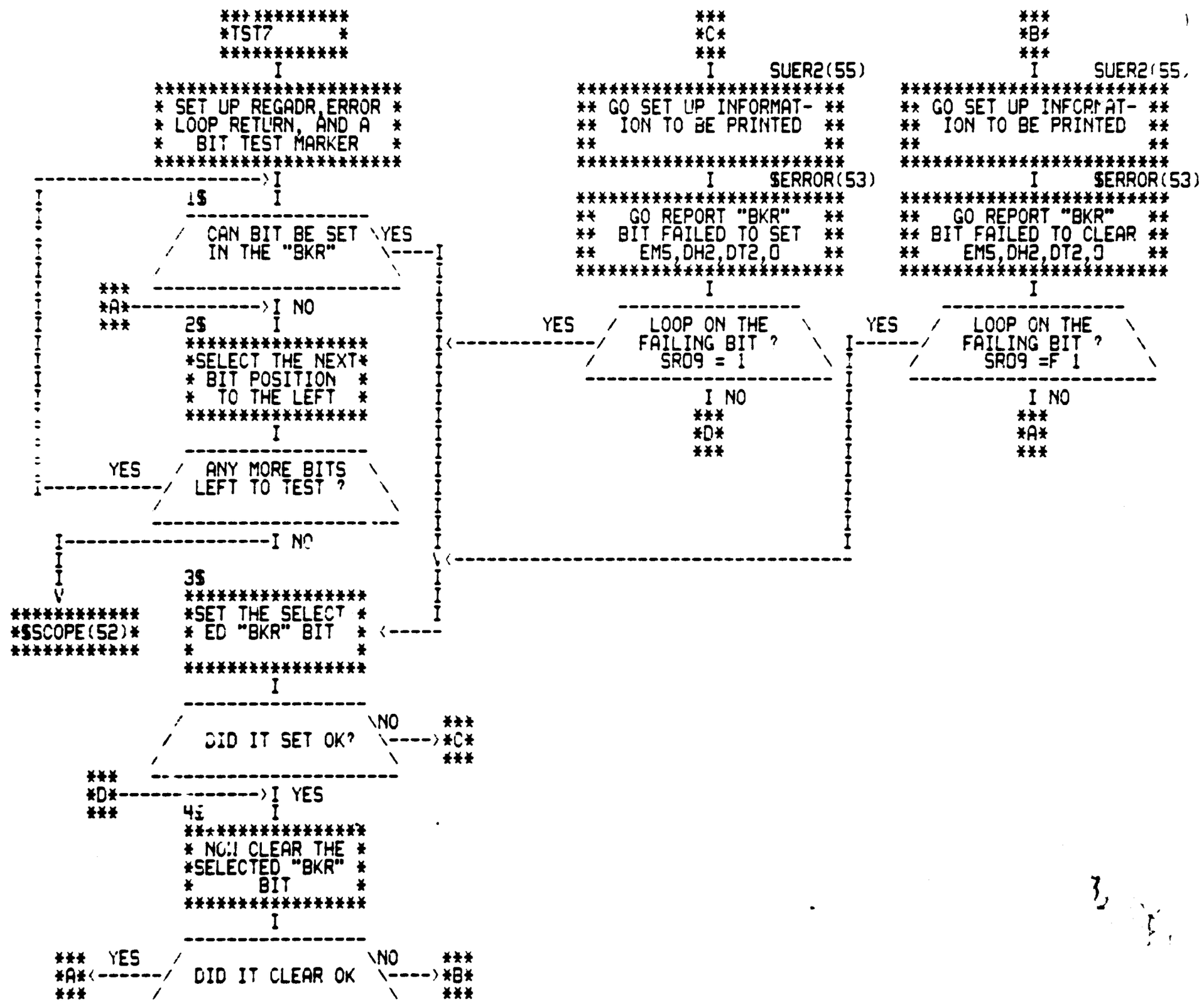

$ST3
SET UP REGADR,ERROR
LOOP RETURN,AND BIT
MARKER TO SELECT BIT
1$
IS THIS BIT SELECTED FOR TEST (RGMSK1)
C
NO
2$
SELECT NEXT BIT POSITION TO THE LEFT
3$
ANY BITS LEFT TO TEST ?
YES
NO
$SCOPE(52)
4$
SET SELECTED BIT IN "SCR"
DID SELECTED BIT SET PROPERLY
NO
A
YES
NOW CLEAR THE SELECTED BIT
DID SELECTED BIT CLEAR PROPERLY ?
NO
B
YES
C
A
SUER2(55)
SET UP ERROR INFORM- ATION TO BE PRINTED
$ERROR(53)
REPORT "SCR" REG BIT FAILURE
LOOP ON FAILING BIT ? SR09=1 EM2,DH2,DT2,0
NO
C
YES
B
SUER2(55)
SET UP ERROR INFORM- ATION TO BE PRINTED
$ERROR(53)
REPORT "SCR" REG BIT FAILURE
LOOP ON FAILING BIT ? SR09=1 EM2,DH2,DT2,0
NO
C
YES

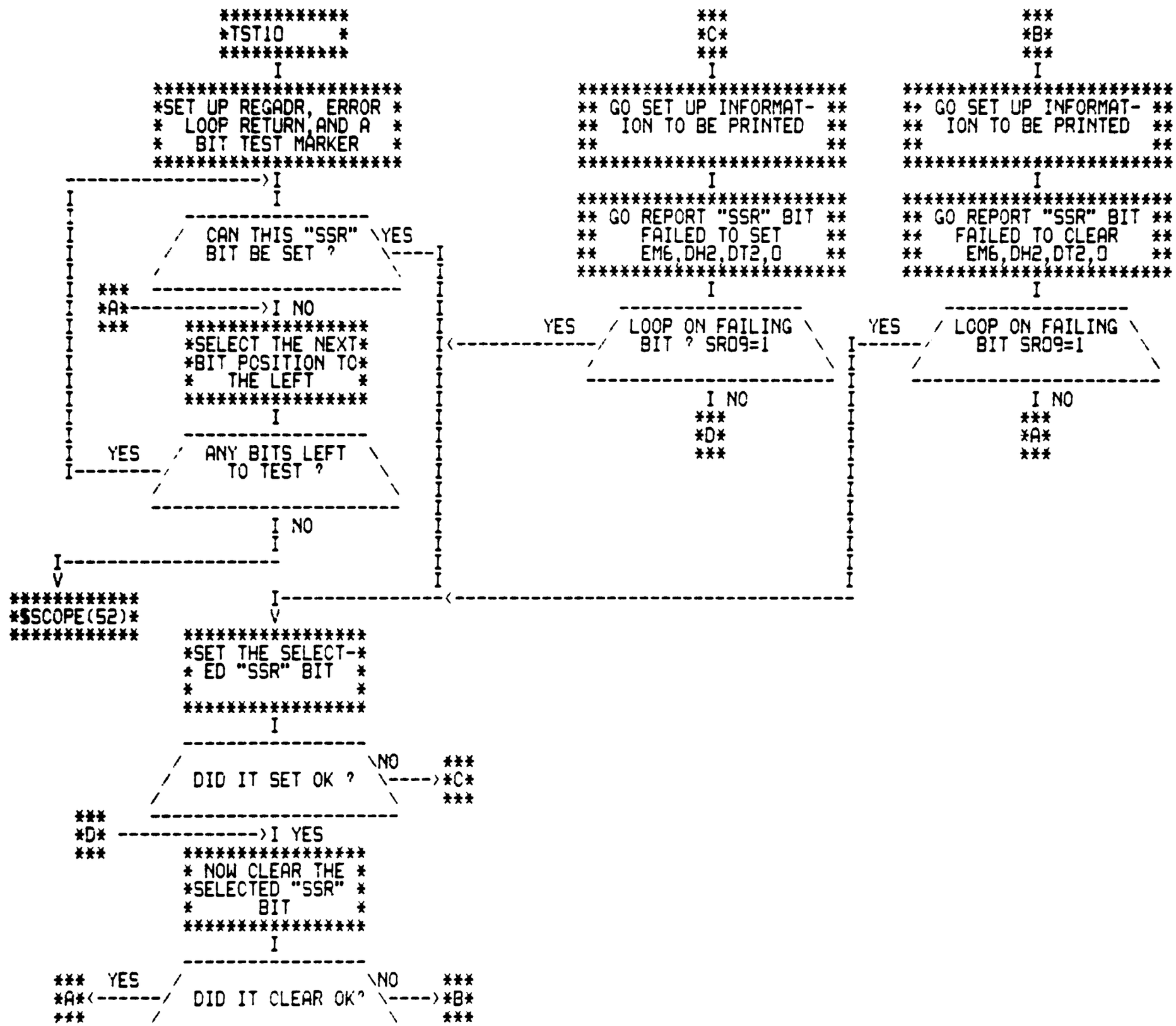

TST4
SET UP REGADR, AND BIT TEST MARKER
1$
IS THIS A READ ONLY BIT ?
YES
NO
2$
SELECT NEXT BIT POSITION TO THE LEFT
ANY BITS LEFT TO TEST ?
YES
NO
$SCOPE 52)
3$
TRY TO SET THE READ ONLY BIT
DID THE READ ONLY BIT SET ?
YES
A
B
SUER2(55)
GO SET UP ERROR INFORMATION TO BE PRINTED
SET UP TO LOOP BACK TO 3$: ON EROR
$ERROR(53)
REPORT THE READ ONLY BIT GOT SET EM2,DH2,DT2,0
LOOP ON THE FAILING BIT
NO
YES

H05

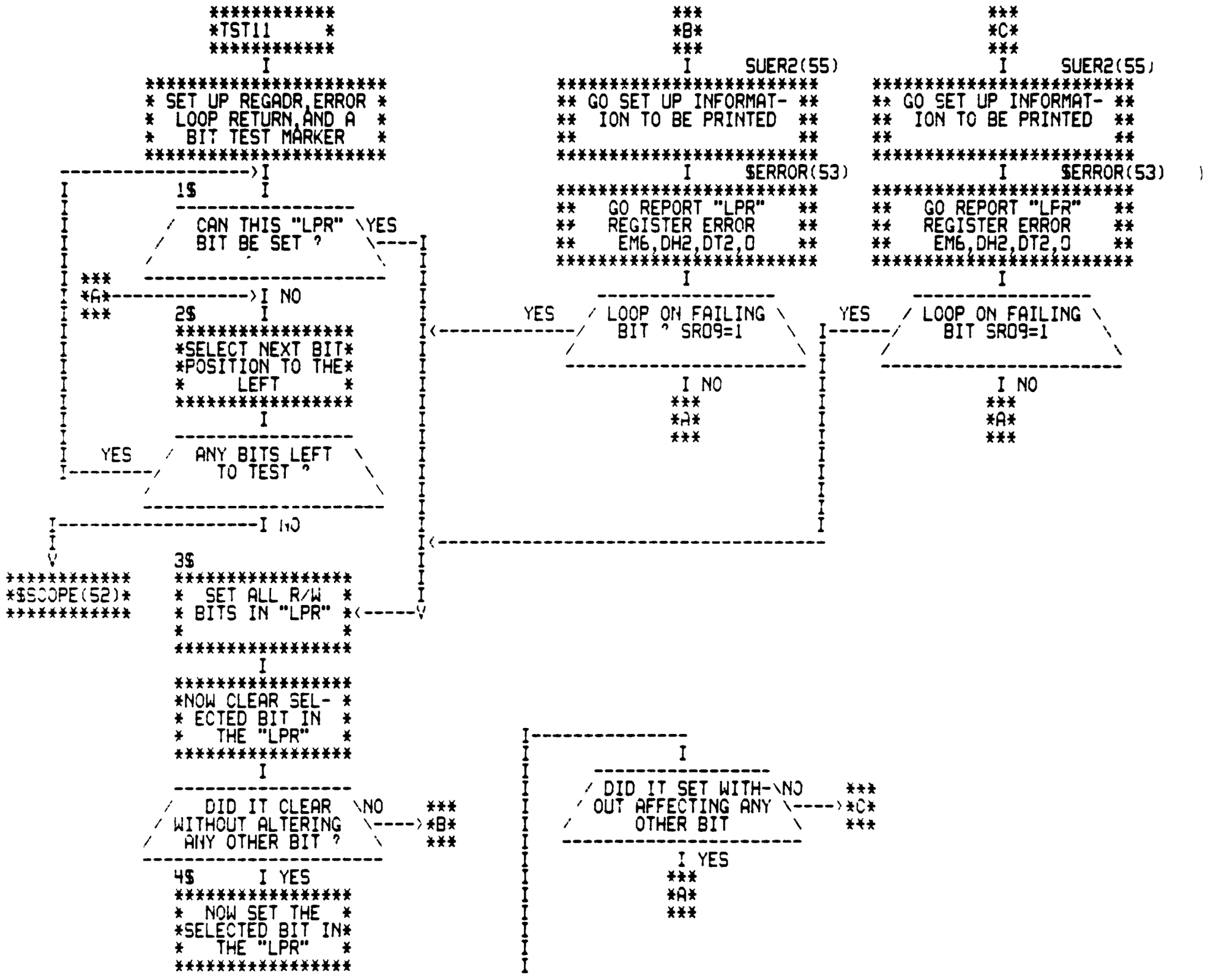

*TST5 *
SET UP REGADR, ERROR LOOP RETURN, AND A BIT TEST MARKER
1$
CAN BIT BE SET IN MAINT. MODE ?
YES
NO
2$
SELECT NEXT BIT POSITION TO THE LEFT
ANY BITS LEFT TO TEST ?
YES
NO
3$
ENABLE MAINT. MODE AND SET THE BIT
DID SELECTED BIT GET SET IN MAINT. MODE ?
NO
YES
4$
DISABLE MAINT. MODE AND TRY TO CLEAR THE BIT
DID THE BIT CLEAR WITH MAINT. MODE TURNED OFF
NO
YES
A
B
C
D
E
$SCOPE(52)
5$
ENABLE MAINT MODE AND CLEAR THE BIT
DID THE BIT GET CLEARED ?
NO
YES
SUER2(55)
SET UP ERROR INFORM- ATION TO BE PRINTED
$ERROR(53)
REPORT THE "SCR" REGISTER BIT FAILURE EM2,DH2,DT2,0
LOOP ON FAILING BIT ?
YES
NO
SUER2(55)
GO SET UP ERROR INFORMATION TO BE PRINTED
$ERROR(53)
REPORT THE "SCR" REG BIT FAILURE EM2,DH2,DT2,0
LOOP ON FAILING BIT ?
YES
NO
SUER2(55)
GO SET UP ERROR INFORMATION TO BE PRINTED
$ERROR(53)
REPORT THE "SCR" REG BIT FAILURE EM2,DH2,DT2,0
LOOP ON FAILING BIT ?
YES
NO

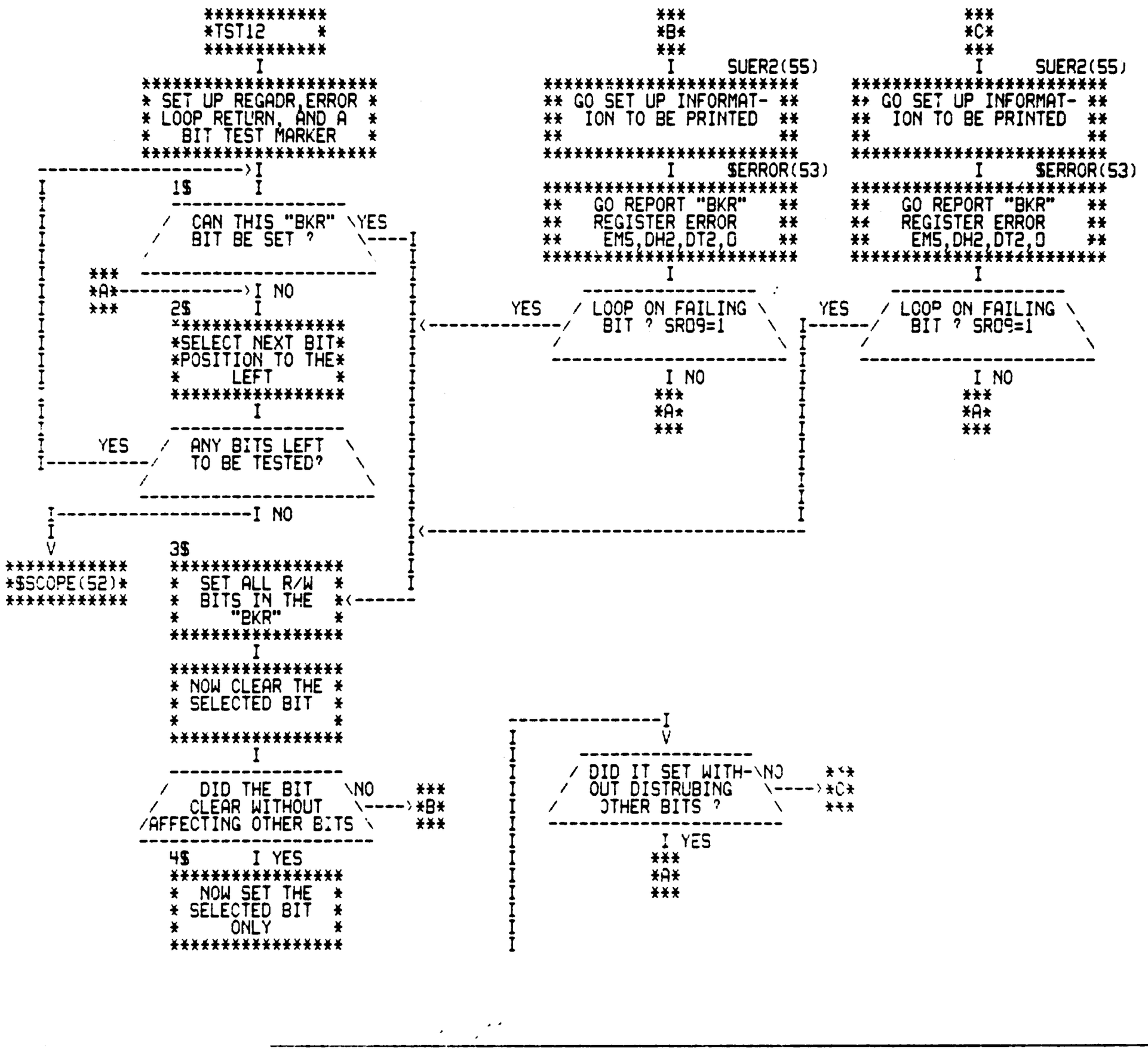

*TST6 *
SET UP REGADR, ERROR LOOP RETURN, AND A BIT TEST MARKER
1$
CAN BIT BE SET IN THE "LPR"
YES
NO
A
2$
SELECT NEXT BIT POSITION TO THE LEFT
ANY MORE BITS LEFT TO TEST ?
YES
NO
$SCOPE,52)
3$
SET THE SELECTED BIT IN THE "LPR" REG
DID SELECTED BIT SET OK ?
NO
C
YES
4$
NOW CLEAR THE SELECTED BIT
DID SELECTED BIT CLEAR OK
YES
A
NO
B
C
SUER2(55)
GO SET UP INFORM- ATION TO BE PRINTED
$ERROR(53)
REPORT "LPR" BIT FAILED TO SET
EM4,DH2,DT2,0
LOOP ON FAILING BIT ? SR09 = 1
NO
D
B
SUER2(55)
GO SET UP INFORM- ATION TO BE PRINTED
$ERROR(53)
REPORT "LPR" BIT FAILED TO CLEAR
EM4,DH2,DT2,0
LOOP ON FAILING BIT ? SR09 = 1
NO
A

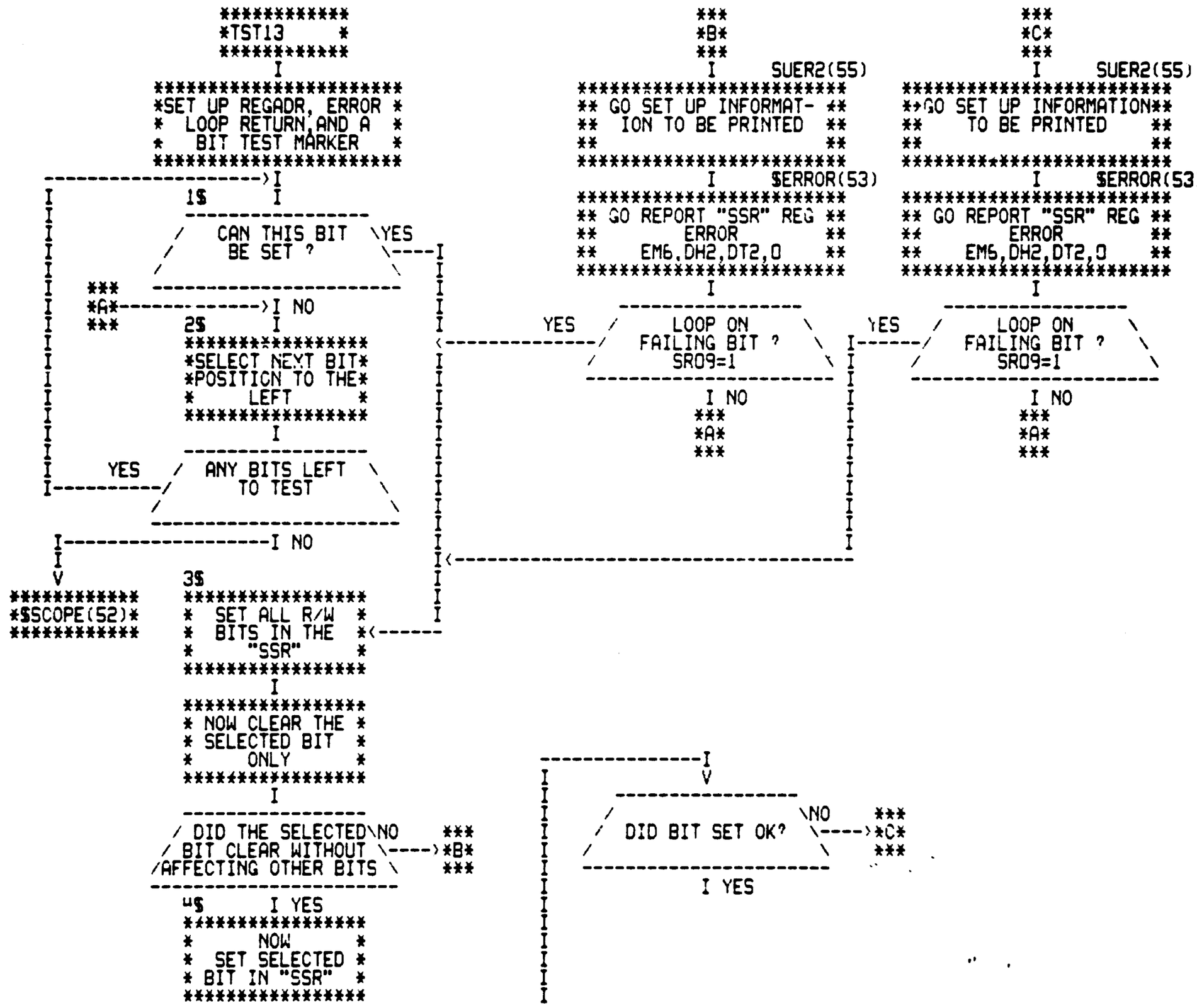

*TST7
SET UP REGADR,ERROR
LOOP RETURN, AND A
BIT TEST MARKER
1$
CAN BIT BE SET IN THE "BKR"
YES
NO
A
2$
SELECT THE NEXT
BIT POSITION
TO THE LEFT
ANY MORE BITS LEFT TO TEST ?
YES
NO
3$
SET THE SELECTED "BKR" BIT
$SCOPE(52)
DID IT SET OK?
NO
C
D
4$
NOW CLEAR THE SELECTED "BKR" BIT
DID IT CLEAR OK
NO
B
A
C
SUER2(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT "BKR" BIT FAILED TO SET
EMS,DH2,DT2,0
LOOP ON THE FAILING BIT ?
SR09 = 1
NO
D
B
SUER2(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT "BKR" BIT FAILED TO CLEAR
EMS,DH2,DT2,0
LOOP ON THE FAILING BIT ?
SR09 =F 1
NO
A

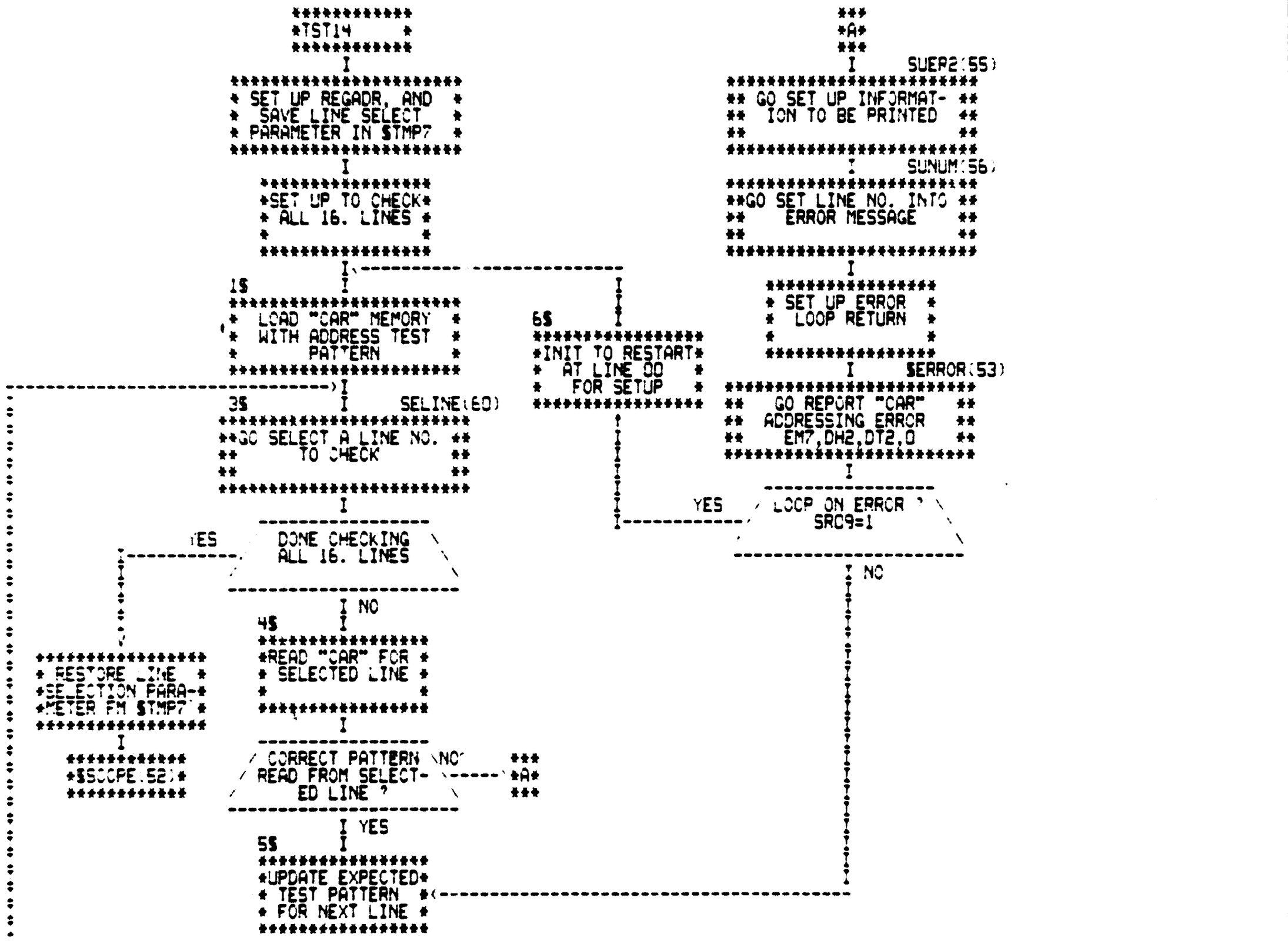

*TST10
*SET UP REGADR, ERROR LOOP RETURN, AND A BIT TEST MARKER
CAN THIS "SSR" BIT BE SET ?
YES
NO
A
*SELECT THE NEXT BIT POSITION TO THE LEFT
ANY BITS LEFT TO TEST ?
YES
NO
$SCOPE(52)
*SET THE SELECTED "SSR" BIT
DID IT SET OK ?
NO
C
D
YES
* NOW CLEAR THE SELECTED "SSR" BIT
DID IT CLEAR OK?
NO
B
YES
A
C
GO SET UP INFORMATION TO BE PRINTED
GO REPORT "SSR" BIT FAILED TO SET EM6,DH2,DT2,0
LOOP ON FAILING BIT ? SR09=1
NO
D
B
GO SET UP INFORMATION TO BE PRINTED
GO REPORT "SSR" BIT FAILED TO CLEAR EM6,DH2,DT2,0
LOOP ON FAILING BIT SR09=1
NO
A

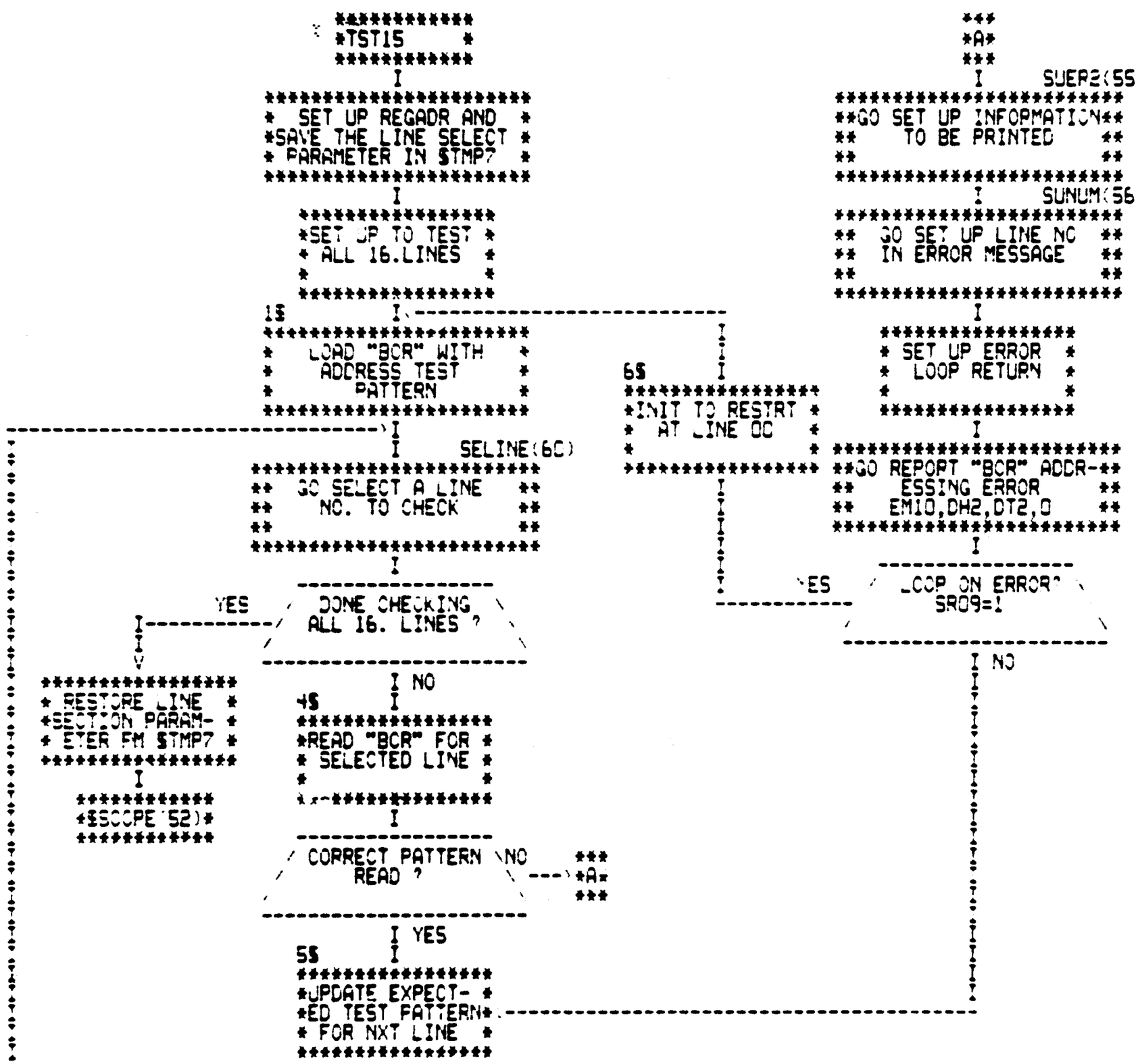
LO5

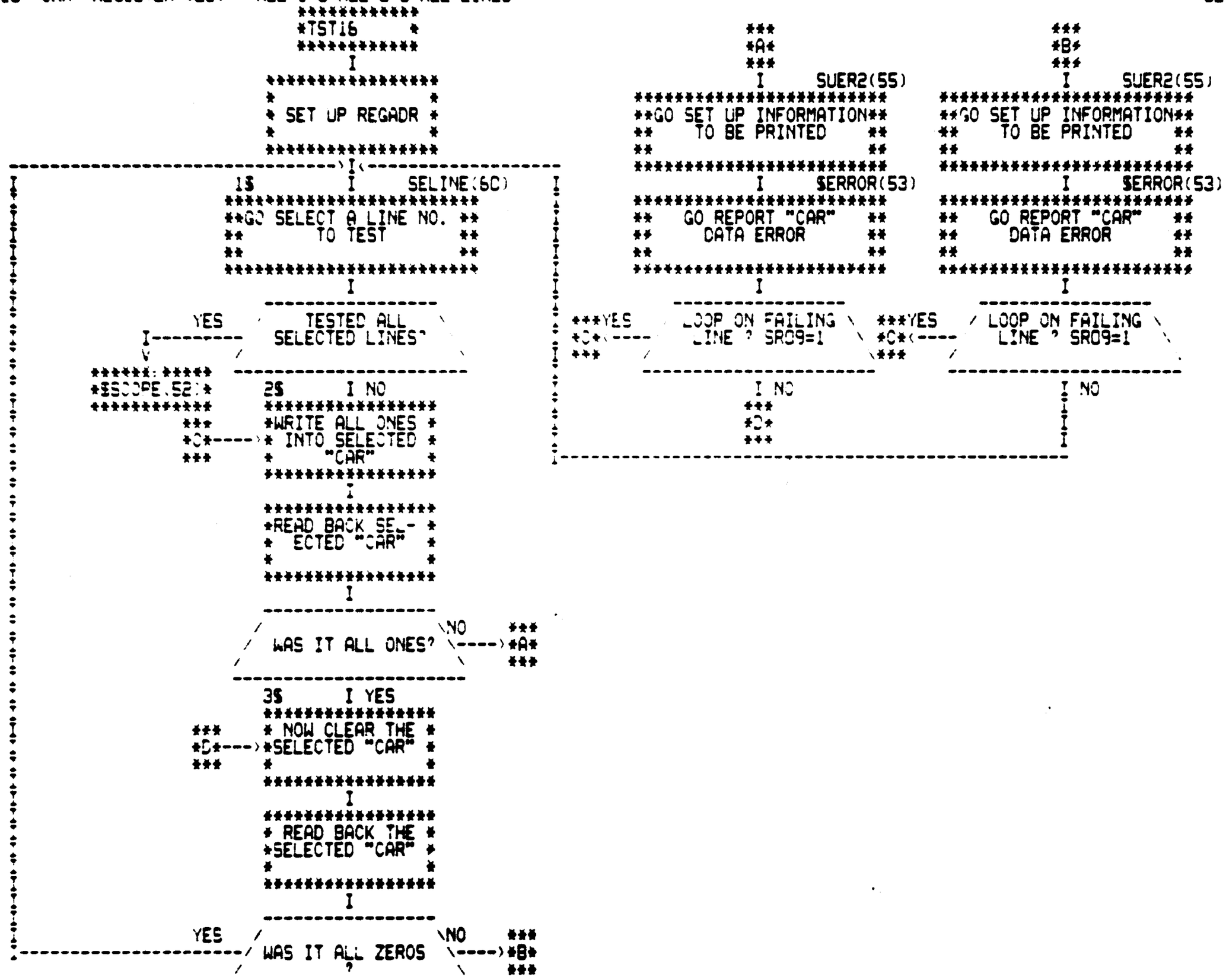

*TST11
SET UP REGADR,ERROR LOOP RETURN,AND A BIT TEST MARKER
1$
CAN THIS "LPR" BIT BE SET ?
YES
NO
A
2$
SELECT NEXT BIT *POSITION TO THE* LEFT
ANY BITS LEFT TO TEST ?
YES
NO
3$
$SCOPE(52)
SET ALL R/W BITS IN "LPR"
*NOW CLEAR SEL- ECTED BIT IN THE "LPR"
DID IT CLEAR WITHOUT ALTERING ANY OTHER BIT ?
NO
B
4$
YES
NOW SET THE *SELECTED BIT IN* THE "LPR"
B
SUER2(55)
GO SET UP INFORMAT- ION TO BE PRINTED
$ERROR(53)
GO REPORT "LPR" REGISTER ERROR EM6,DH2,DT2,0
LOOP ON FAILING BIT ? SR09=1
YES
NO
A
C
SUER2(55)
GO SET UP INFORMAT- ION TO BE PRINTED
$ERROR(53)
GO REPORT "LPR" REGISTER ERROR EM6,DH2,DT2,0
LOOP ON FAILING BIT SR09=1
YES
NO
A
DID IT SET WITH- OUT AFFECTING ANY OTHER BIT
NO
C
YES
A


*TST12
SET UP REGADR,ERROR
LOOP RETURN, AND A
BIT TEST MARKER
1$
CAN THIS "BKR"
BIT BE SET ?
YES
NO
A
2$
SELECT NEXT BIT
POSITION TO THE
LEFT
ANY BITS LEFT
TO BE TESTED?
YES
NO
$SCOPE(52)
3$
SET ALL R/W
BITS IN THE
"BKR"
NOW CLEAR THE
SELECTED BIT
DID THE BIT
CLEAR WITHOUT
AFFECTING OTHER BITS
NO
B
4$
YES
NOW SET THE
SELECTED BIT
ONLY
B
SUER2(55)
GO SET UP INFORMAT-
ION TO BE PRINTED
$ERROR(53)
GO REPORT "BKR"
REGISTER ERROR
EM5,DH2,DT2,0
LOOP ON FAILING
BIT ? SR09=1
YES
NO
A
C
SUER2(55)
GO SET UP INFORMAT-
ION TO BE PRINTED
$ERROR(53)
GO REPORT "BKR"
REGISTER ERROR
EM5,DH2,DT2,0
LOOP ON FAILING
BIT ? SR09=1
YES
NO
A
DID IT SET WITH-
OUT DISTRUBING
OTHER BITS ?
NO
C
YES
A

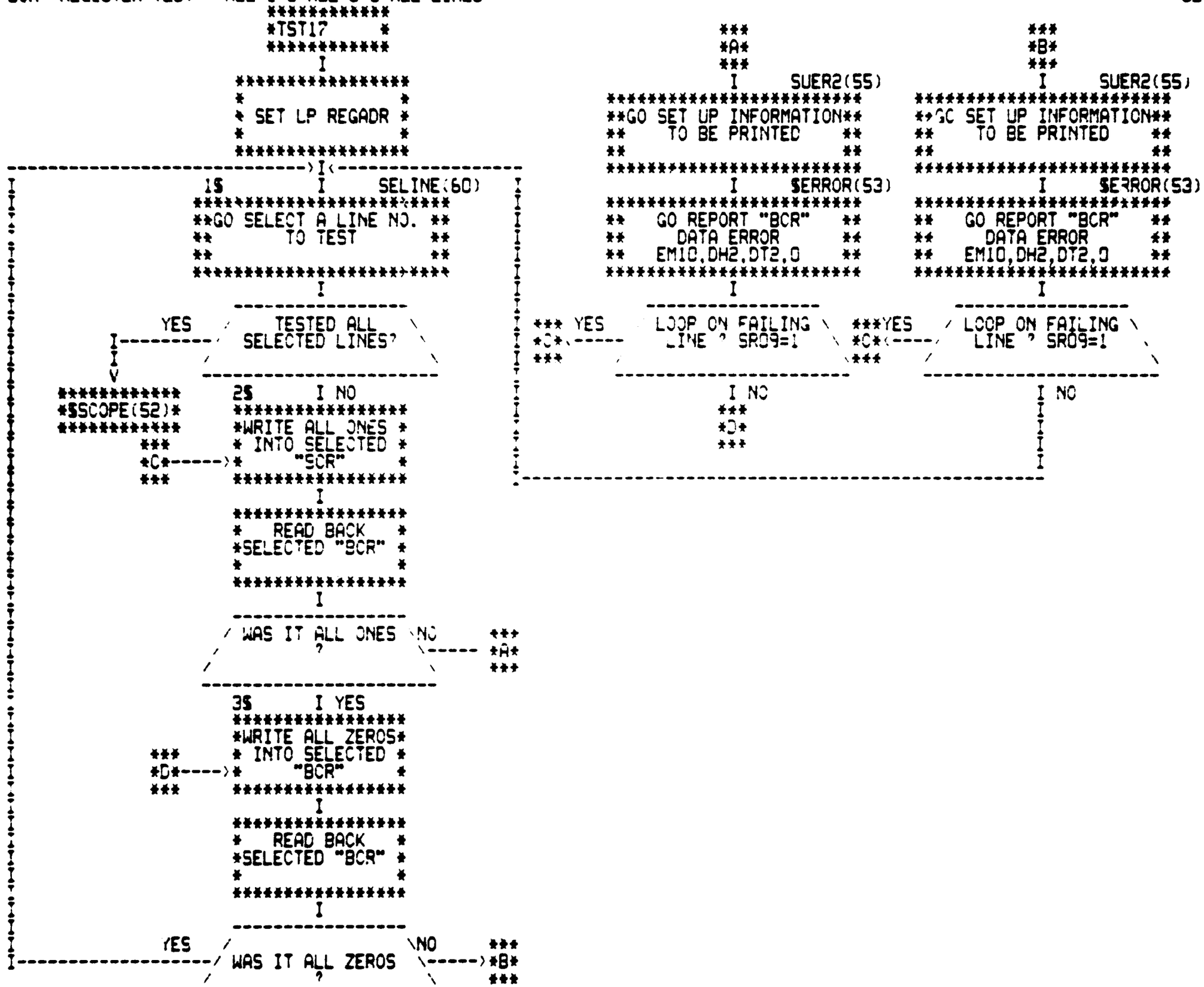

*TST13
*SET UP REGADR, ERROR LOOP RETURN, AND A BIT TEST MARKER
1$
CAN THIS BIT BE SET ?
YES
NO
A
2$
SELECT NEXT BIT *POSITION TO THE* LEFT
ANY BITS LEFT TO TEST
YES
NO
$SCOPE(52)
3$
SET ALL R/W BITS IN THE "SSR"
NOW CLEAR THE SELECTED BIT ONLY
DID THE SELECTED BIT CLEAR WITHOUT AFFECTING OTHER BITS
NO
B
4$
NOW SET SELECTED BIT IN "SSR"
YES
B
SUER2(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT "SSR" REG ERROR EM6,DH2,DT2,0
LOOP ON FAILING BIT ? SR09=1
YES
NO
A
C
SUER2(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT "SSR" REG ERROR EM6,DH2,DT2,0
LOOP ON FAILING BIT ? SR09=1
YES
NO
A
DID BIT SET OK?
NO
C
YES


TST14
SET UP REGADR, AND SAVE LINE SELECT PARAMETER IN $TMP7
SET UP TO CHECK ALL 16. LINES
1$
LOAD "CAR" MEMORY WITH ADDRESS TEST PATTERN
3$
SELINE(60)
GO SELECT A LINE NO. TO CHECK
DONE CHECKING ALL 16. LINES
YES
NO
4$
READ "CAR" FOR SELECTED LINE
CORRECT PATTERN READ FROM SELECTED LINE ?
NO
YES
5$
UPDATE EXPECTED TEST PATTERN FOR NEXT LINE
RESTORE LINE SELECTION PARAMETER FM $TMP7
$SCOPE.52
6$
INIT TO RESTART AT LINE 00 FOR SETUP
A
SUER2(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET LINE NO. INTO ERROR MESSAGE
SET UP ERROR LOOP RETURN
SERROR(53)
GO REPORT "CAR" ADDRESSING ERROR EM7,DH2,DT2,0
LOOP ON ERROR SR09=1
YES
NO

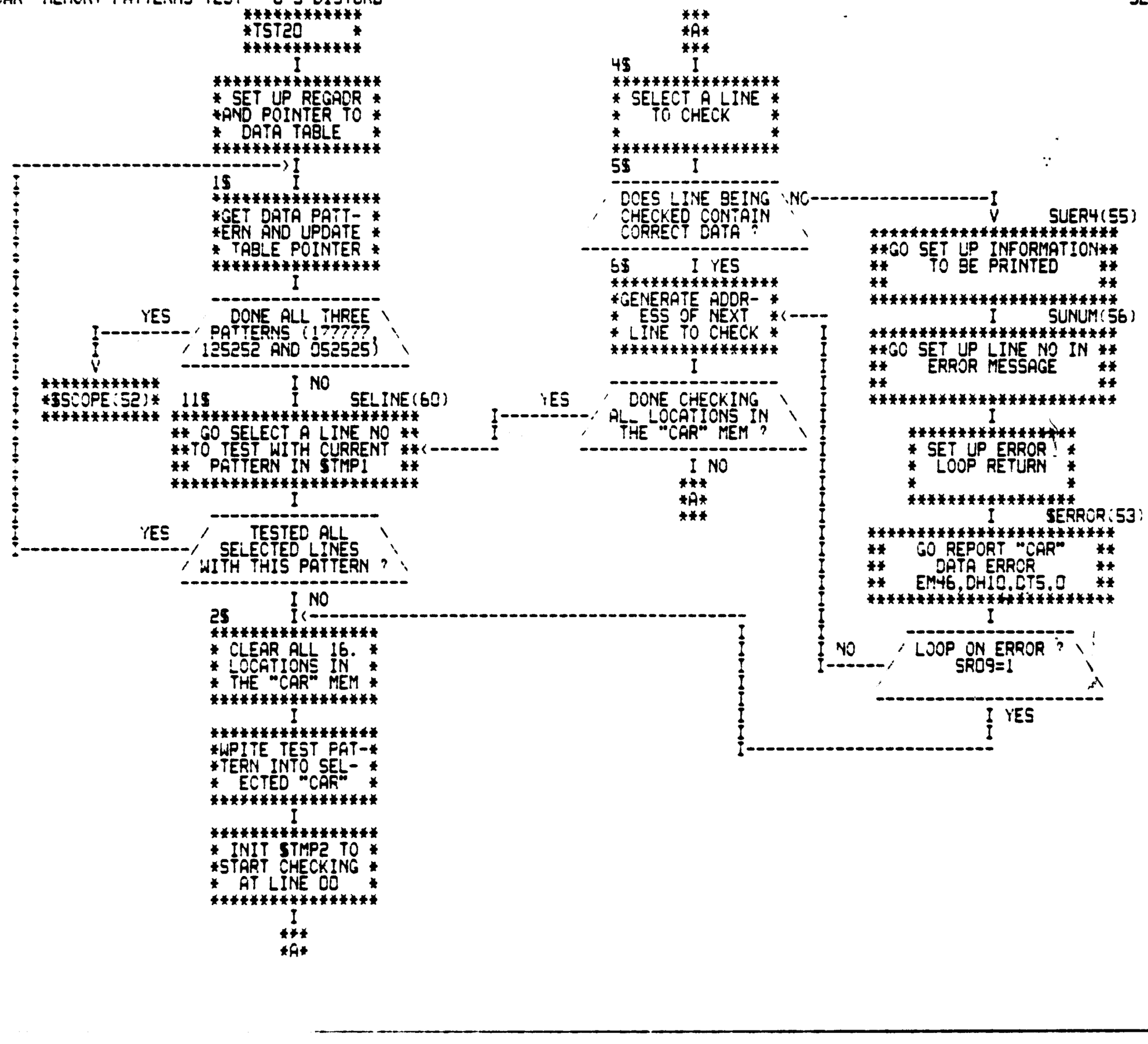

TST15
SET UP REGADR AND SAVE THE LINE SELECT PARAMETER IN $TMP7
SET UP TO TEST ALL 16.LINES
1$
LOAD "BCR" WITH ADDRESS TEST PATTERN
SELINE(6C)
GO SELECT A LINE NO. TO CHECK
DONE CHECKING ALL 16. LINES ?
YES
NO
RESTORE LINE SECTION PARAM- ETER FM $TMP7
$SCOPE(52)
4$
READ "BCR" FOR SELECTED LINE
CORRECT PATTERN READ ?
NO
A
YES
5$
UPDATE EXPECT- ED TEST PATTERN FOR NXT LINE
6$
INIT TO RESTRT AT LINE 00
A
SUER2(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO IN ERROR MESSAGE
SET UP ERROR LOOP RETURN
GO REPORT "BCR" ADDR-ESSING ERROR EM10,DH2,DT2,0
LOOP ON ERROR? SR09=1
YES
NO

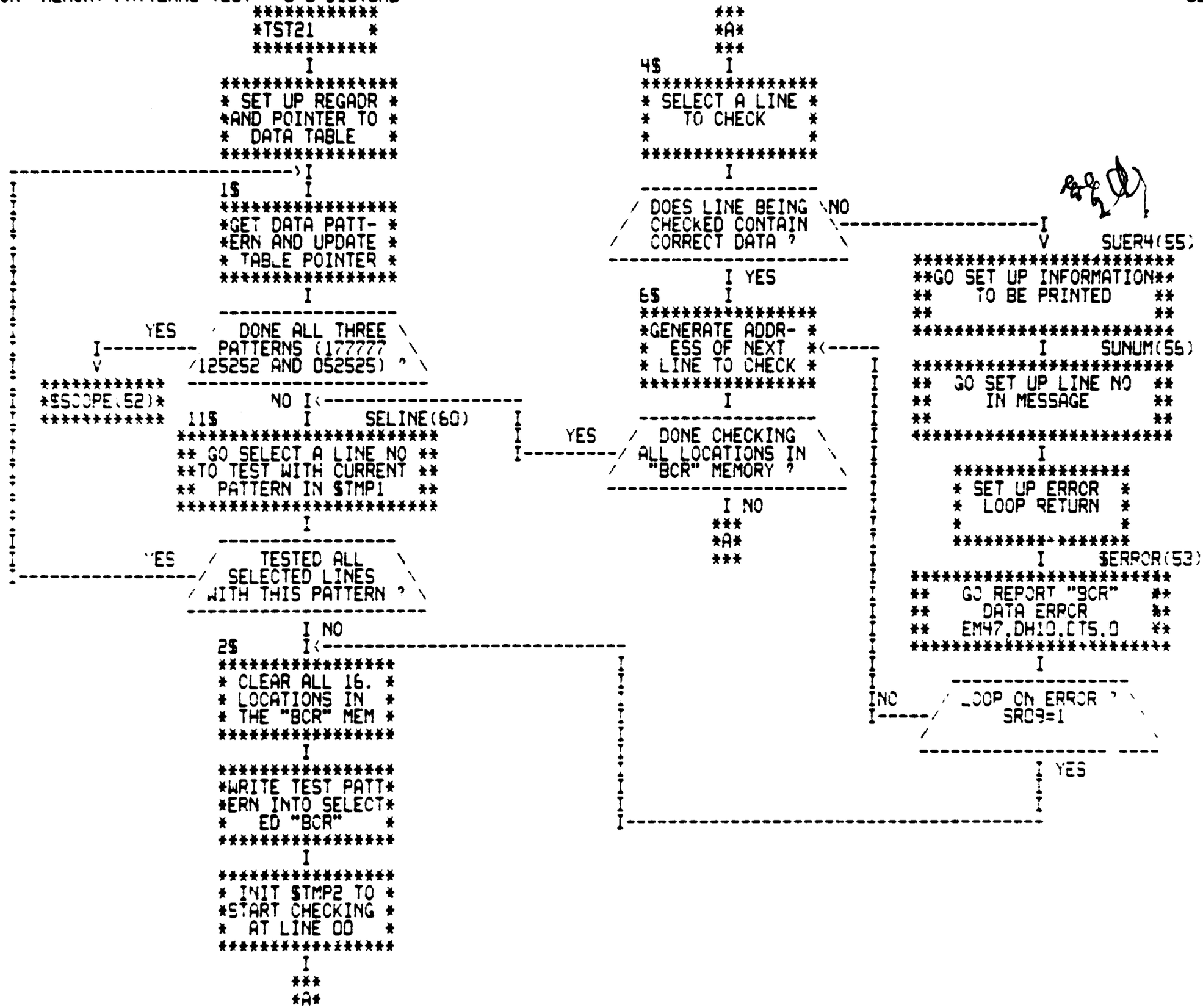

*TST16
SET UP REGADR
SELINE(6C)
GO SELECT A LINE NO. TO TEST
TESTED ALL SELECTED LINES?
YES
2$
NO
WRITE ALL ONES INTO SELECTED "CAR"
READ BACK SELECTED "CAR"
WAS IT ALL ONES?
NO
3$
YES
NOW CLEAR THE SELECTED "CAR"
READ BACK THE SELECTED "CAR"
WAS IT ALL ZEROS ?
NO
SUER2(55)
GO SET UP INFORMATION TO BE PRINTED
SERROR(53)
GO REPORT "CAR" DATA ERROR
YES
LOOP ON FAILING LINE ? SR09=1
NO
SUER2(55)
GO SET UP INFORMATION TO BE PRINTED
SERROR(53)
GO REPORT "CAR" DATA ERROR
YES
LOOP ON FAILING LINE ? SR09=1
NO

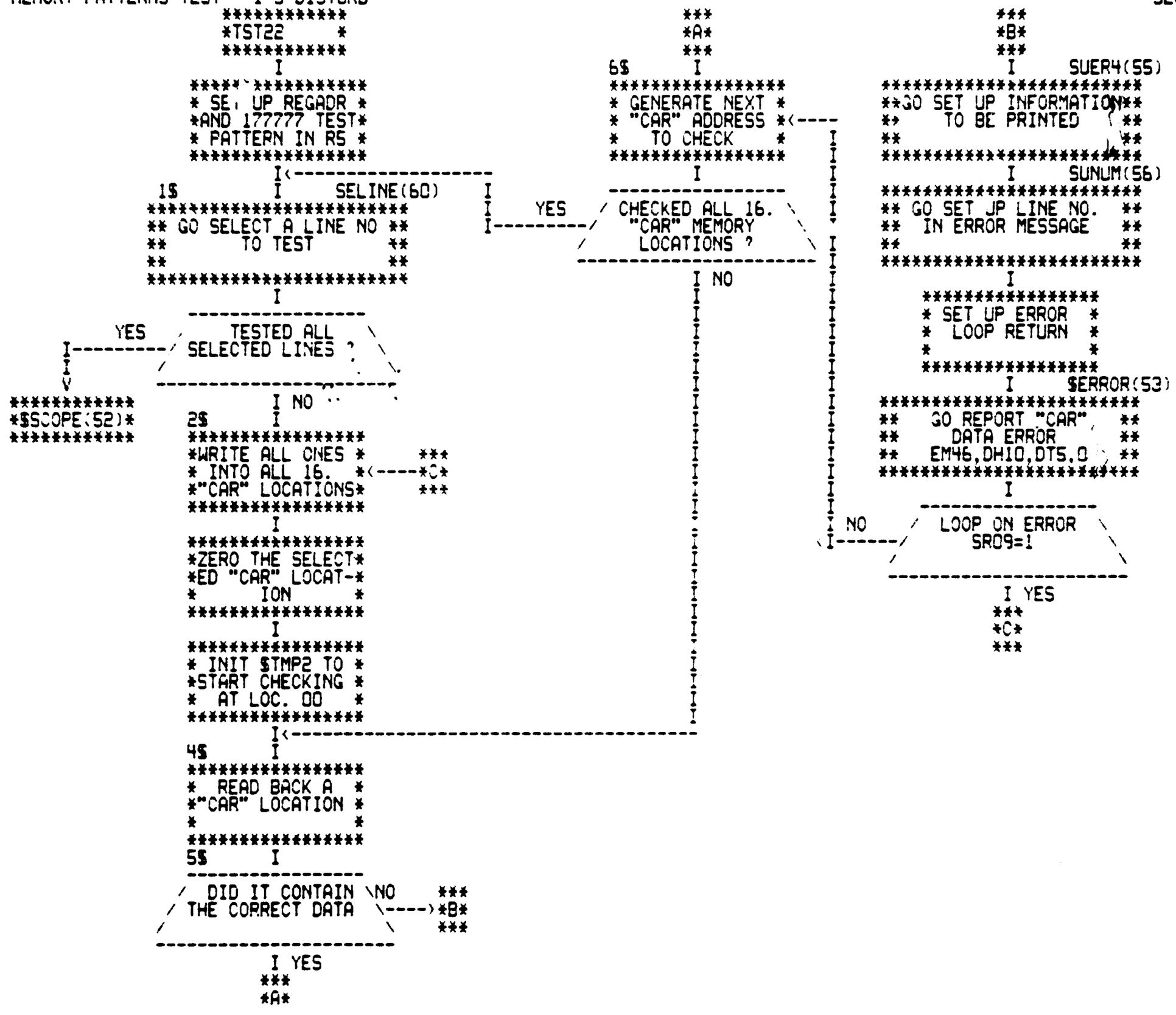
TEST 17 "BCR" REGISTER TEST - ALL 1'S ALL 0'S ALL LINES

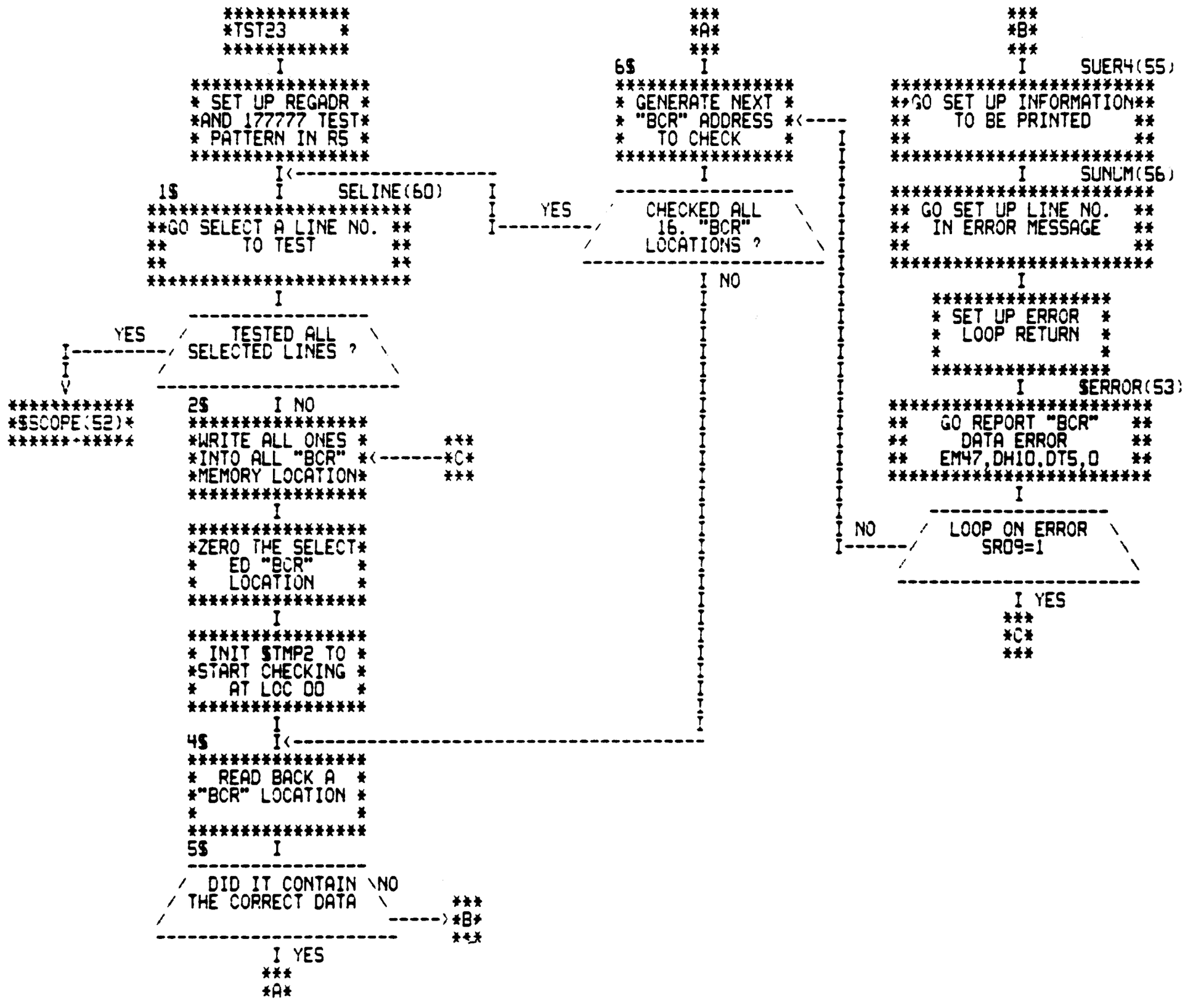


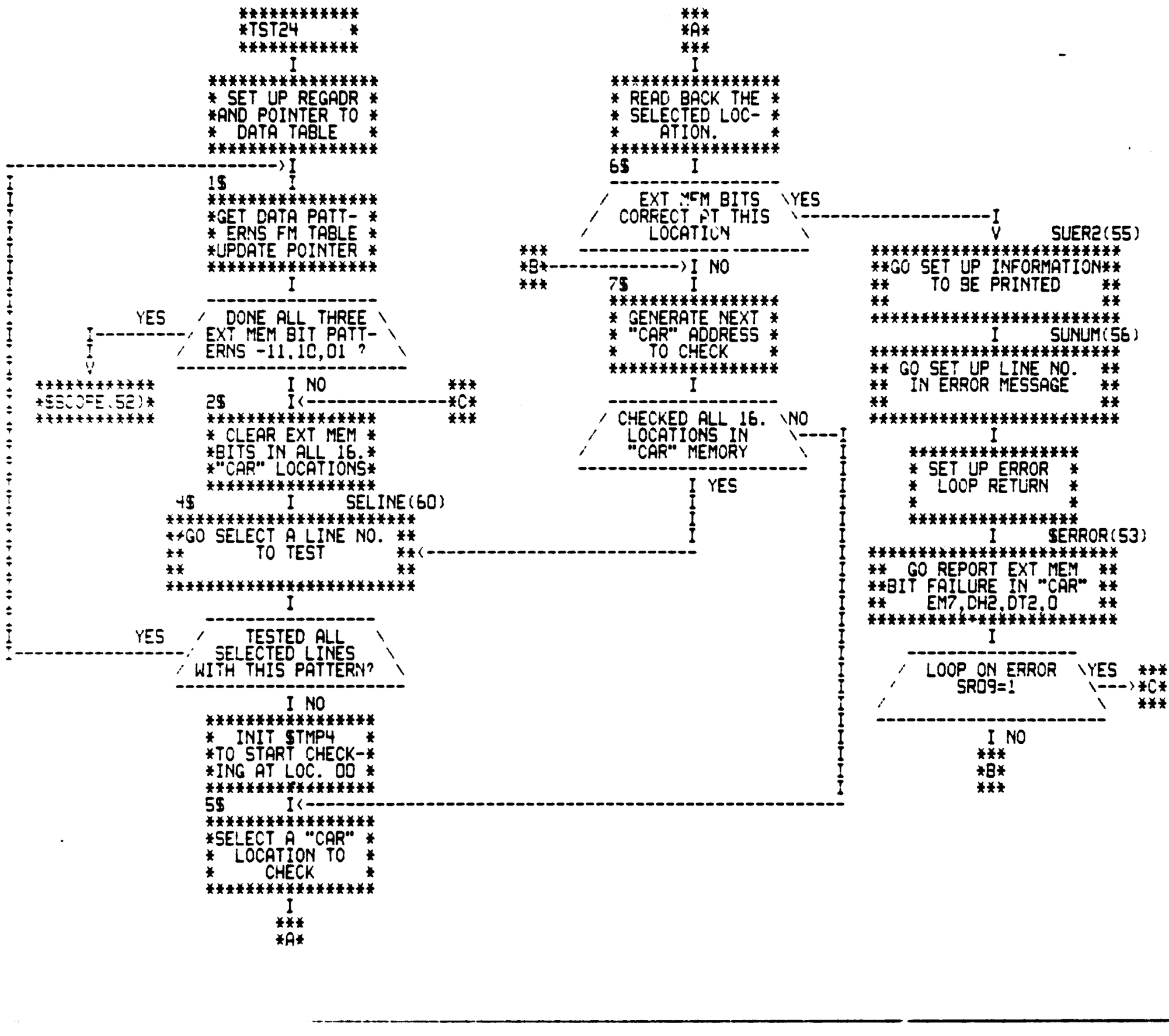
F06

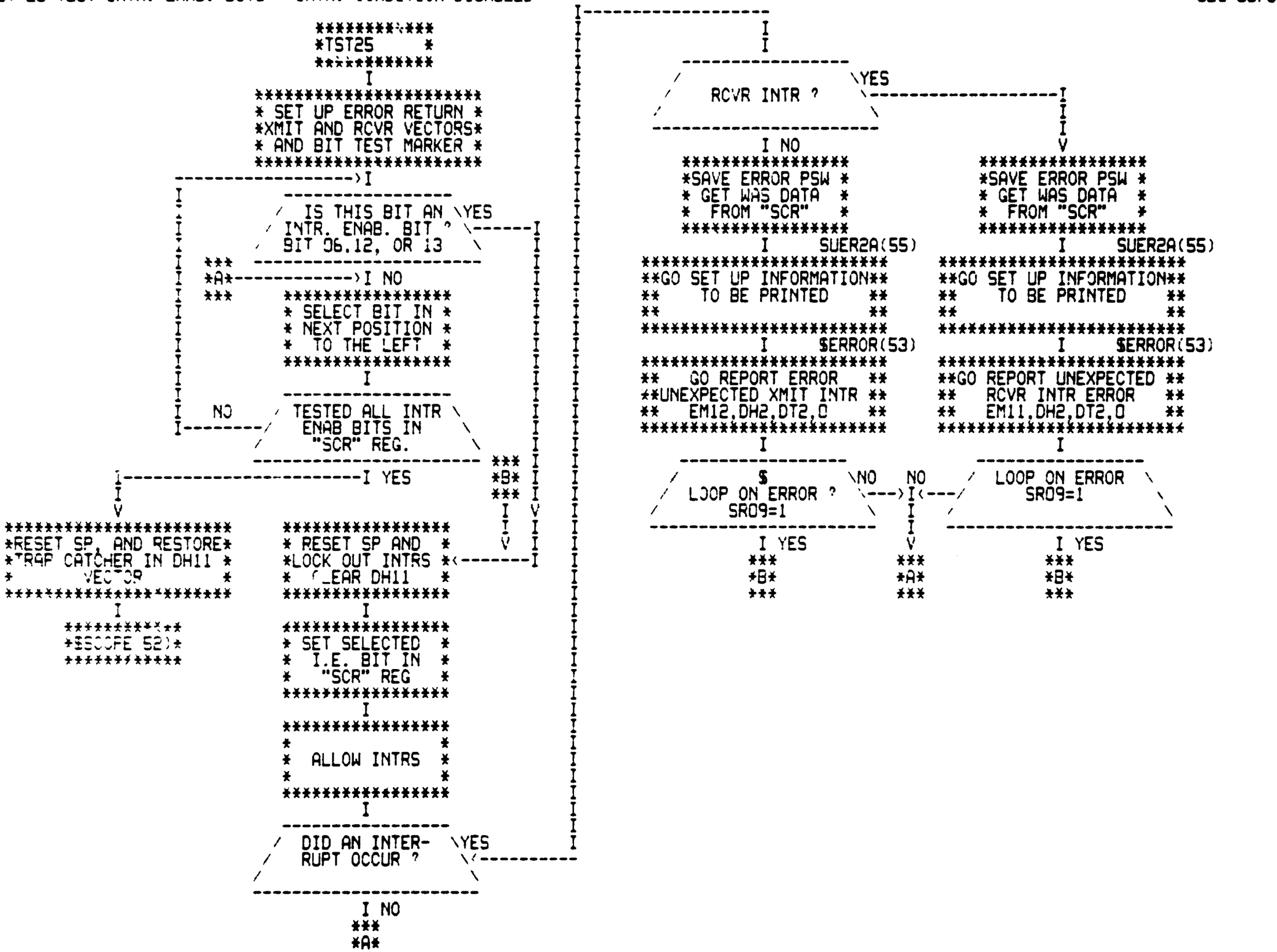

*TST20
SET UP REGADR AND POINTER TO DATA TABLE
1$
GET DATA PATTERN AND UPDATE TABLE POINTER
DONE ALL THREE PATTERNS (177777, 125252 AND 052525)
YES
$SCOPE(52)
11$
SELINE(60)
GO SELECT A LINE NO TO TEST WITH CURRENT PATTERN IN $TMP1
TESTED ALL SELECTED LINES WITH THIS PATTERN ?
2$
CLEAR ALL 16. LOCATIONS IN THE "CAR" MEM
WRITE TEST PATTERN INTO SELECTED "CAR"
INIT $TMP2 TO START CHECKING AT LINE 00
A
4$
SELECT A LINE TO CHECK
5$
DOES LINE BEING CHECKED CONTAIN CORRECT DATA ?
NO
SUER4(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO IN ERROR MESSAGE
$ERROR(53)
GO REPORT "CAR" DATA ERROR EM46,DH10,CT5.0
LOOP ON ERROR ? SR09=1
YES
6$
GENERATE ADDRESS OF NEXT LINE TO CHECK
DONE CHECKING ALL LOCATIONS IN THE "CAR" MEM ?

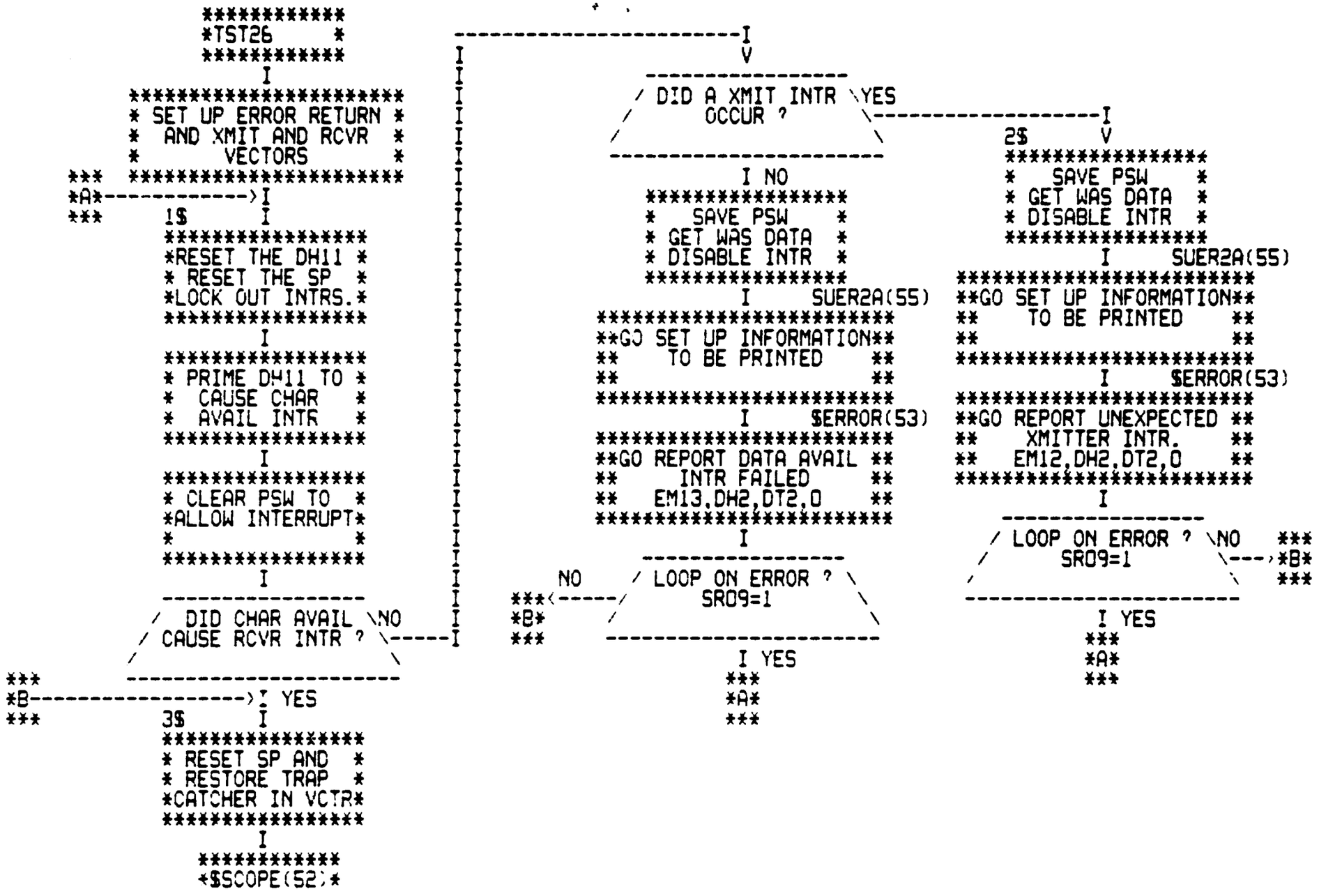

*TST21
SET UP REGADR AND POINTER TO DATA TABLE
1$
GET DATA PATTERN AND UPDATE TABLE POINTER
DONE ALL THREE PATTERNS (177777 /125252 AND 052525) ?
YES
$SCOPE,52)
NO
11$
SELINE(60)
GO SELECT A LINE NO TO TEST WITH CURRENT PATTERN IN $TMP1
TESTED ALL SELECTED LINES WITH THIS PATTERN ?
YES
NO
2$
CLEAR ALL 16. LOCATIONS IN THE "BCR" MEM
WRITE TEST PATTERN INTO SELECTED "BCR"
INIT $TMP2 TO START CHECKING AT LINE 00
A
4$
SELECT A LINE TO CHECK
DOES LINE BEING CHECKED CONTAIN CORRECT DATA ?
NO
SUER4(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO IN MESSAGE
YES
6$
GENERATE ADDRESS OF NEXT LINE TO CHECK
DONE CHECKING ALL LOCATIONS IN "BCR" MEMORY ?
YES
NO
A
SET UP ERROR LOOP RETURN
$ERROR(53)
GO REPORT "BCR" DATA ERROR EM47,DH10,DTS,0
LOOP ON ERROR SR09=1
INC
YES

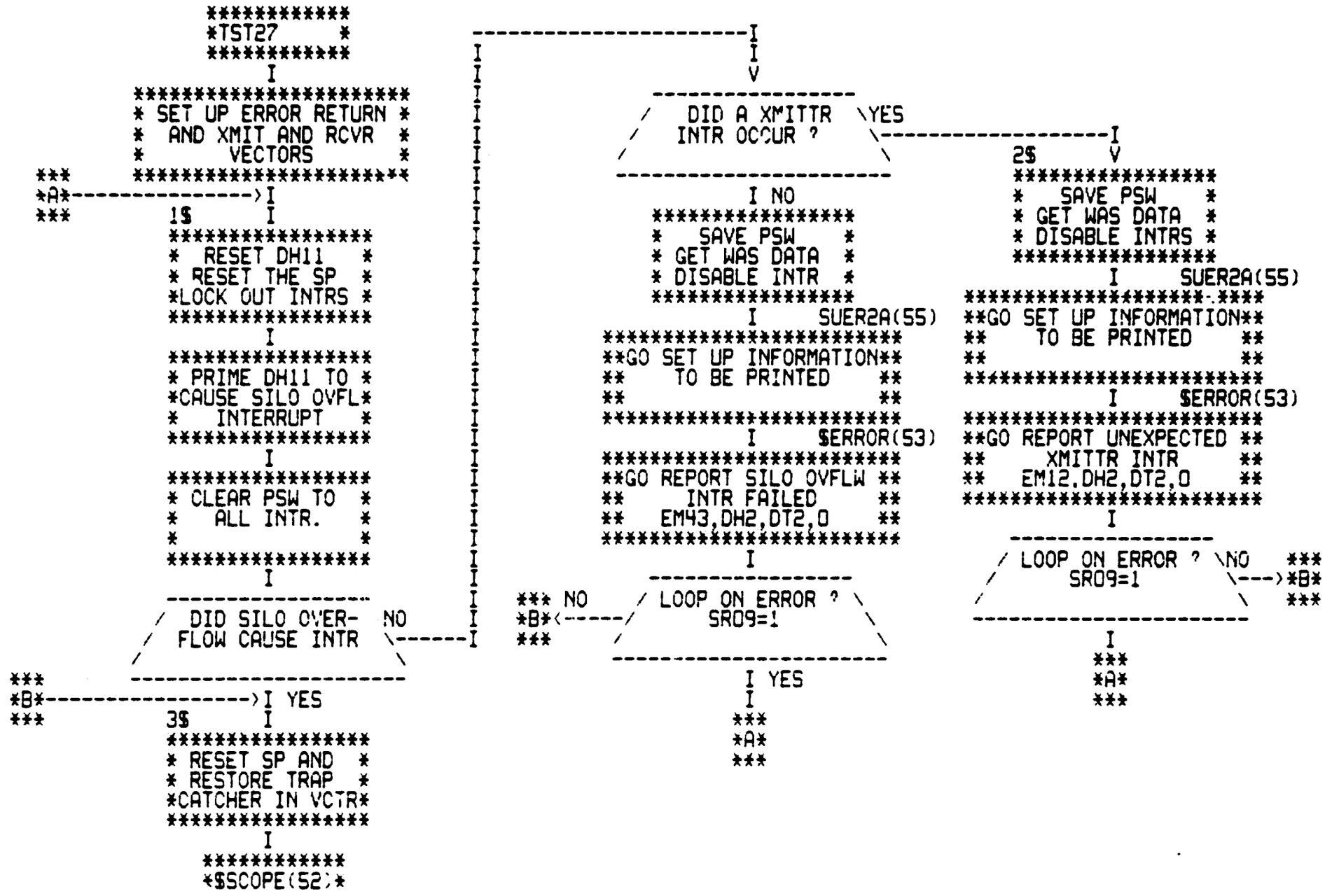

*TST22
SET UP REGADR *AND 177777 TEST* PATTERN IN R5
1$ SELINE(60)
GO SELECT A LINE NO TO TEST
TESTED ALL SELECTED LINES ?
YES
NO
$SCOPE(52)
2$
WRITE ALL ONES INTO ALL 16. "CAR" LOCATIONS
ZERO THE SELECTED "CAR" LOCATION
INIT $TMP2 TO START CHECKING AT LOC. 00
4$
READ BACK A "CAR" LOCATION
5$
DID IT CONTAIN THE CORRECT DATA
NO
YES
A
6$
GENERATE NEXT "CAR" ADDRESS TO CHECK
CHECKED ALL 16. "CAR" MEMORY LOCATIONS ?
NO
YES
B
SUER4(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO. IN ERROR MESSAGE
SET UP ERROR LOOP RETURN
$ERROR(53)
GO REPORT "CAR" DATA ERROR EM46,DH10,DT5.0
LOOP ON ERROR SR09=1
NO
YES
C

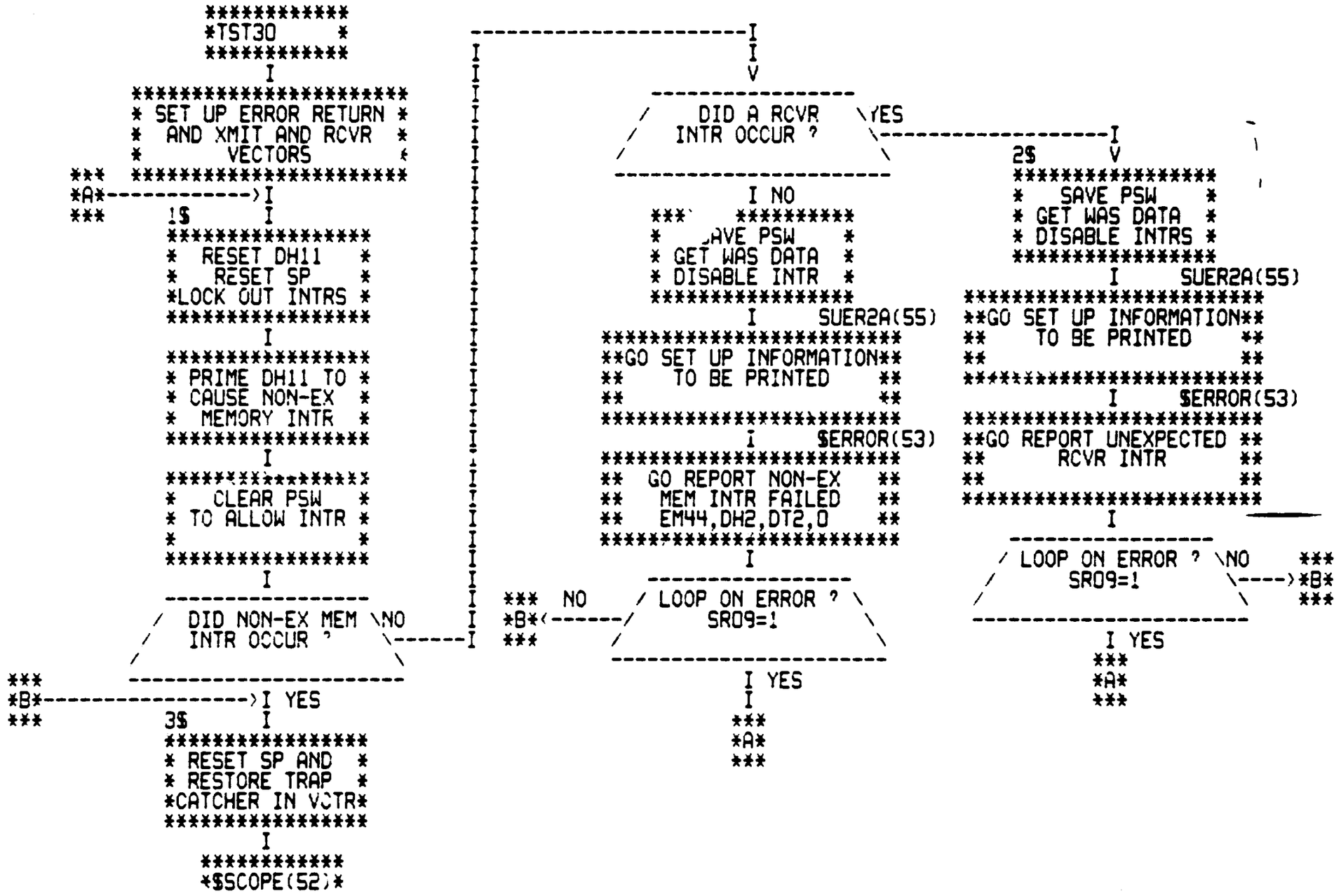

*TST23
SET UP REGADR *AND 177777 TEST* PATTERN IN R5
1$ SELINE(60)
GO SELECT A LINE NO. TO TEST
TESTED ALL SELECTED LINES ?
YES
$SCOPE(52)
2$ NO
WRITE ALL ONES *INTO ALL "BCR" *MEMORY LOCATION*
C
ZERO THE SELECTED "BCR" LOCATION
INIT $TMP2 TO START CHECKING AT LOC 00
4$
READ BACK A "BCR" LOCATION
5$
DID IT CONTAIN THE CORRECT DATA
NO
B
YES
A
6$
GENERATE NEXT "BCR" ADDRESS *TO CHECK
CHECKED ALL 16. "BCR" LOCATIONS ?
YES
NO
SUER4(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO. IN ERROR MESSAGE
SET UP ERROR LOOP RETURN
$ERROR(53)
GO REPORT "BCR" DATA ERROR
EM47,DH10,DT5,0
LOOP ON ERROR SR09=1
NO
YES
C

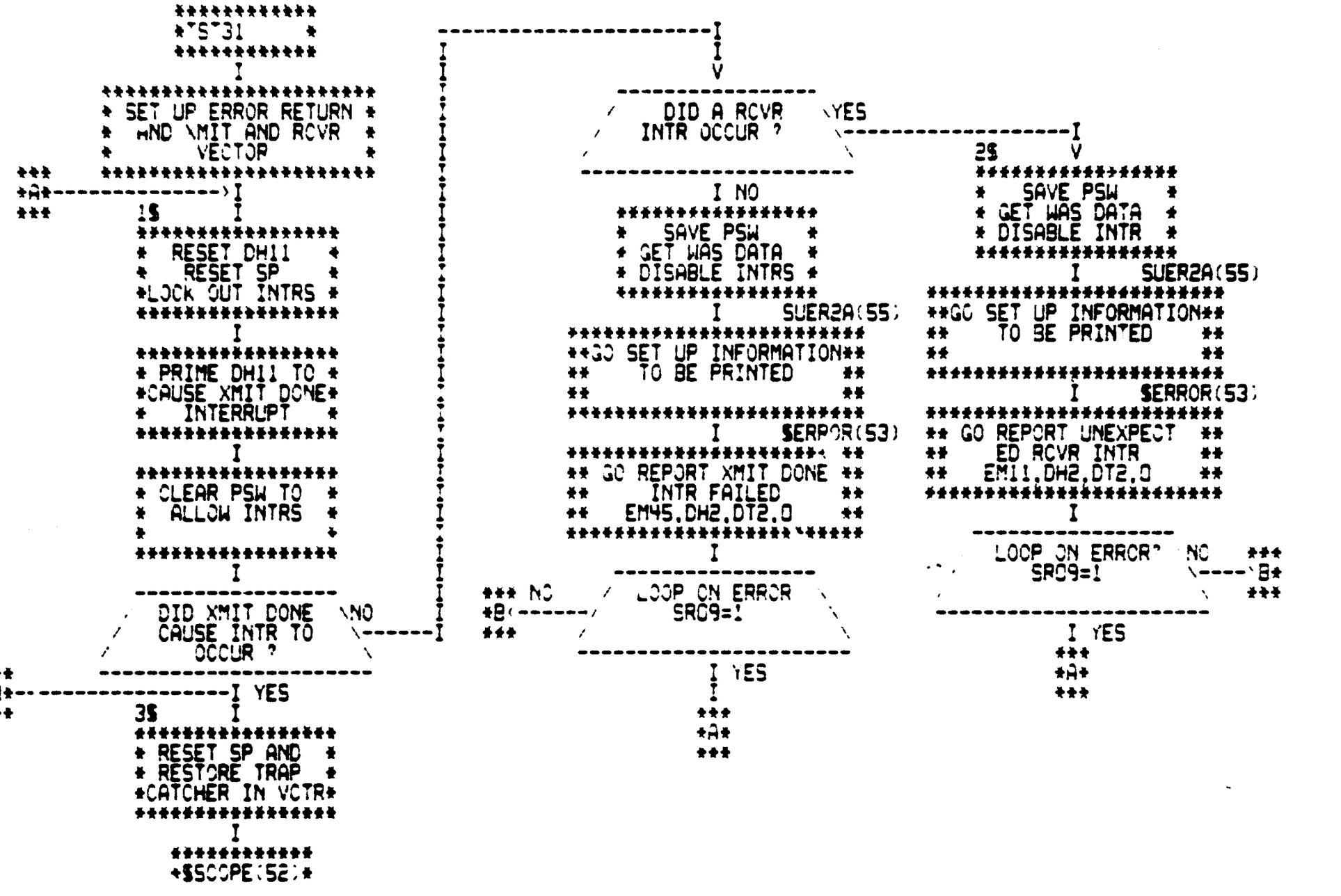

*TST24
SET UP REGADR AND POINTER TO DATA TABLE
1$
GET DATA PATT- ERNS FM TABLE UPDATE POINTER
DONE ALL THREE EXT MEM BIT PATT- ERNS -11,10,01 ?
YES
NO
SSCOPE,52)
2$
CLEAR EXT MEM BITS IN ALL 16. "CAR" LOCATIONS
4$
SELINE(60)
GO SELECT A LINE NO. TO TEST
TESTED ALL SELECTED LINES WITH THIS PATTERN?
INIT $TMP4 TO START CHECK-ING AT LOC. 00
5$
SELECT A "CAR" LOCATION TO CHECK
A
A
READ BACK THE SELECTED LOC- ATION.
6$
EXT MEM BITS CORRECT AT THIS LOCATION
NO
B
B
7$
GENERATE NEXT "CAR" ADDRESS TO CHECK
CHECKED ALL 16. LOCATIONS IN "CAR" MEMORY
YES
NO
SUER2(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO. IN ERROR MESSAGE
SET UP ERROR LOOP RETURN
$ERROR(53)
GO REPORT EXT MEM BIT FAILURE IN "CAR" EM7,DH2,DT2,0
LOOP ON ERROR SR09=1
YES
NO
C
C
B

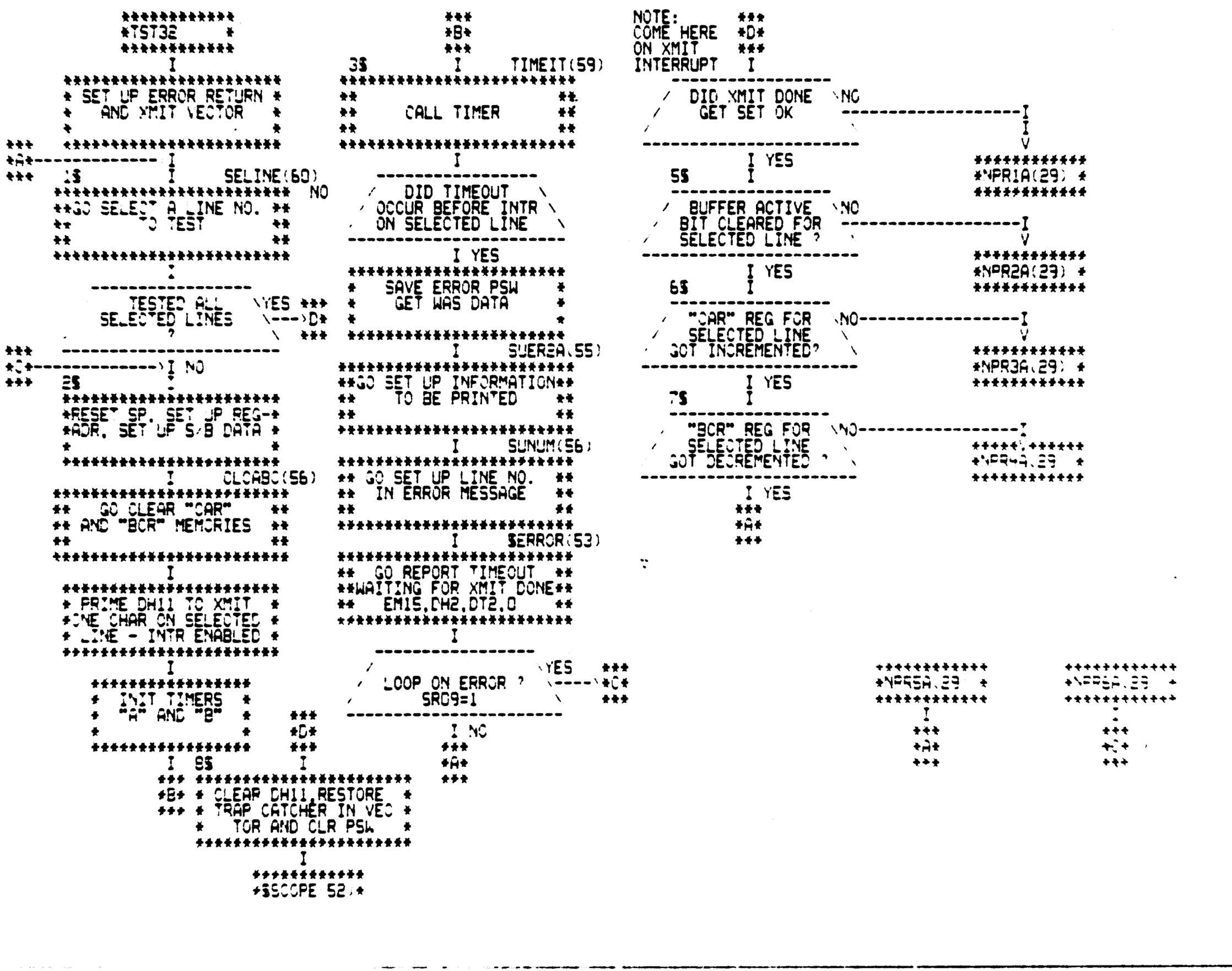

TST25
SET UP ERROR RETURN
XMIT AND RCVR VECTORS
AND BIT TEST MARKER
IS THIS BIT AN INTR. ENAB. BIT ? BIT 06,12, OR 13
YES
NO
A
SELECT BIT IN NEXT POSITION TO THE LEFT
TESTED ALL INTR ENAB BITS IN "SCR" REG.
NO
YES
B
RESET SP, AND RESTORE TRAP CATCHER IN DH11 VECTOR
SCOPE 52
RESET SP AND LOCK OUT INTRS
CLEAR DH11
SET SELECTED I.E. BIT IN "SCR" REG
ALLOW INTRS
DID AN INTERRUPT OCCUR ?
YES
NO
A
RCVR INTR ?
YES
NO
SAVE ERROR PSW
GET WAS DATA FROM "SCR"
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT ERROR
UNEXPECTED XMIT INTR
EM12,DH2,DT2,0
GO REPORT UNEXPECTED RCVR INTR ERROR
EM11,DH2,DT2,0
LOOP ON ERROR SR09=1
$
NO
YES
B
A

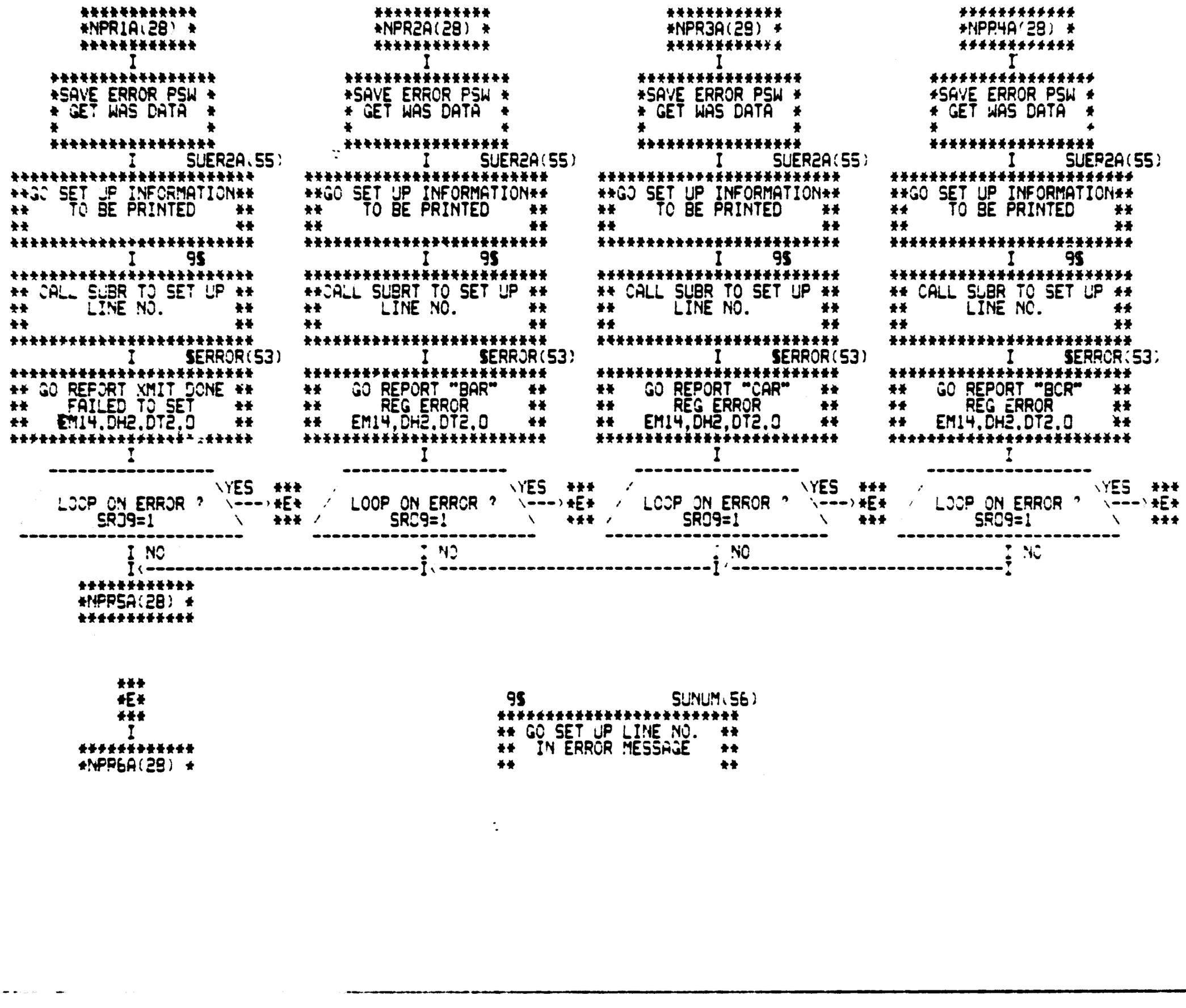

*TST26
SET UP ERROR RETURN AND XMIT AND RCVR VECTORS
1$
RESET THE DH11 RESET THE SP LOCK OUT INTRS.
PRIME DH11 TO CAUSE CHAR AVAIL INTR
CLEAR PSW TO ALLOW INTERRUPT
DID CHAR AVAIL CAUSE RCVR INTR ?
NO
YES
3$
RESET SP AND RESTORE TRAP CATCHER IN VCTR
$SCOPE(52)
DID A XMIT INTR OCCUR ?
YES
2$
SAVE PSW GET WAS DATA DISABLE INTR
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT UNEXPECTED XMITTER INTR. EM12,DH2,DT2,0
LOOP ON ERROR ? SR09=1
A
B
***

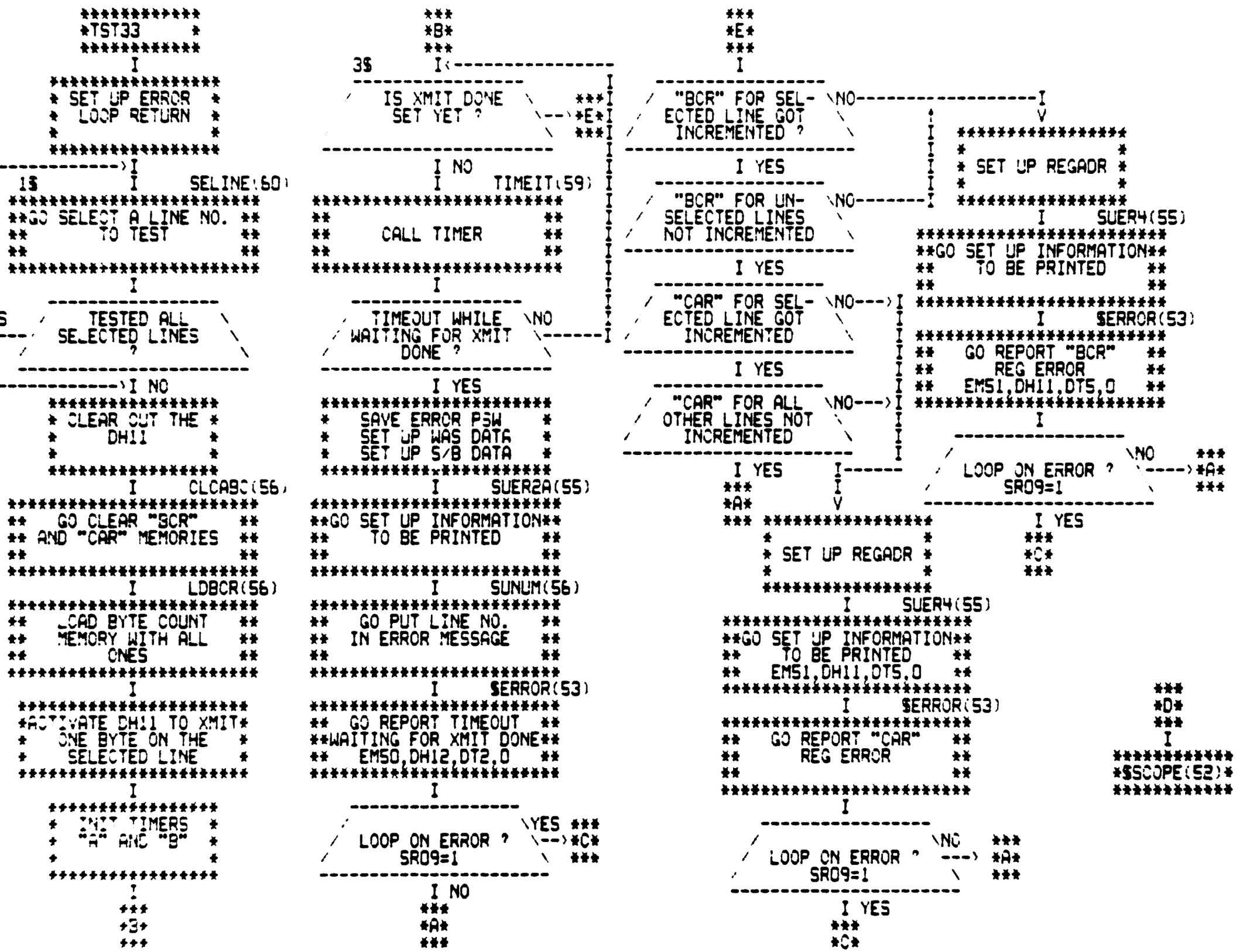

TST27
SET UP ERROR RETURN AND XMIT AND RCVR VECTORS
1$
RESET DH11
RESET THE SP
LOCK OUT INTRS
PRIME DH11 TO CAUSE SILO OVFL INTERRUPT
CLEAR PSW TO ALL INTR.
DID SILO OVER-FLOW CAUSE INTR
NO
YES
3$
RESET SP AND RESTORE TRAP CATCHER IN VCTR
$SCOPE(52);
DID A XMITTR INTR OCCUR ?
NO
YES
2$
SAVE PSW
GET WAS DATA
DISABLE INTR
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT SILO OVFLW INTR FAILED
EM43,DH2,DT2,0
LOOP ON ERROR ? SR09=1
SAVE PSW
GET WAS DATA
DISABLE INTRS
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT UNEXPECTED XMITTR INTR
EM12,DH2,DT2,0
LOOP ON ERROR ? SR09=1
A
B

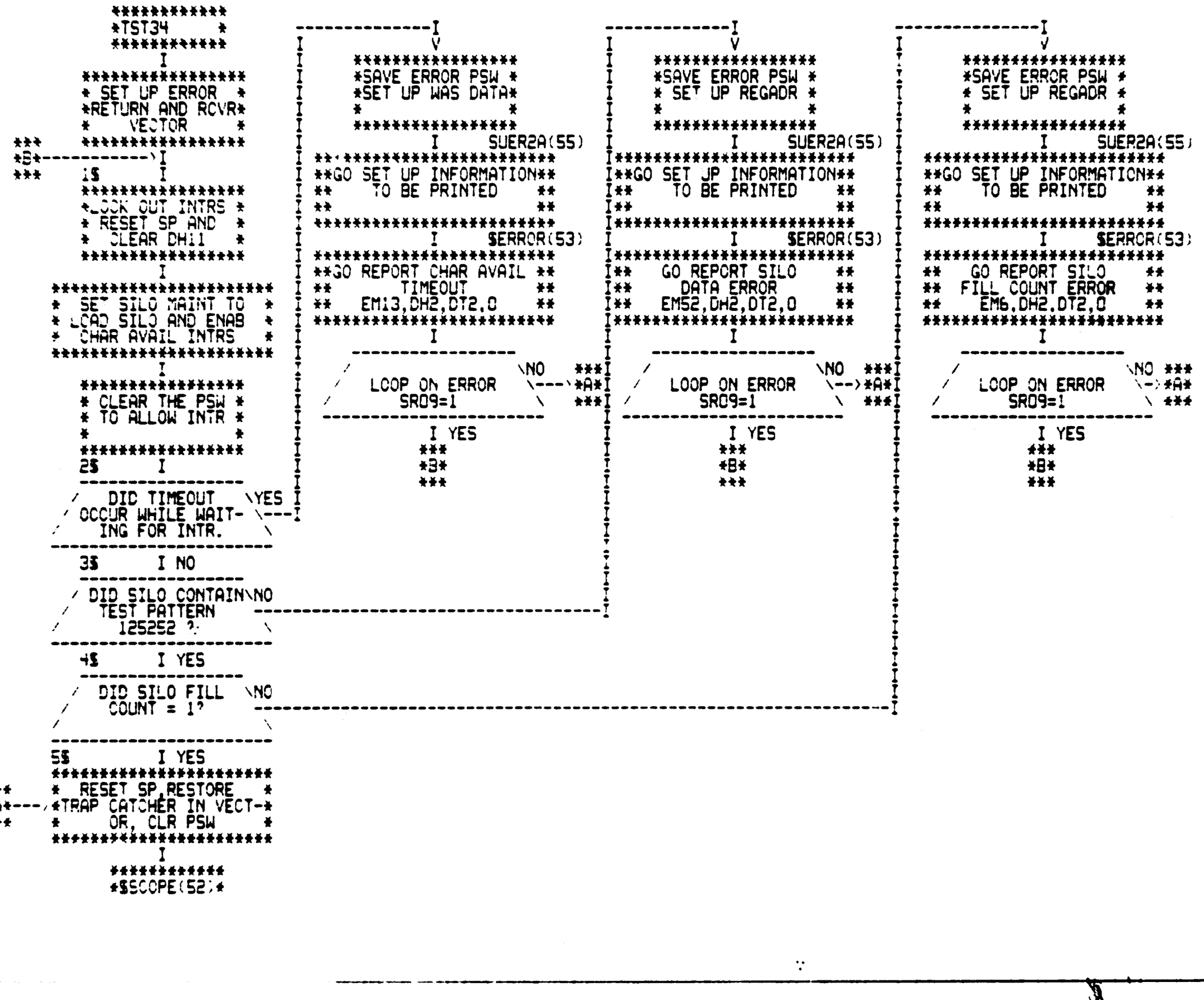

*TST30
SET UP ERROR RETURN AND XMIT AND RCVR VECTORS
A
1$
RESET DH11
RESET SP
LOCK OUT INTRS
PRIME DH11 TO CAUSE NON-EX MEMORY INTR
CLEAR PSW TO ALLOW INTR
DID NON-EX MEM INTR OCCUR ?
NO
B
3$
YES
RESET SP AND RESTORE TRAP CATCHER IN VCTR
$SCOPE(52)
DID A RCVR INTR OCCUR ?
YES
2$
SAVE PSW GET WAS DATA DISABLE INTRS
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT UNEXPECTED RCVR INTR
LOOP ON ERROR ? SR09=1
NO
B
YES
A
SAVE PSW GET WAS DATA DISABLE INTR
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT NON-EX MEM INTR FAILED EM44,DH2,DT2,0
LOOP ON ERROR ? SR09=1
NO
B
YES
A


TST31
SET UP ERROR RETURN AND XMIT AND RCVR VECTOR
1$
RESET DH11
RESET SP
LOCK OUT INTRS
PRIME DH11 TO CAUSE XMIT DONE INTERRUPT
CLEAR PSW TO ALLOW INTRS
DID XMIT DONE CAUSE INTR TO OCCUR ?
NO
YES
3$
RESET SP AND RESTORE TRAP CATCHER IN VCTR
$SCOPE(52)
DID A RCVR INTR OCCUR ?
YES
NO
SAVE PSW
GET WAS DATA
DISABLE INTRS
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
SERROR(53)
GO REPORT XMIT DONE INTR FAILED
EM45,DH2,DT2,0
LOOP ON ERROR SR09=1
2$
SAVE PSW
GET WAS DATA
DISABLE INTR
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
SERROR(53)
GO REPORT UNEXPECT ED RCVR INTR
EM11,DH2,DT2,0
LOOP ON ERROR SR09=1
NO
YES
A
B

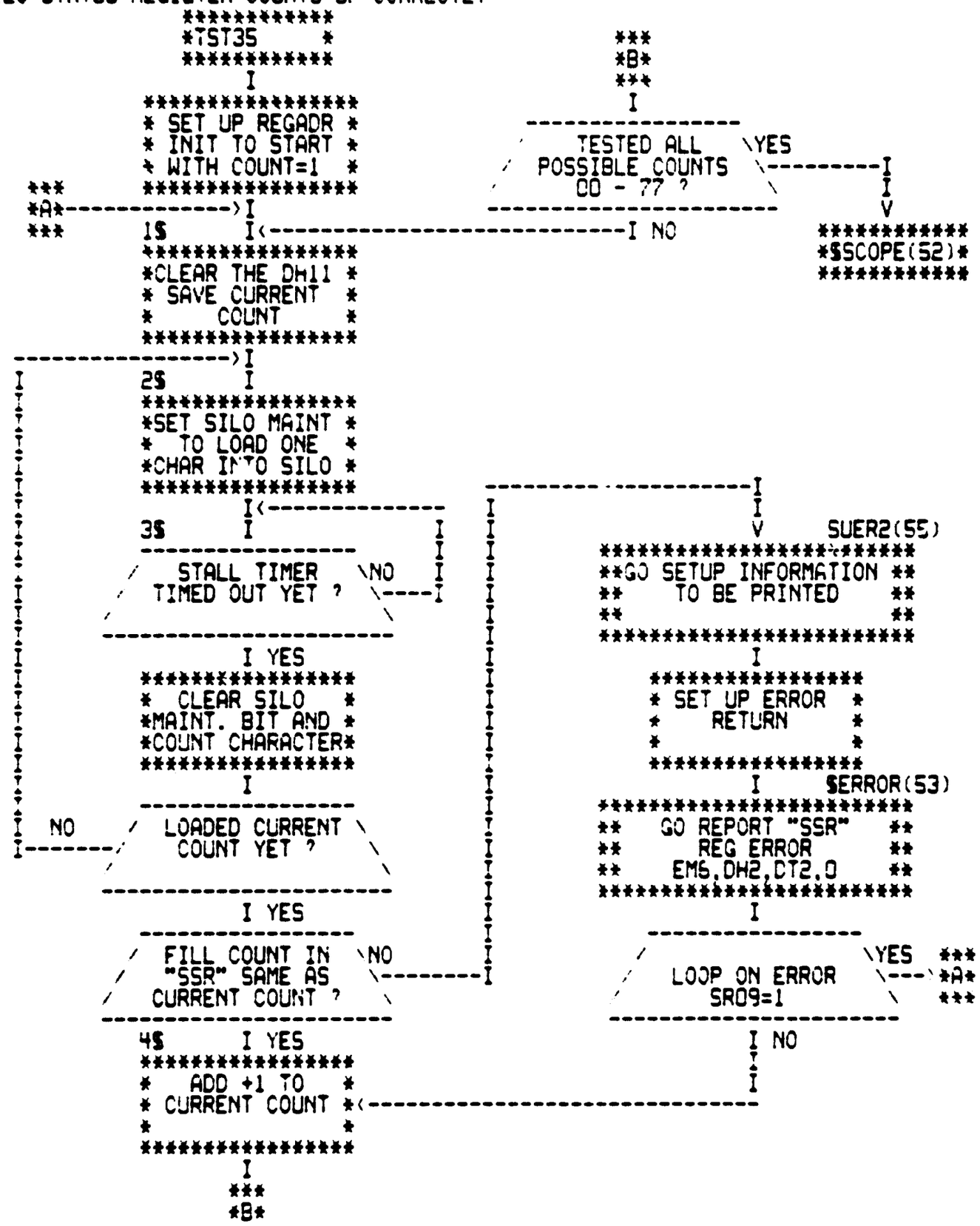

TST32
SET UP ERROR RETURN AND XMIT VECTOR
1$
SELINE(60)
NO
GO SELECT A LINE NO. TO TEST
TESTED ALL SELECTED LINES
YES
D
2$
NO
RESET SP, SET UP REG-ADR, SET UP S/B DATA
CLCABC(56)
GO CLEAR "CAR" AND "BCR" MEMORIES
PRIME DH11 TO XMIT ONE CHAR ON SELECTED LINE - INTR ENABLED
INIT TIMERS "A" AND "B"
8$
CLEAR DH11,RESTORE TRAP CATCHER IN VECTOR AND CLR PSW
$SCOPE 52
B
3$
TIMEIT(59)
CALL TIMER
DID TIMEOUT OCCUR BEFORE INTR ON SELECTED LINE
YES
SAVE ERROR PSW
GET WAS DATA
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO. IN ERROR MESSAGE
$ERROR(53)
GO REPORT TIMEOUT WAITING FOR XMIT DONE
EM15,DH2,DT2,0
LOOP ON ERROR ? SR09=1
YES
C
NO
A
NOTE: COME HERE ON XMIT INTERRUPT
D
DID XMIT DONE GET SET OK
NO
5$
BUFFER ACTIVE BIT CLEARED FOR SELECTED LINE ?
NO
6$
"CAR" REG FOR SELECTED LINE GOT INCREMENTED?
NO
7$
"BCR" REG FOR SELECTED LINE GOT DECREMENTED ?
NO
YES
NPR1A(29)
NPR2A(29)
NPR3A(29)
NPR4A(29)
NPR5A(29)
NPR6A(29)

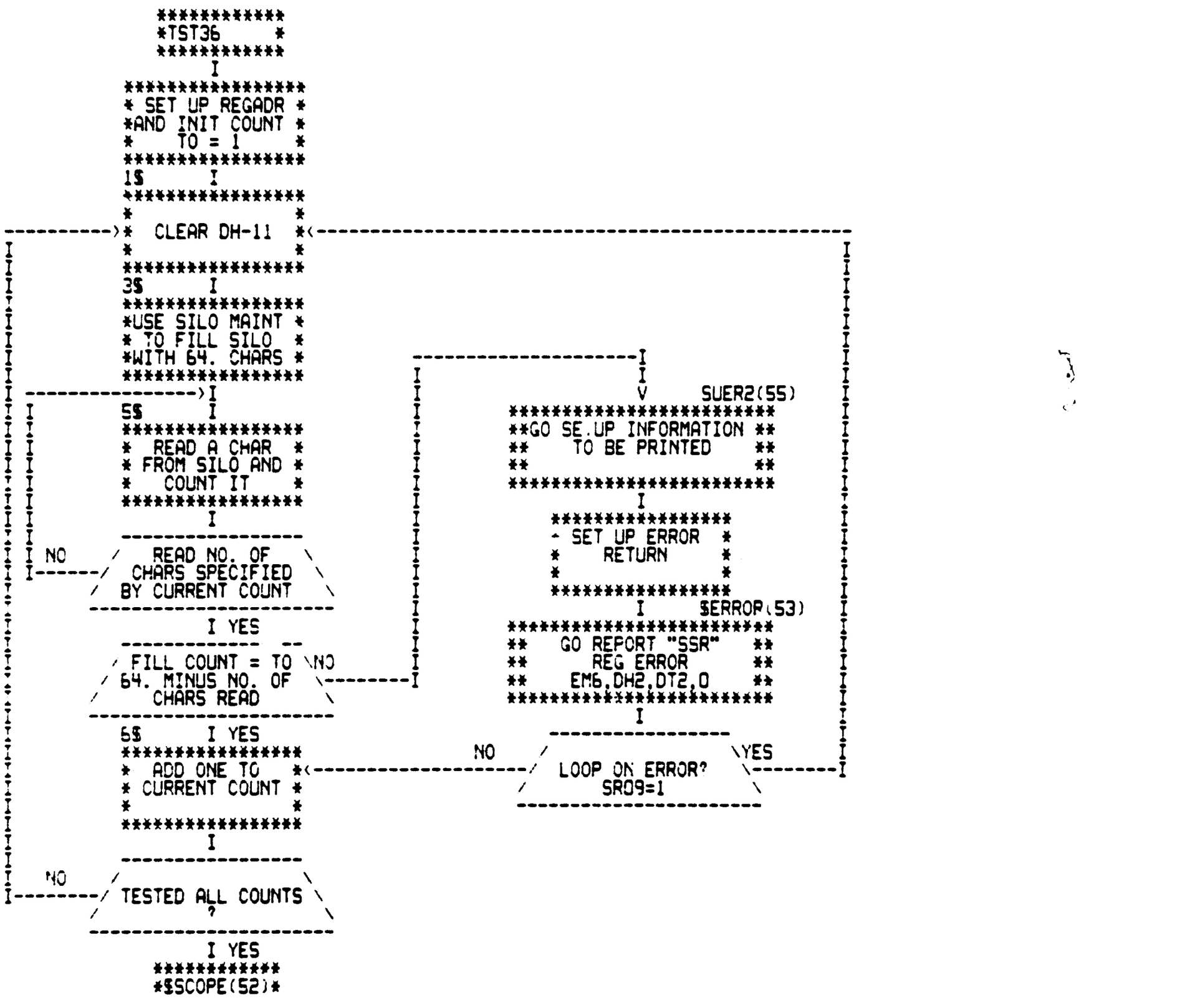

*NPR1A(28) *
*SAVE ERROR PSW *
* GET WAS DATA *
SUER2A(55)
GO SET UP INFORMATION
** TO BE PRINTED **
9$
** CALL SUBR TO SET UP **
** LINE NO. **
$ERROR(53)
** GO REPORT XMIT DONE **
** FAILED TO SET **
** EM14,DH2,DT2,0 **
LOOP ON ERROR ?
SR09=1
YES
E
NO
*NPR2A(28) *
** GO REPORT "BAR" **
*NPR3A(29) *
** GO REPORT "CAR" **
*NPR4A(28) *
** GO REPORT "BCR" **
*NPR5A(28) *
*NPR6A(28) *
SUNUM(56)
** GO SET UP LINE NO. **
** IN ERROR MESSAGE **

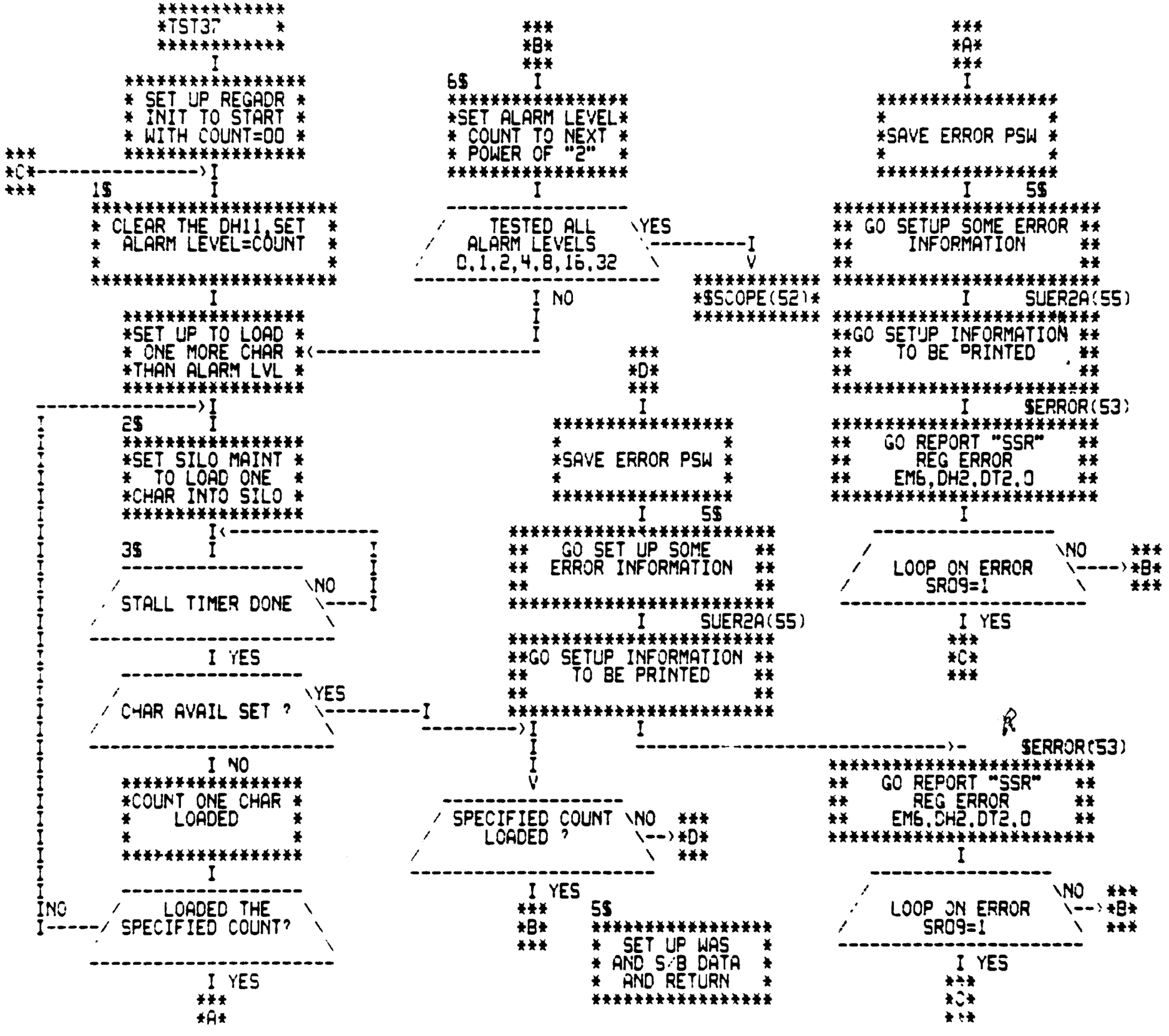

TST33
SET UP ERROR LOOP RETURN
1$
SELINE(60)
GO SELECT A LINE NO. TO TEST
TESTED ALL SELECTED LINES ?
YES
2$
NO
CLEAR OUT THE DH11
CLCABC(56)
GO CLEAR "SCR" AND "CAR" MEMORIES
LDBCR(56)
LOAD BYTE COUNT MEMORY WITH ALL ONES
ACTIVATE DH11 TO XMIT ONE BYTE ON THE SELECTED LINE
INIT TIMERS "A" AND "B"
3$
B
IS XMIT DONE SET YET ?
NO
TIMEIT(59)
CALL TIMER
TIMEOUT WHILE WAITING FOR XMIT DONE ?
YES
SAVE ERROR PSW
SET UP WAS DATA
SET UP S/B DATA
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN ERROR MESSAGE
$ERROR(53)
GO REPORT TIMEOUT WAITING FOR XMIT DONE
EM50,DH12,DT2,0
LOOP ON ERROR ? SR09=1
YES
NO
C
A
E
"BCR" FOR SELECTED LINE GOT INCREMENTED ?
NO
YES
"BCR" FOR UNSELECTED LINES NOT INCREMENTED
"CAR" FOR SELECTED LINE GOT INCREMENTED
"CAR" FOR ALL OTHER LINES NOT INCREMENTED
SET UP REGADR
SUER4(55)
GO SET UP INFORMATION TO BE PRINTED
EM51,DH11,DT5,0
GO REPORT "BCR" REG ERROR
GO REPORT "CAR" REG ERROR
D
$SCOPE(52)

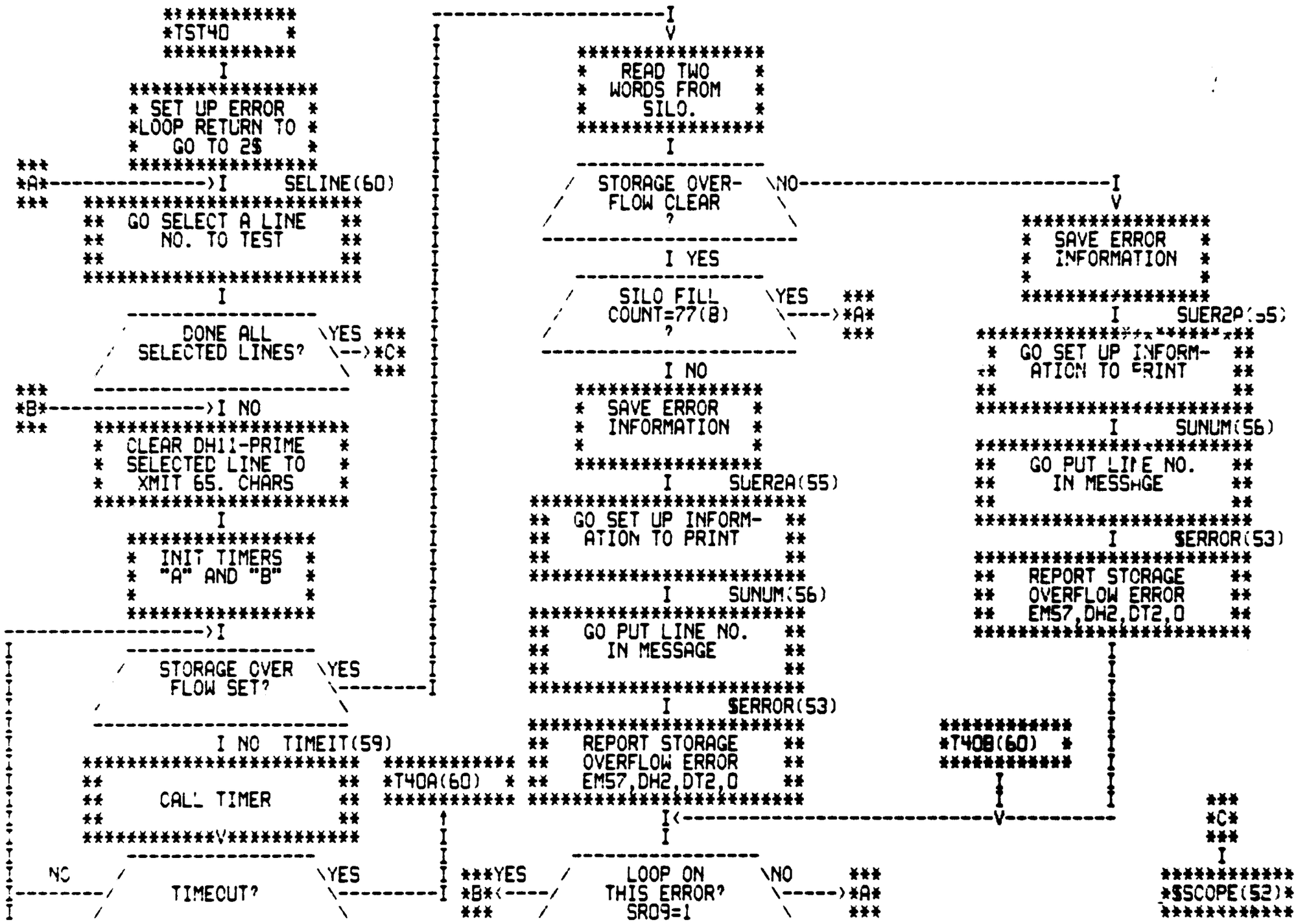

*TST34
SET UP ERROR RETURN AND RCVR VECTOR
B
1$
LOCK OUT INTRS RESET SP AND CLEAR DH11
SET SILO MAINT TO LOAD SILO AND ENAB CHAR AVAIL INTRS
CLEAR THE PSW TO ALLOW INTR
2$
DID TIMEOUT OCCUR WHILE WAIT-ING FOR INTR.
YES
3$
NO
DID SILO CONTAIN TEST PATTERN 125252 ?
NO
4$
YES
DID SILO FILL COUNT = 1?
NO
5$
YES
RESET SP,RESTORE *TRAP CATCHER IN VECT-OR, CLR PSW
A
$SCOPE(52)
SAVE ERROR PSW
SET UP WAS DATA
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT CHAR AVAIL TIMEOUT
EM13,DH2,DT2,0
LOOP ON ERROR SR09=1
YES
3
SAVE ERROR PSW
SET UP REGADR
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT SILO DATA ERROR
EM52,DH2,DT2,0
LOOP ON ERROR SR09=1
YES
B
SAVE ERROR PSW
SET UP REGADR
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
$ERROR(53)
GO REPORT SILO FILL COUNT ERROR
EM6,DH2,DT2,0
LOOP ON ERROR SR09=1
YES
B

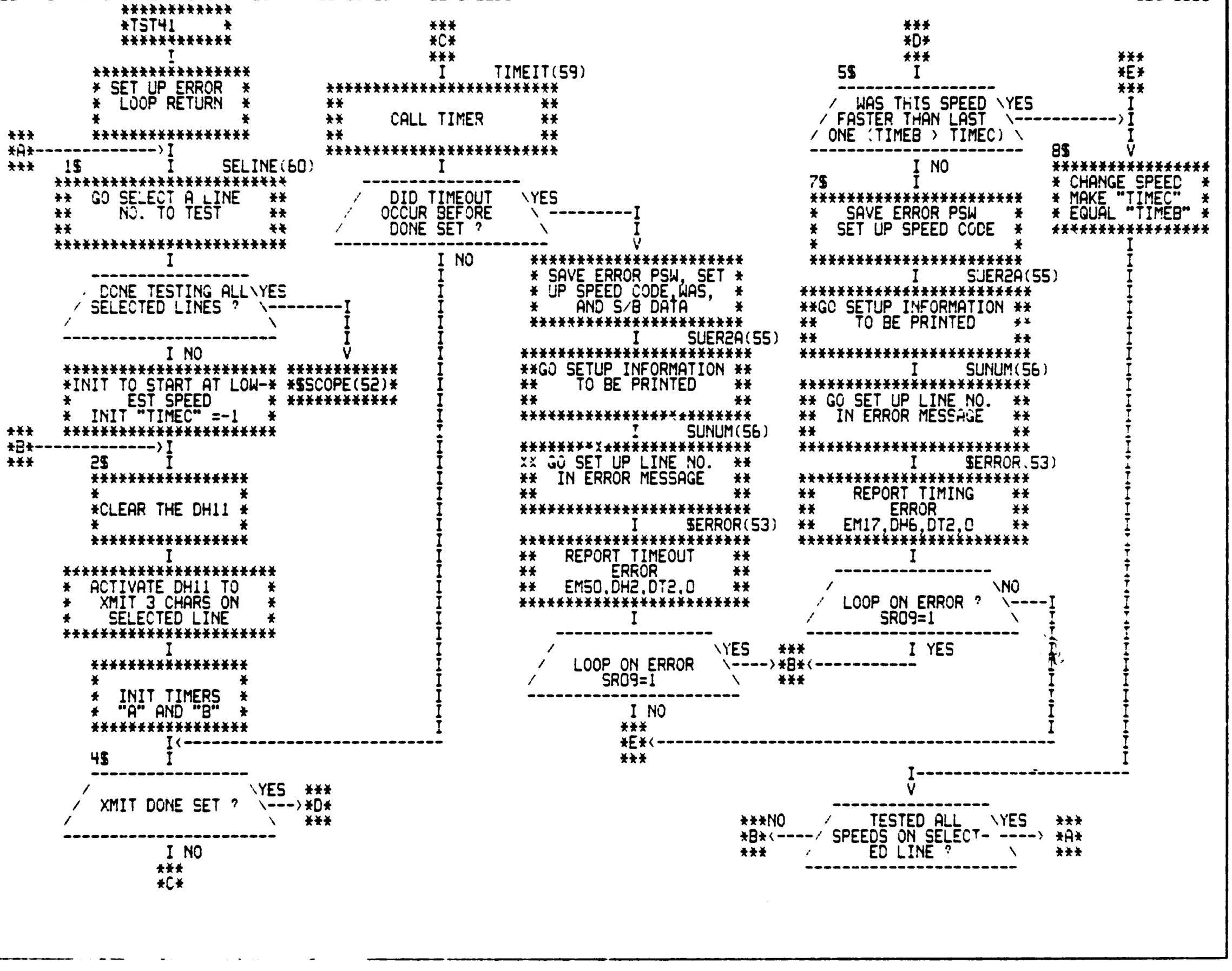
G07

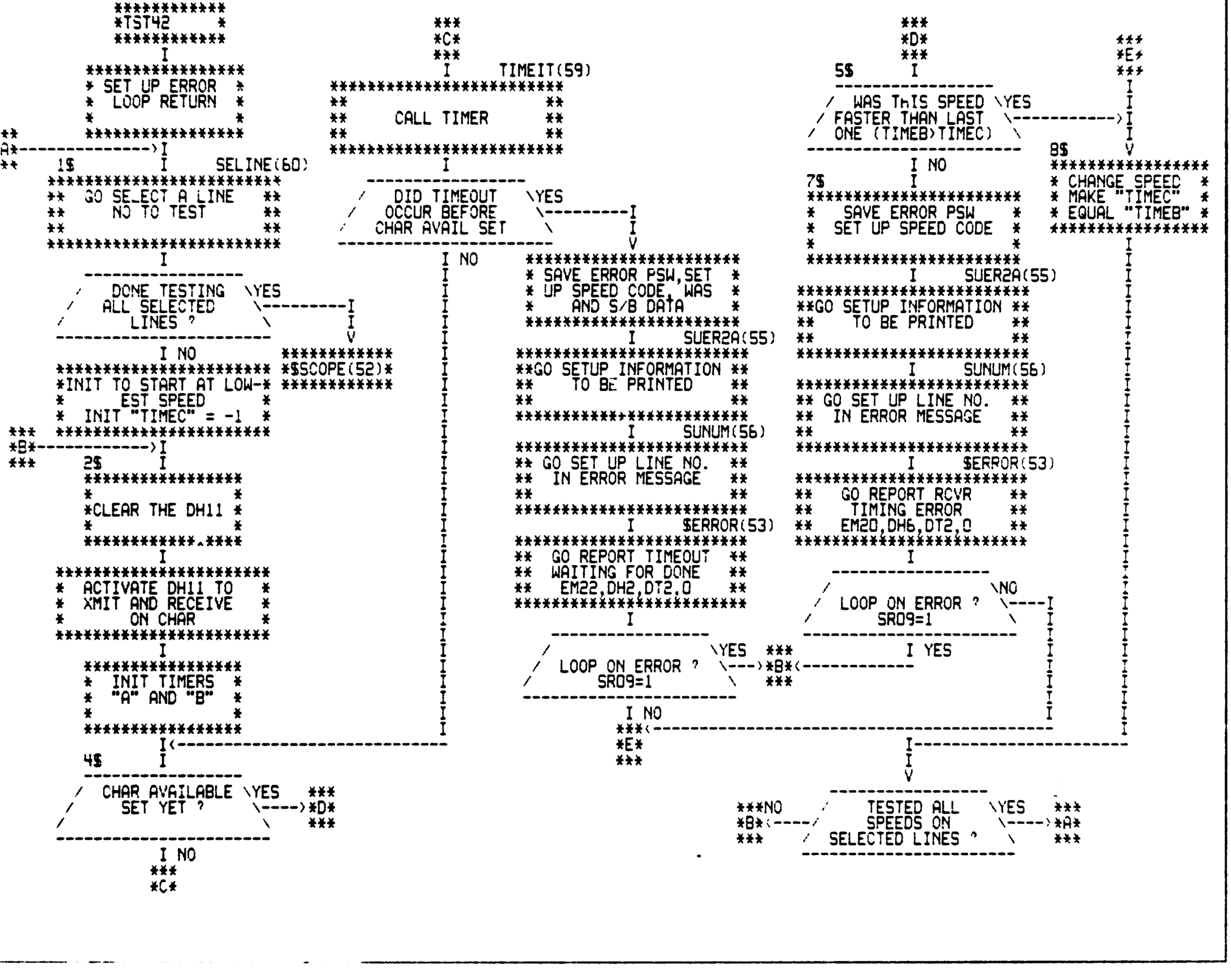

*TST35
SET UP REGADR
INIT TO START
WITH COUNT=1
1$
CLEAR THE DH11
SAVE CURRENT
COUNT
2$
SET SILO MAINT
TO LOAD ONE
CHAR INTO SILO
3$
STALL TIMER
TIMED OUT YET ?
NO
YES
CLEAR SILO
MAINT. BIT AND
COUNT CHARACTER
LOADED CURRENT
COUNT YET ?
NO
YES
FILL COUNT IN
"SSR" SAME AS
CURRENT COUNT ?
NO
4$
YES
ADD +1 TO
CURRENT COUNT
B
A
TESTED ALL
POSSIBLE COUNTS
00 - 77 ?
YES
NO
$SCOPE(52)
SUER2(55)
GO SETUP INFORMATION
TO BE PRINTED
SET UP ERROR
RETURN
$ERROR(53)
GO REPORT "SSR"
REG ERROR
EM6,DH2,CT2,0
LOOP ON ERROR
SR09=1
YES
NO

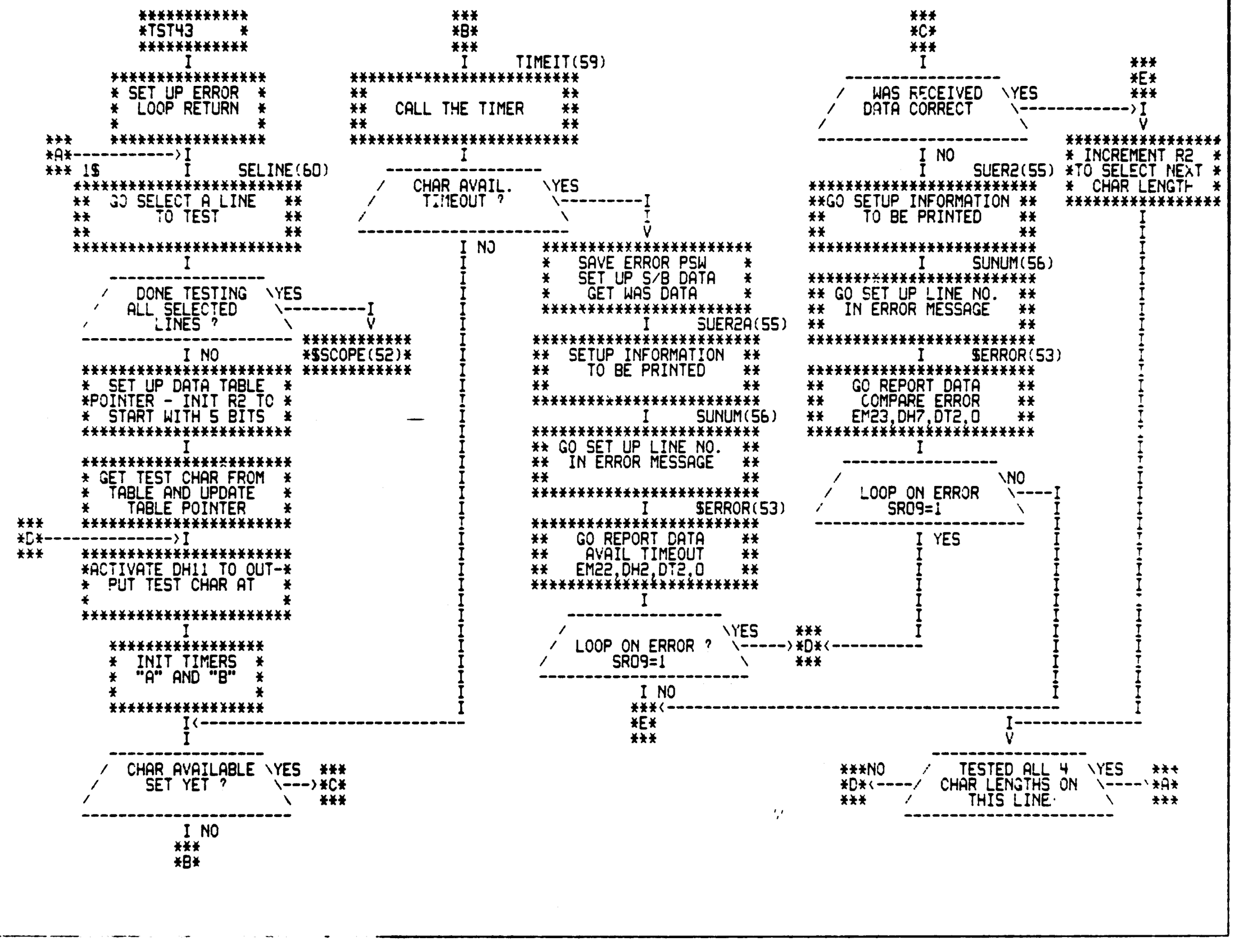

*TST36
SET UP REGADR AND INIT COUNT TO = 1
1$
CLEAR DH-11
3$
USE SILO MAINT TO FILL SILO WITH 64. CHARS
5$
READ A CHAR FROM SILO AND COUNT IT
NO
READ NO. OF CHARS SPECIFIED BY CURRENT COUNT
YES
FILL COUNT = TO 64. MINUS NO. OF CHARS READ
NO
6$
YES
ADD ONE TO CURRENT COUNT
NO
TESTED ALL COUNTS ?
YES
$SCOPE(52)
SUER2(55)
GO SE.UP INFORMATION TO BE PRINTED
SET UP ERROR RETURN
$ERROR(53)
GO REPORT "SSR" REG ERROR EM6,DH2,DT2,0
NO
LOOP ON ERROR? SR09=1
YES

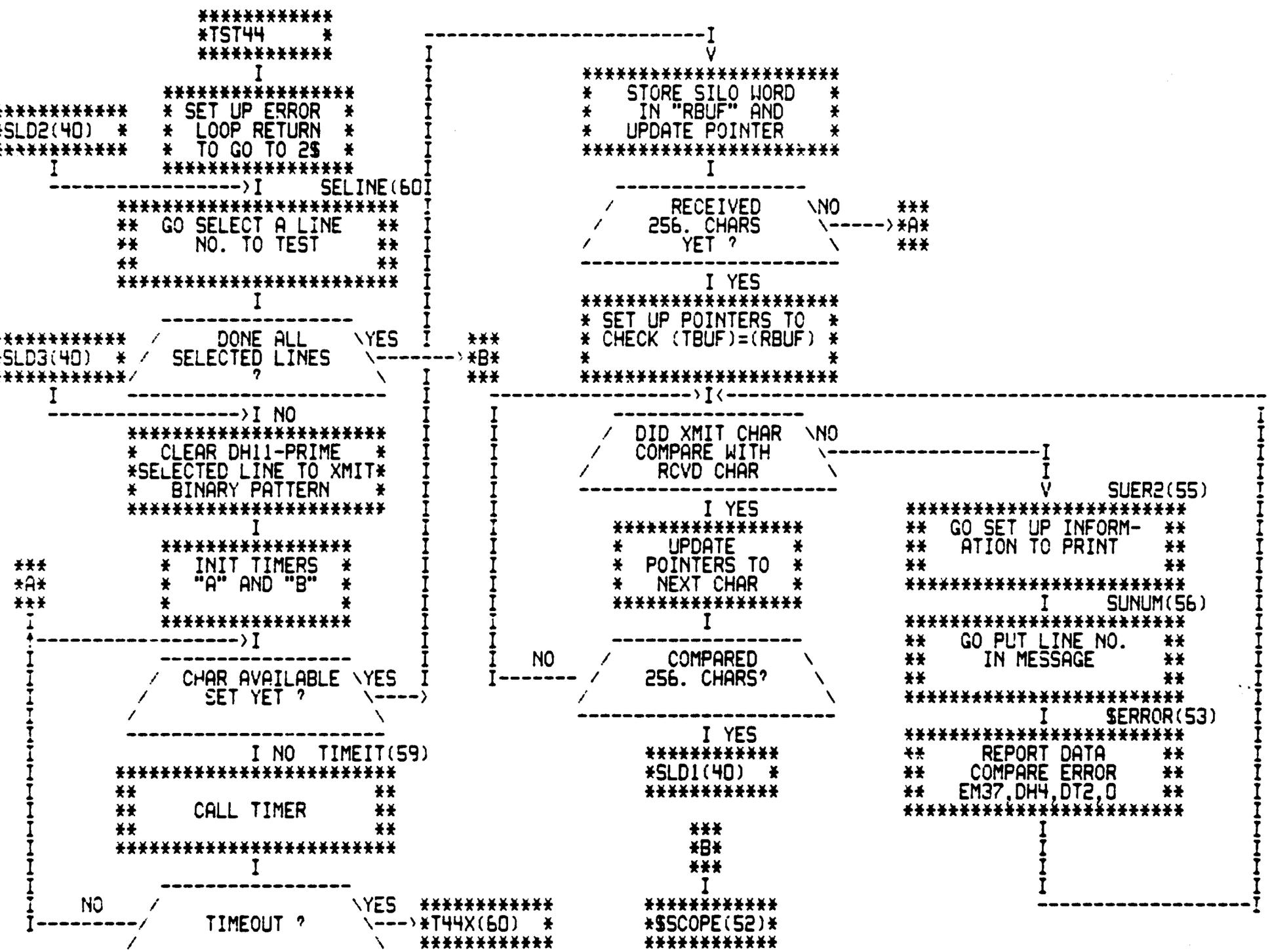

TST37
SET UP REGADR
INIT TO START
WITH COUNT=00
C
1$
CLEAR THE DH11,SET
ALARM LEVEL=COUNT
SET UP TO LOAD
ONE MORE CHAR
THAN ALARM LVL
2$
SET SILO MAINT
TO LOAD ONE
CHAR INTO SILO
3$
STALL TIMER DONE
NO
YES
CHAR AVAIL SET ?
NO
COUNT ONE CHAR
LOADED
LOADED THE
SPECIFIED COUNT?
NO
YES
A
B
6$
SET ALARM LEVEL
COUNT TO NEXT
POWER OF "2"
TESTED ALL
ALARM LEVELS
0,1,2,4,8,16,32
YES
NO
$SCOPE(52)
D
SAVE ERROR PSW
5$
GO SET UP SOME
ERROR INFORMATION
SUER2A(55)
GO SETUP INFORMATION
TO BE PRINTED
SPECIFIED COUNT
LOADED ?
NO
YES
5$
SET UP WAS
AND S/B DATA
AND RETURN
A
SAVE ERROR PSW
GO SET SOME ERROR
INFORMATION
SUER2A(55)
GO SETUP INFORMATION
TO BE PRINTED
$ERROR(53)
GO REPORT "SSR"
REG ERROR
EM6,DH2,DT2,0
LOOP ON ERROR
SR09=1
NO
YES
B
C
$ERROR(53)
GO REPORT "SSR"
REG ERROR
EM6,DH2,DT2,0
LOOP ON ERROR
SR09=1
NO
YES
B
C

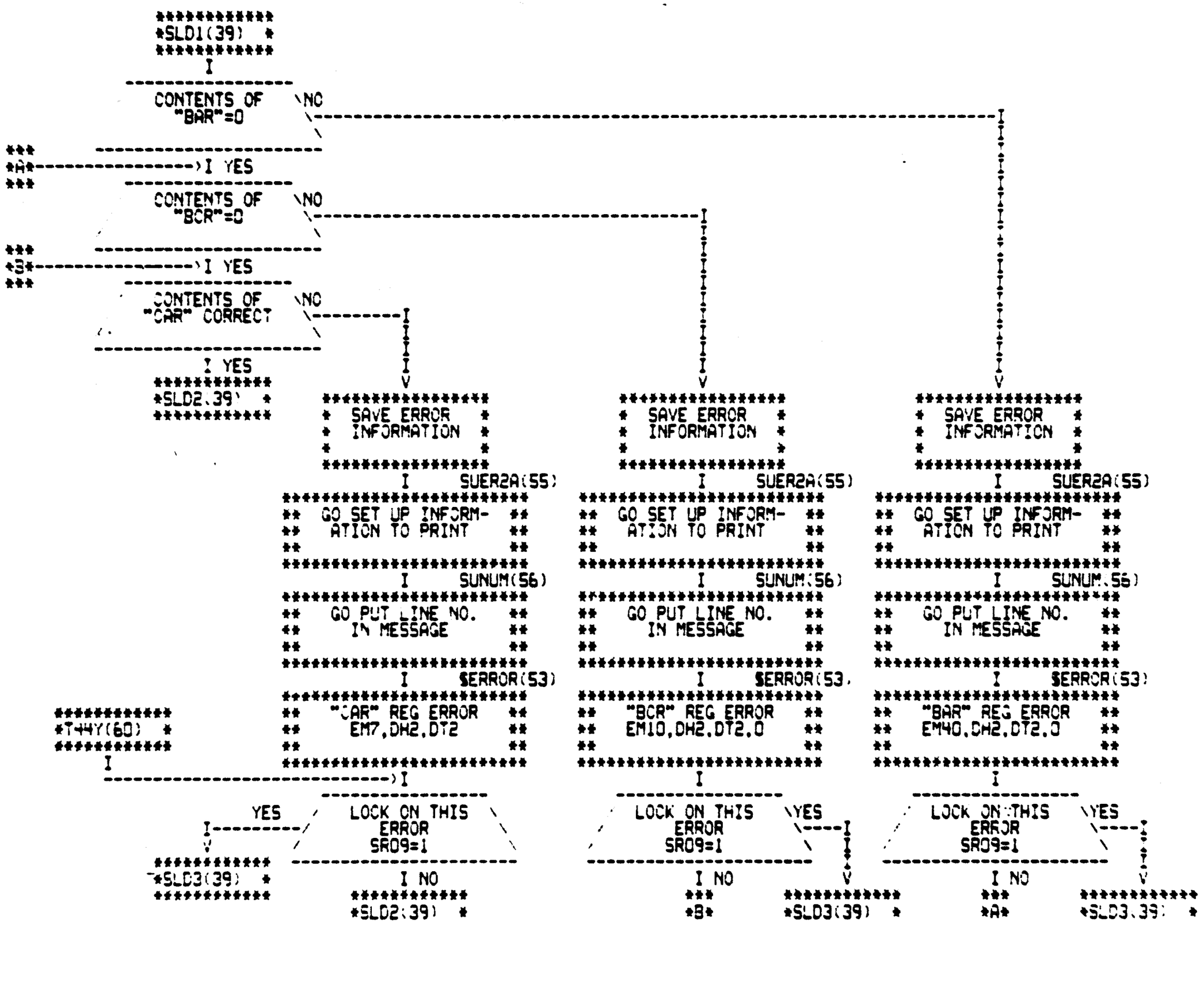

*TST40
SET UP ERROR LOOP RETURN TO GO TO 2$
SELINE(60)
GO SELECT A LINE NO. TO TEST
DONE ALL SELECTED LINES?
YES
C
NO
CLEAR DH11-PRIME SELECTED LINE TO XMIT 65. CHARS
INIT TIMERS "A" AND "B"
STORAGE OVER FLOW SET?
YES
NO
TIMEIT(59)
CALL TIMER
*T40A(60)
TIMEOUT?
YES
NO
B
A
READ TWO WORDS FROM SILO.
STORAGE OVER- FLOW CLEAR ?
NO
YES
SILO FILL COUNT=77(8) ?
YES
A
NO
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM- ATION TO PRINT
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
$ERROR(53)
REPORT STORAGE OVERFLOW ERROR EM57,DH2,DT2,0
*T40B(60)
LOOP ON THIS ERROR? SR09=1
YES
NO
B
A
SAVE ERROR INFORMATION
SUER2P(55)
GO SET UP INFORM- ATION TO PRINT
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
$ERROR(53)
REPORT STORAGE OVERFLOW ERROR EM57,DH2,DT2,0
*T40B(60)
C
$SCOPE(52)

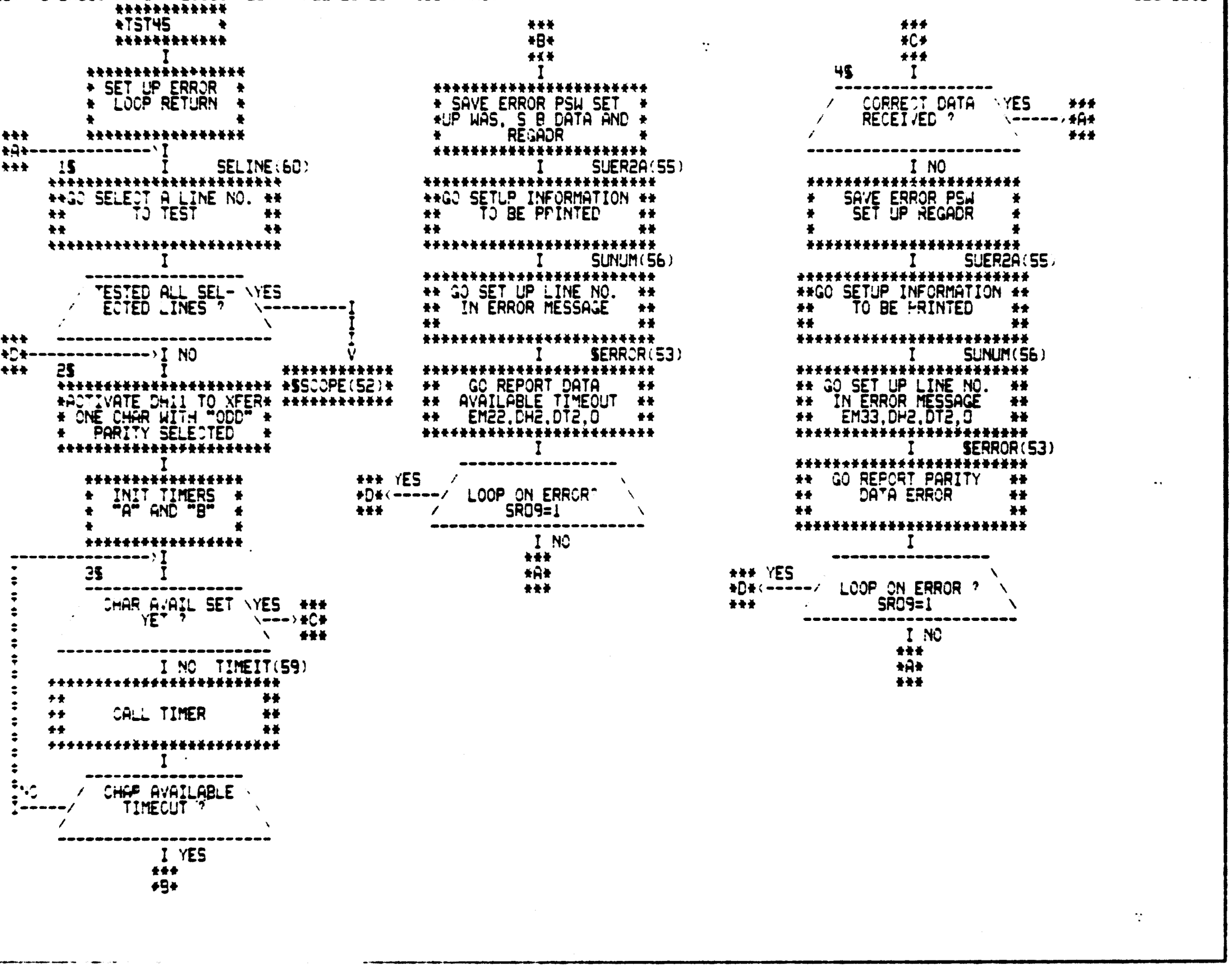

*TST41
SET UP ERROR LOOP RETURN
A
1$ SELINE(60)
GO SELECT A LINE NO. TO TEST
DONE TESTING ALL SELECTED LINES ?
YES
I NO
*INIT TO START AT LOW-EST SPEED
INIT "TIMEC" =-1
B
2$
*CLEAR THE DH11
ACTIVATE DH11 TO XMIT 3 CHARS ON SELECTED LINE
INIT TIMERS "A" AND "B"
4$
XMIT DONE SET ?
YES
C
TIMEIT(59)
CALL TIMER
DID TIMEOUT OCCUR BEFORE DONE SET ?
YES
I NO
SAVE ERROR PSW, SET UP SPEED CODE,WAS, AND S/B DATA
SUER2A(55)
GO SETUP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO. IN ERROR MESSAGE
$ERROR(53)
REPORT TIMEOUT ERROR EM50,DH2,DT2,0
LOOP ON ERROR SR09=1
YES
I NO
E
D
5$
WAS THIS SPEED FASTER THAN LAST ONE (TIMEB > TIMEC)
YES
I NO
7$
SAVE ERROR PSW SET UP SPEED CODE
SUER2A(55)
GO SETUP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO. IN ERROR MESSAGE
$ERROR(53)
REPORT TIMING ERROR EM17,DH6,DT2,0
LOOP ON ERROR ? SR09=1
NO
I YES
E
8$
CHANGE SPEED MAKE "TIMEC" EQUAL "TIMEB"
TESTED ALL SPEEDS ON SELECTED LINE ?
YES
***NO
B
A

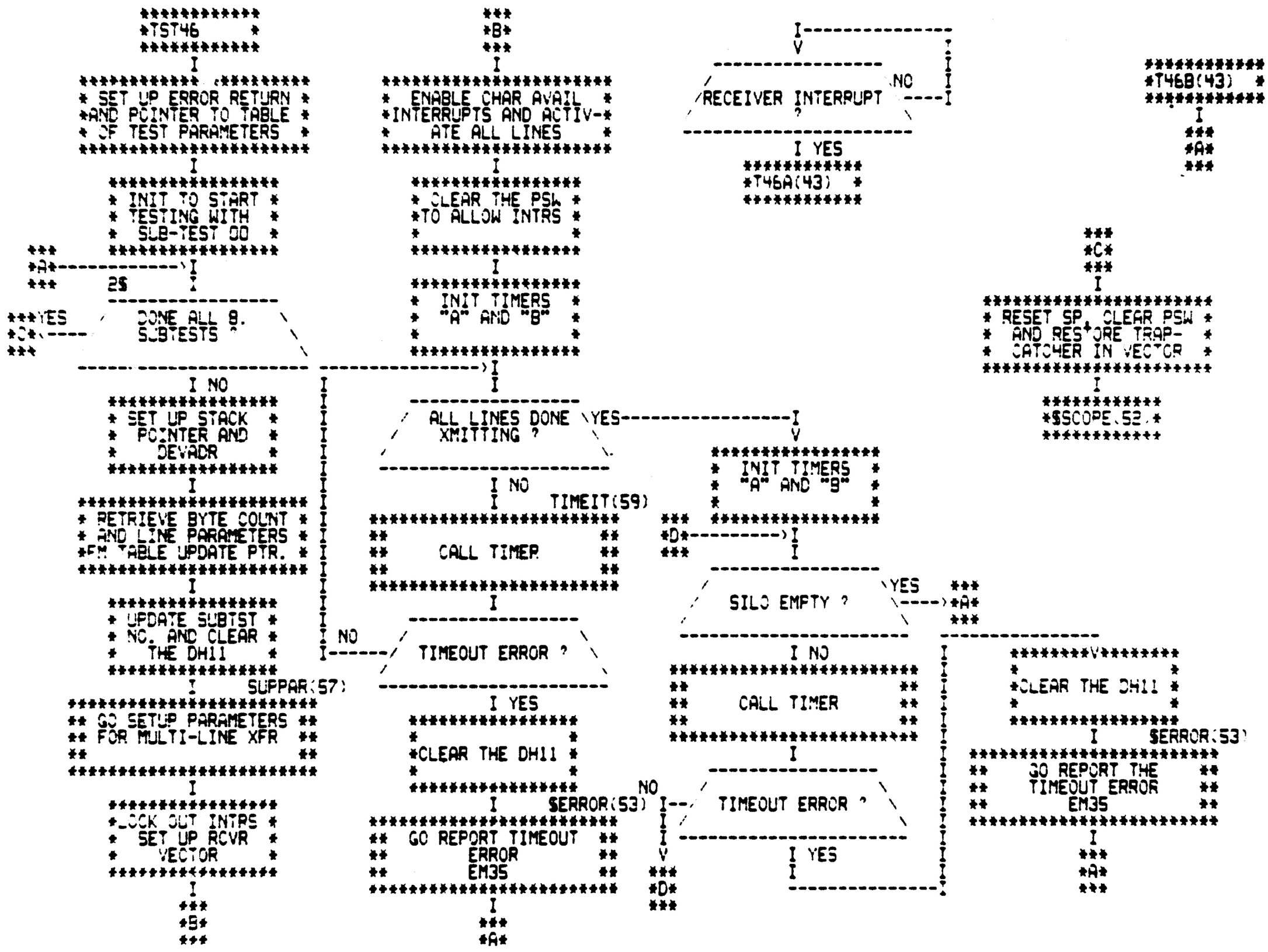

*TST42
SET UP ERROR LOOP RETURN
SELINE(60)
GO SELECT A LINE NO TO TEST
1$
DONE TESTING ALL SELECTED LINES ?
YES
NO
*INIT TO START AT LOW-EST SPEED INIT "TIMEC" = -1
$SCOPE(52)
2$
CLEAR THE DH11
ACTIVATE DH11 TO XMIT AND RECEIVE ON CHAR
INIT TIMERS "A" AND "B"
4$
CHAR AVAILABLE SET YET ?
D
C
TIMEIT(59)
CALL TIMER
DID TIMEOUT OCCUR BEFORE CHAR AVAIL SET
SAVE ERROR PSW,SET UP SPEED CODE, WAS AND S/B DATA
SUER2A(55)
GO SETUP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO. IN ERROR MESSAGE
$ERROR(53)
GO REPORT TIMEOUT WAITING FOR DONE EM22,DH2,DT2,0
LOOP ON ERROR ? SR09=1
B
E
5$
WAS THIS SPEED FASTER THAN LAST ONE (TIMEB>TIMEC)
8$
CHANGE SPEED MAKE "TIMEC" EQUAL "TIMEB"
7$
SAVE ERROR PSW
SET UP SPEED CODE
GO SETUP INFORMATION TO BE PRINTED
GO SET UP LINE NO. IN ERROR MESSAGE
GO REPORT RCVR TIMING ERROR EM20,DH6,DT2,0
LOOP ON ERROR ? SR09=1
TESTED ALL SPEEDS ON SELECTED LINES ?
A

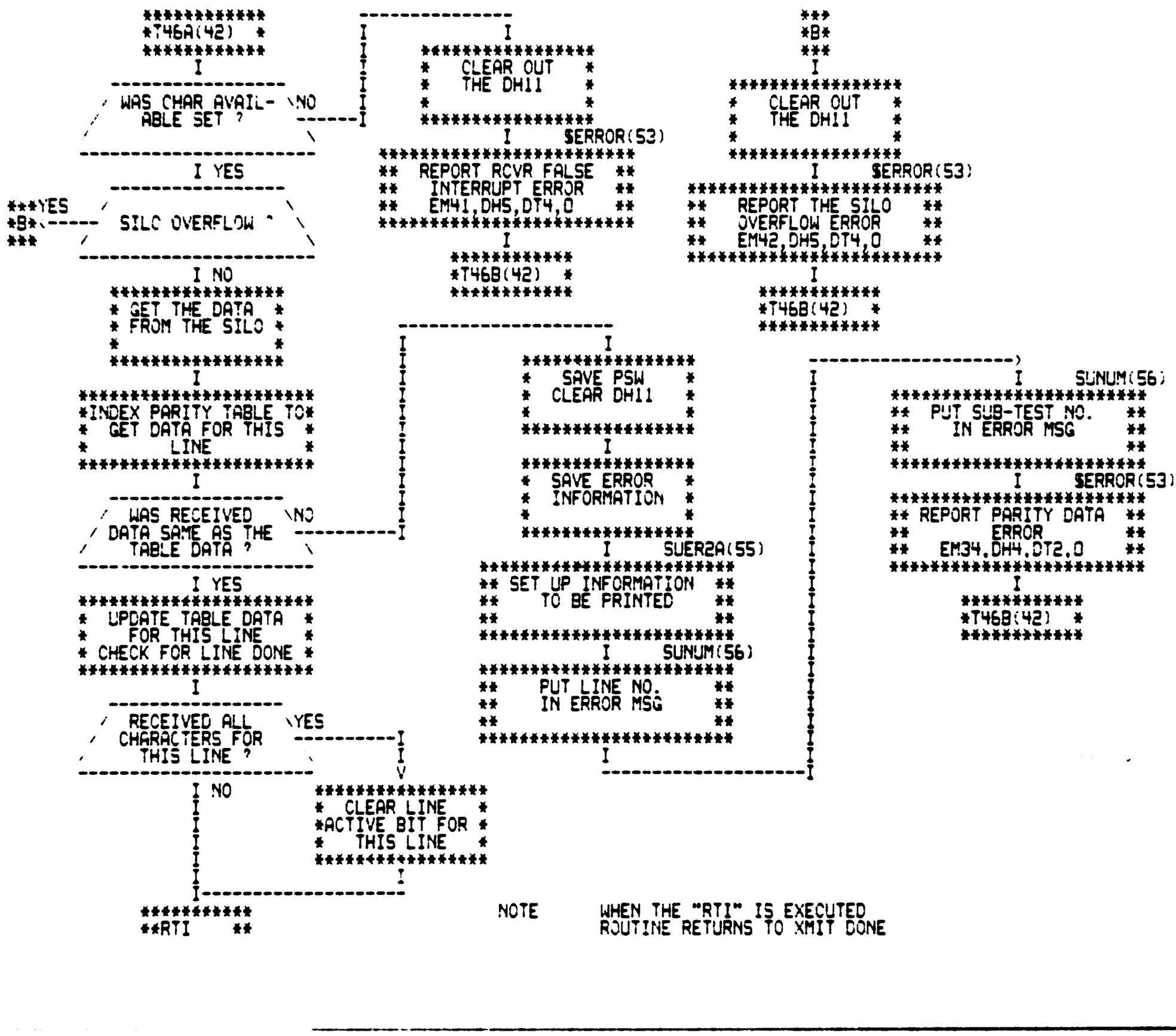

TST43
SET UP ERROR LOOP RETURN
A
1$
SELINE(60)
GO SELECT A LINE TO TEST
DONE TESTING ALL SELECTED LINES ?
YES
NO
$SCOPE(52)
SET UP DATA TABLE POINTER - INIT R2 TO START WITH 5 BITS
GET TEST CHAR FROM TABLE AND UPDATE TABLE POINTER
D
ACTIVATE DH11 TO OUT-PUT TEST CHAR AT
INIT TIMERS "A" AND "B"
CHAR AVAILABLE SET YET ?
YES
NO
B
TIMEIT(59)
CALL THE TIMER
CHAR AVAIL. TIMEOUT ?
YES
NO
SAVE ERROR PSW
SET UP S/B DATA
GET WAS DATA
SUER2A(55)
SETUP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO. IN ERROR MESSAGE
$ERROR(53)
GO REPORT DATA
AVAIL TIMEOUT
EM22,DH2,DT2,0
LOOP ON ERROR ? SR09=1
YES
NO
E
C
WAS RECEIVED DATA CORRECT
YES
NO
SUER2(55)
GO SETUP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO. IN ERROR MESSAGE
$ERROR(53)
GO REPORT DATA
COMPARE ERROR
EM23,DH7,DT2,0
LOOP ON ERROR SR09=1
NO
YES
E
INCREMENT R2 TO SELECT NEXT CHAR LENGTH
TESTED ALL 4 CHAR LENGTHS ON THIS LINE
NO
YES
A


*TST44
SET UP ERROR LOOP RETURN TO GO TO 2$
SLD2(40)
SELINE(60)
GO SELECT A LINE NO. TO TEST
DONE ALL SELECTED LINES ?
SLD3(40)
NO
YES
CLEAR DH11-PRIME
SELECTED LINE TO XMIT
BINARY PATTERN
INIT TIMERS "A" AND "B"
CHAR AVAILABLE SET YET ?
NO TIMEIT(59)
CALL TIMER
TIMEOUT ?
T44X(60)
STORE SILO WORD IN "RBUF" AND UPDATE POINTER
RECEIVED 256. CHARS YET ?
SET UP POINTERS TO CHECK (TBUF)=(RBUF)
DID XMIT CHAR COMPARE WITH RCVD CHAR
UPDATE POINTERS TO NEXT CHAR
COMPARED 256. CHARS?
SLD1(40)
$SCOPE(52)
SUER2(55)
GO SET UP INFORM- ATION TO PRINT
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
$ERROR(53)
REPORT DATA COMPARE ERROR
EM37,DH4,DT2,0
A
B

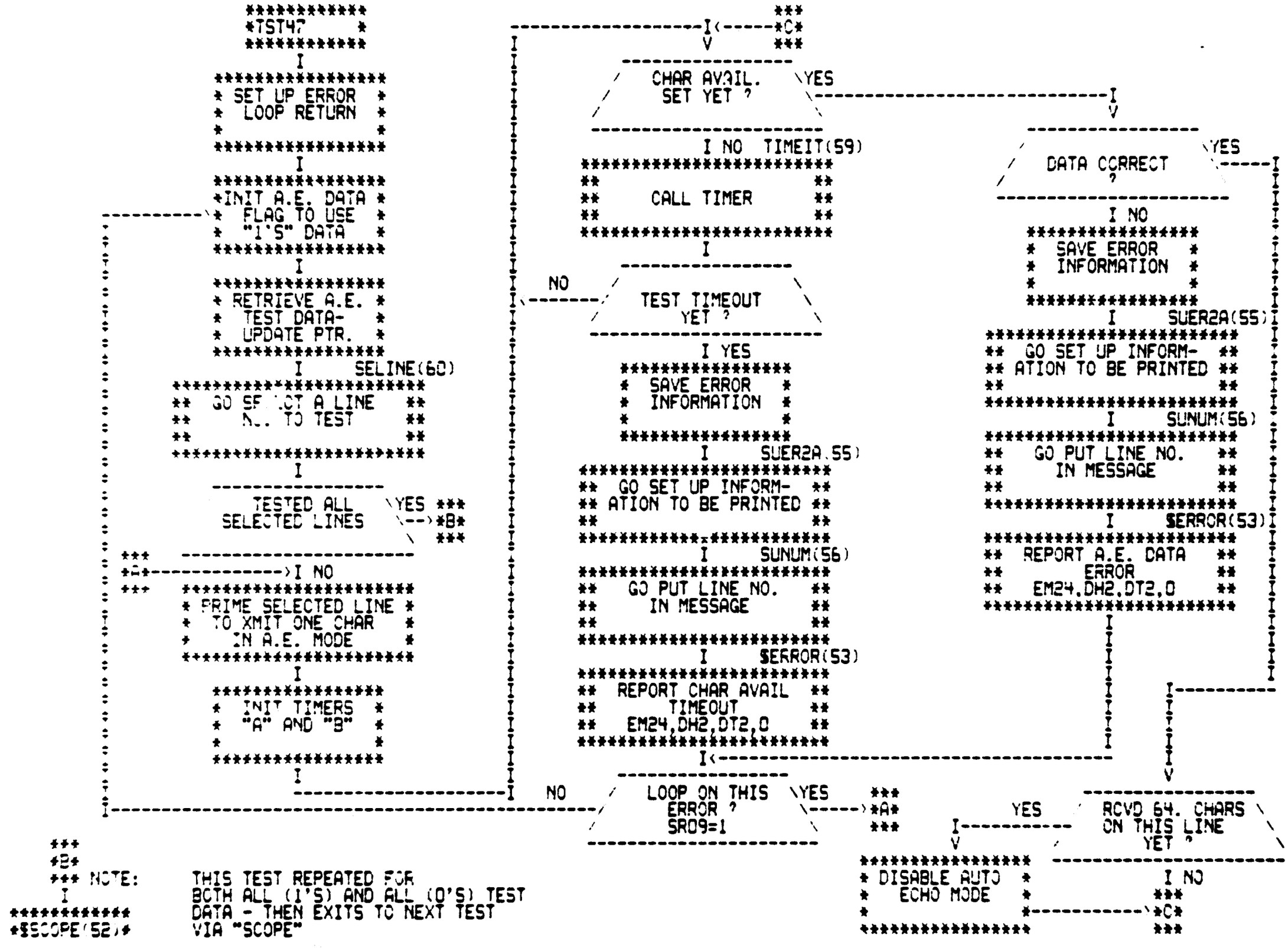

*SLD1(39) *
CONTENTS OF "BAR"=0
NO
YES
CONTENTS OF "BCR"=0
NO
YES
CONTENTS OF "CAR" CORRECT
NO
YES
*SLD2(39) *
*T44Y(60) *
*SLD3(39) *
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM- ATION TO PRINT
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
SERROR(53)
"CAR" REG ERROR EM7,DH2,DT2
"BCR" REG ERROR EM10,DH2,DT2,0
"BAR" REG ERROR EM40,DH2,DT2,0
LOCK ON THIS ERROR SR09=1
*SLD2(39) *
*SLD3(39) *
A
B

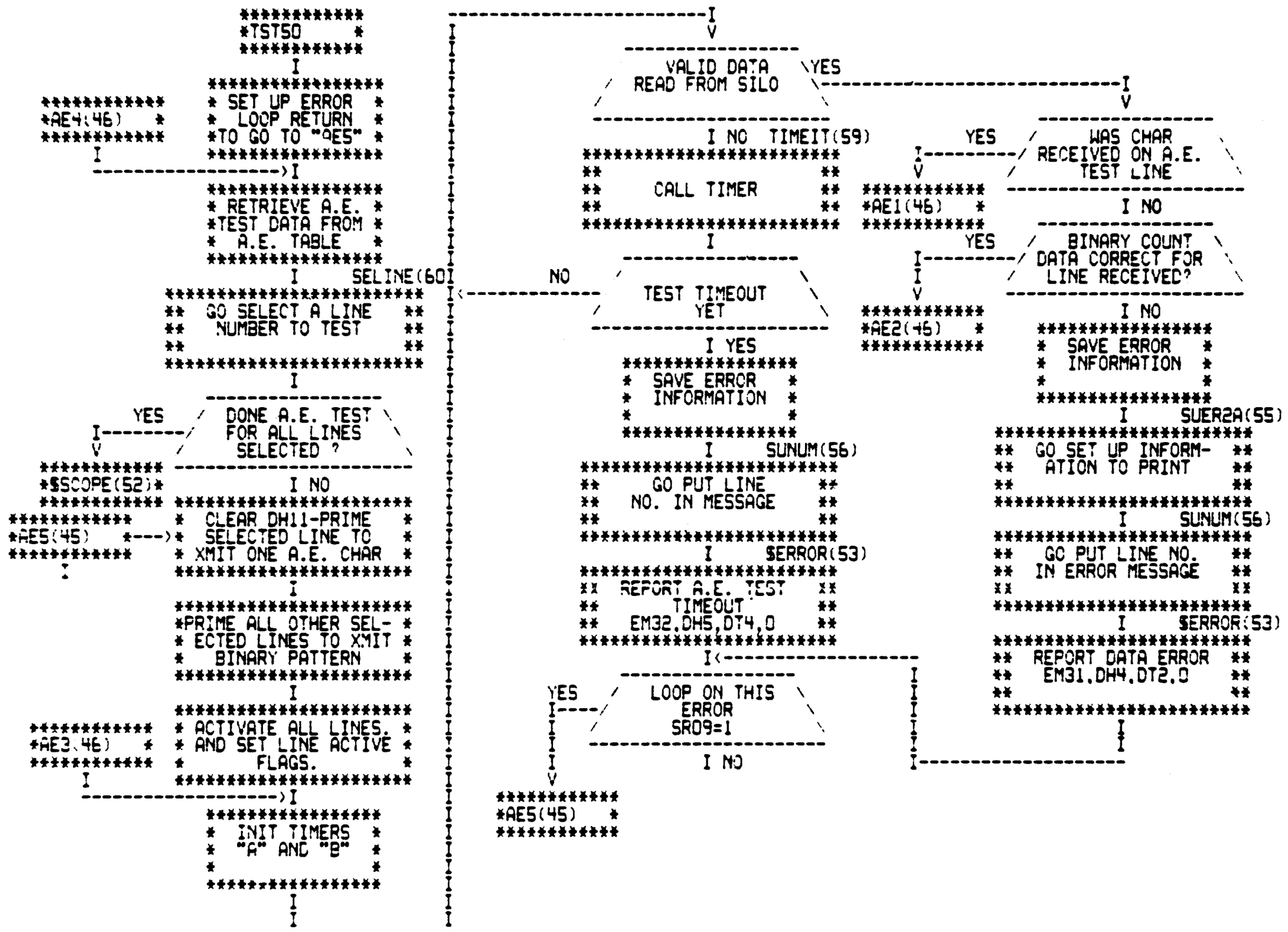

*TST45
SET UP ERROR LOOP RETURN
A
1$
SELINE(60)
GO SELECT A LINE NO. TO TEST
TESTED ALL SEL- ECTED LINES ?
YES
NO
D
2$
SCOPE(52)
ACTIVATE DH11 TO XFER ONE CHAR WITH "ODD" PARITY SELECTED
INIT TIMERS "A" AND "B"
3$
CHAR AVAIL SET YES ?
YES
C
NO
TIMEIT(59)
CALL TIMER
CHAR AVAILABLE TIMEOUT ?
YES
9
B
SAVE ERROR PSW SET UP WAS, S B DATA AND REGADR
SUER2A(55)
GO SETUP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO. IN ERROR MESSAGE
$ERROR(53)
GO REPORT DATA AVAILABLE TIMEOUT EM22,DH2,DT2,0
LOOP ON ERROR ? SR09=1
YES
NO
A
4$
CORRECT DATA RECEIVED ?
YES
NO
SAVE ERROR PSW SET UP REGADR
SUER2A(55)
GO SETUP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE NO. IN ERROR MESSAGE EM33,DH2,DT2,0
$ERROR(53)
GO REPORT PARITY DATA ERROR
LOOP ON ERROR ? SR09=1
YES
NO
A

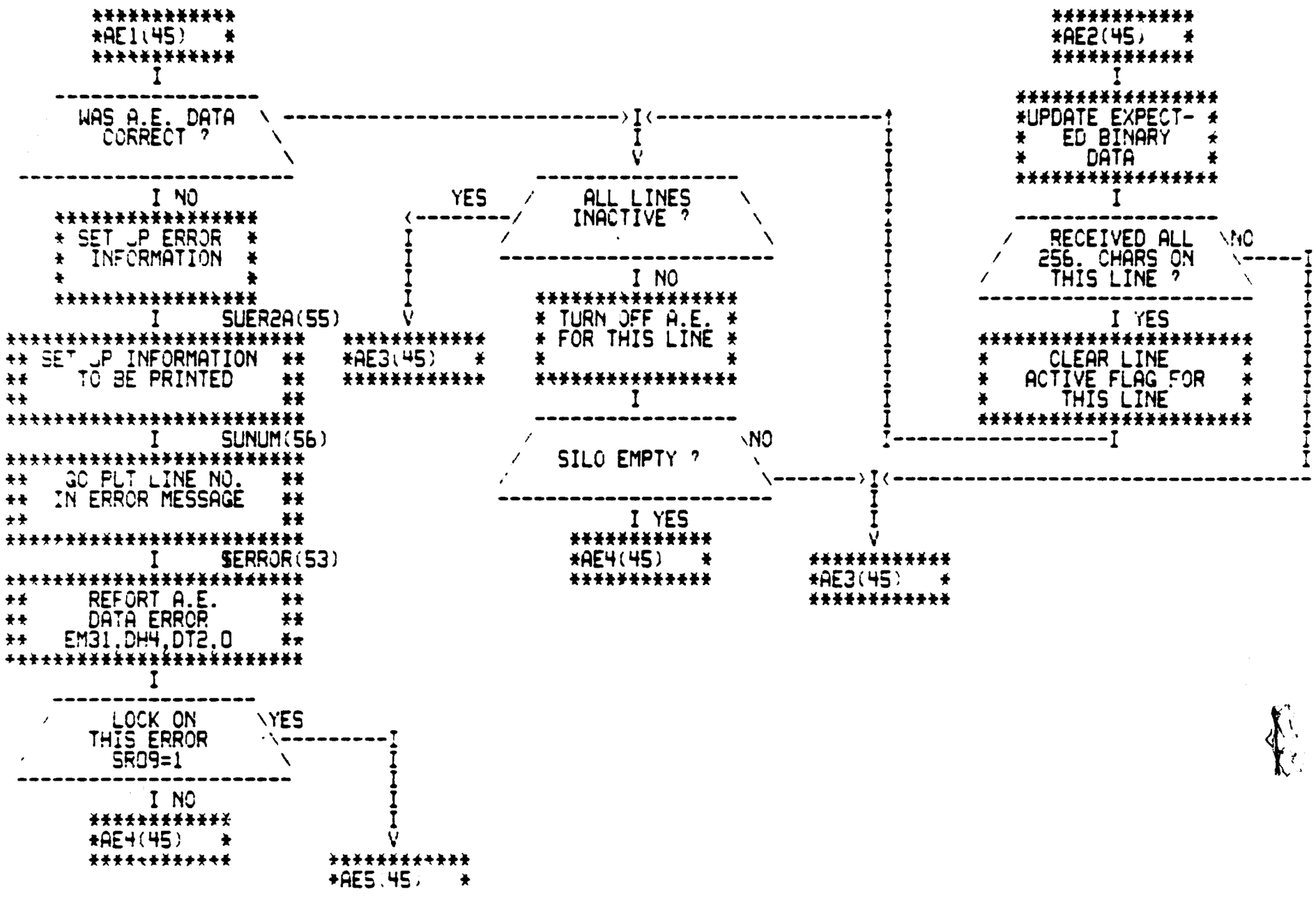

TST46
SET UP ERROR RETURN AND POINTER TO TABLE OF TEST PARAMETERS
INIT TO START TESTING WITH SUB-TEST 00
B
ENABLE CHAR AVAIL INTERRUPTS AND ACTIVATE ALL LINES
CLEAR THE PSW TO ALLOW INTRS
INIT TIMERS "A" AND "B"
A
25
DONE ALL 8. SUBTESTS ?
YES
C
NO
SET UP STACK POINTER AND DEVADR
RETRIEVE BYTE COUNT AND LINE PARAMETERS FM TABLE UPDATE PTR.
UPDATE SUBTST NO. AND CLEAR THE DH11
SUPPAR(57)
GO SETUP PARAMETERS FOR MULTI-LINE XFR
LOCK OUT INTRS SET UP RCVR VECTOR
B
ALL LINES DONE XMITTING ?
YES
NO
TIMEIT(59)
CALL TIMER
TIMEOUT ERROR ?
YES
CLEAR THE DH11
$ERROR(53)
GO REPORT TIMEOUT ERROR EM35
A
RECEIVER INTERRUPT ?
NO
YES
T46A(43)
INIT TIMERS "A" AND "B"
D
SILO EMPTY ?
YES
NO
CALL TIMER
TIMEOUT ERROR ?
YES
NO
A
T46B(43)
A
C
RESET SP, CLEAR PSW AND RESTORE TRAP-CATCHER IN VECTOR
$SCOPE.52
CLEAR THE DH11
$ERROR(53)
GO REPORT THE TIMEOUT ERROR EM35
A

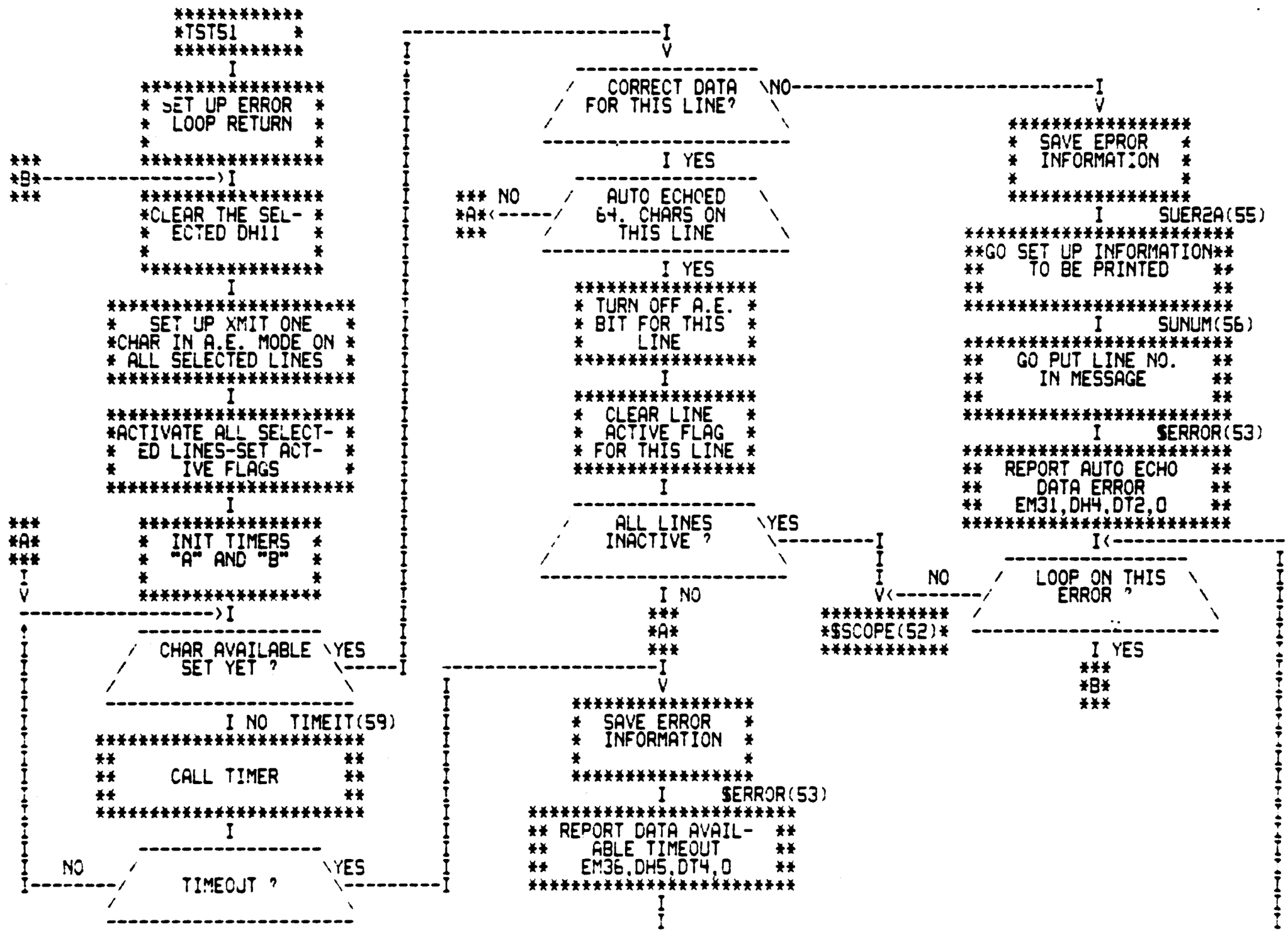


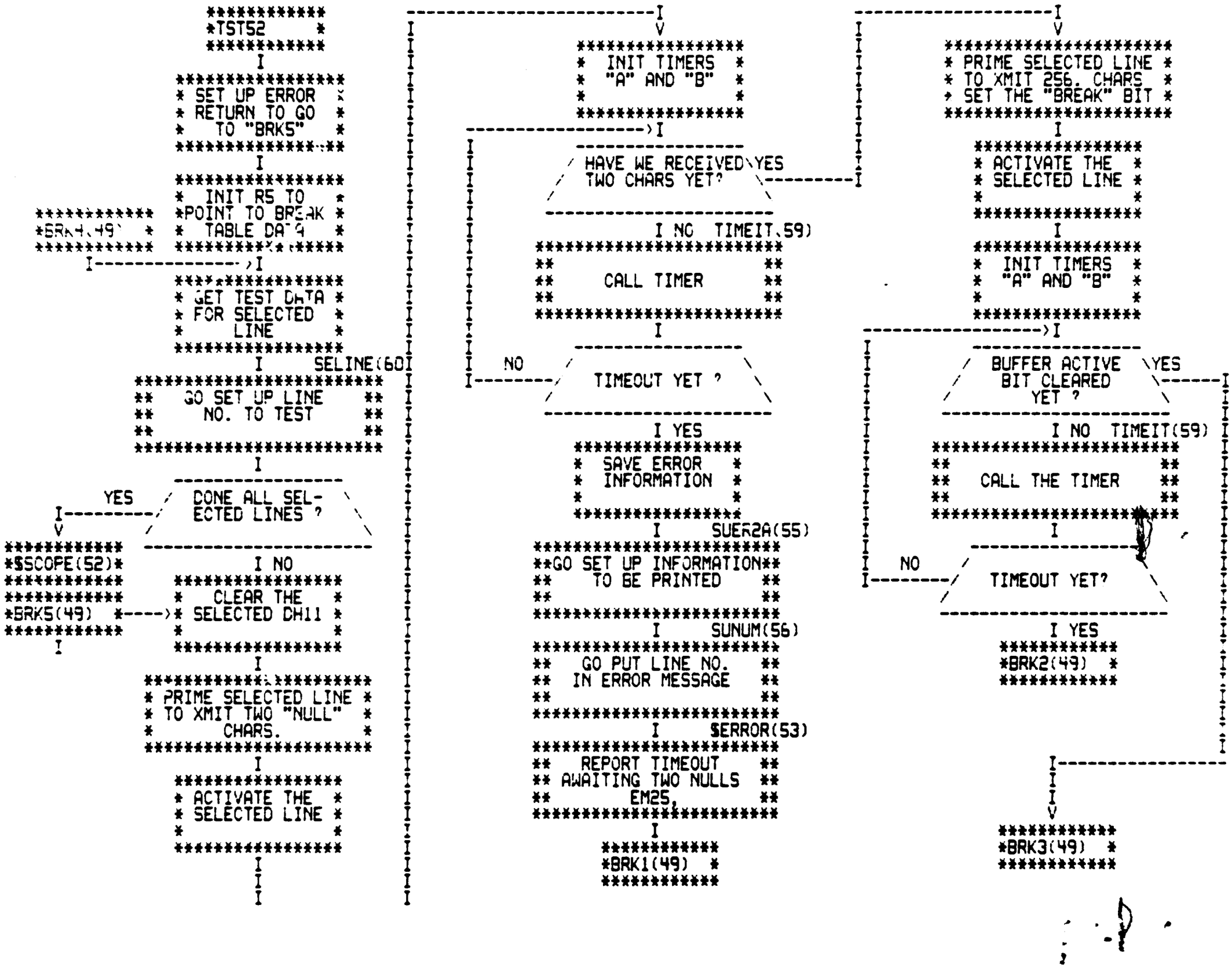
NOTE WHEN THE "RTI" IS EXECUTED ROUTINE RETURNS TO XMIT DONE

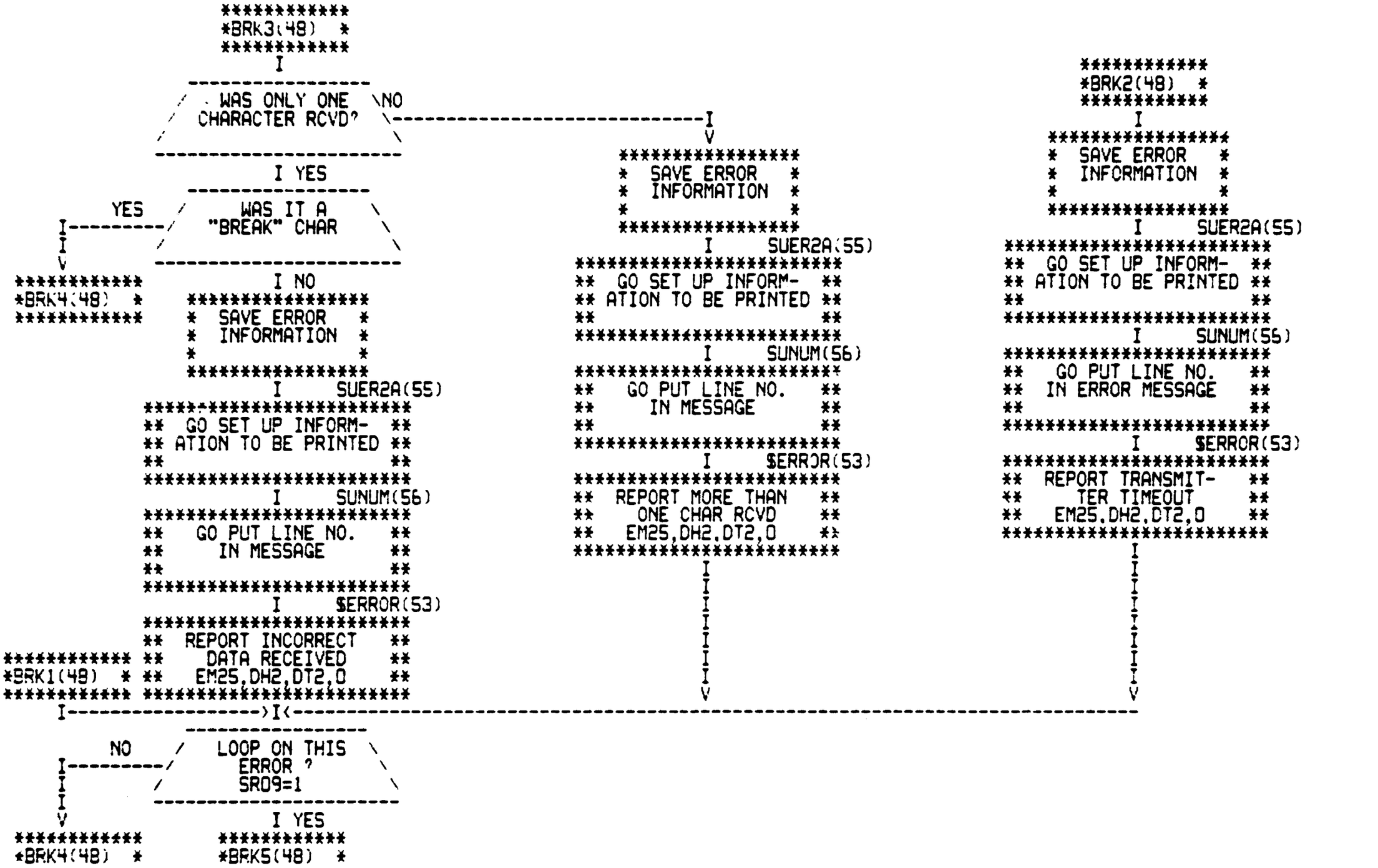

TST47
SET UP ERROR LOOP RETURN
INIT A.E. DATA FLAG TO USE "1'S" DATA
RETRIEVE A.E. TEST DATA- UPDATE PTR.
SELINE(60)
GO SELECT A LINE NO. TO TEST
TESTED ALL SELECTED LINES
YES
B
NO
PRIME SELECTED LINE TO XMIT ONE CHAR IN A.E. MODE
INIT TIMERS "A" AND "B"
C
CHAR AVAIL. SET YET ?
YES
NO
TIMEIT(59)
CALL TIMER
TEST TIMEOUT YET ?
YES
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM- ATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
SERROR(53)
REPORT CHAR AVAIL TIMEOUT
EM24,DH2,DT2,0
LOOP ON THIS ERROR ? SR09=1
YES
A
DATA CORRECT ?
NO
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM- ATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
SERROR(53)
REPORT A.E. DATA ERROR
EM24,DH2,DT2,0
RCVD 64. CHARS ON THIS LINE YET ?
YES
NO
DISABLE AUTO ECHO MODE
C
B
*** NOTE: THIS TEST REPEATED FOR BOTH ALL (1'S) AND ALL (0'S) TEST DATA - THEN EXITS TO NEXT TEST VIA "SCOPE"
$SCOPE(52)

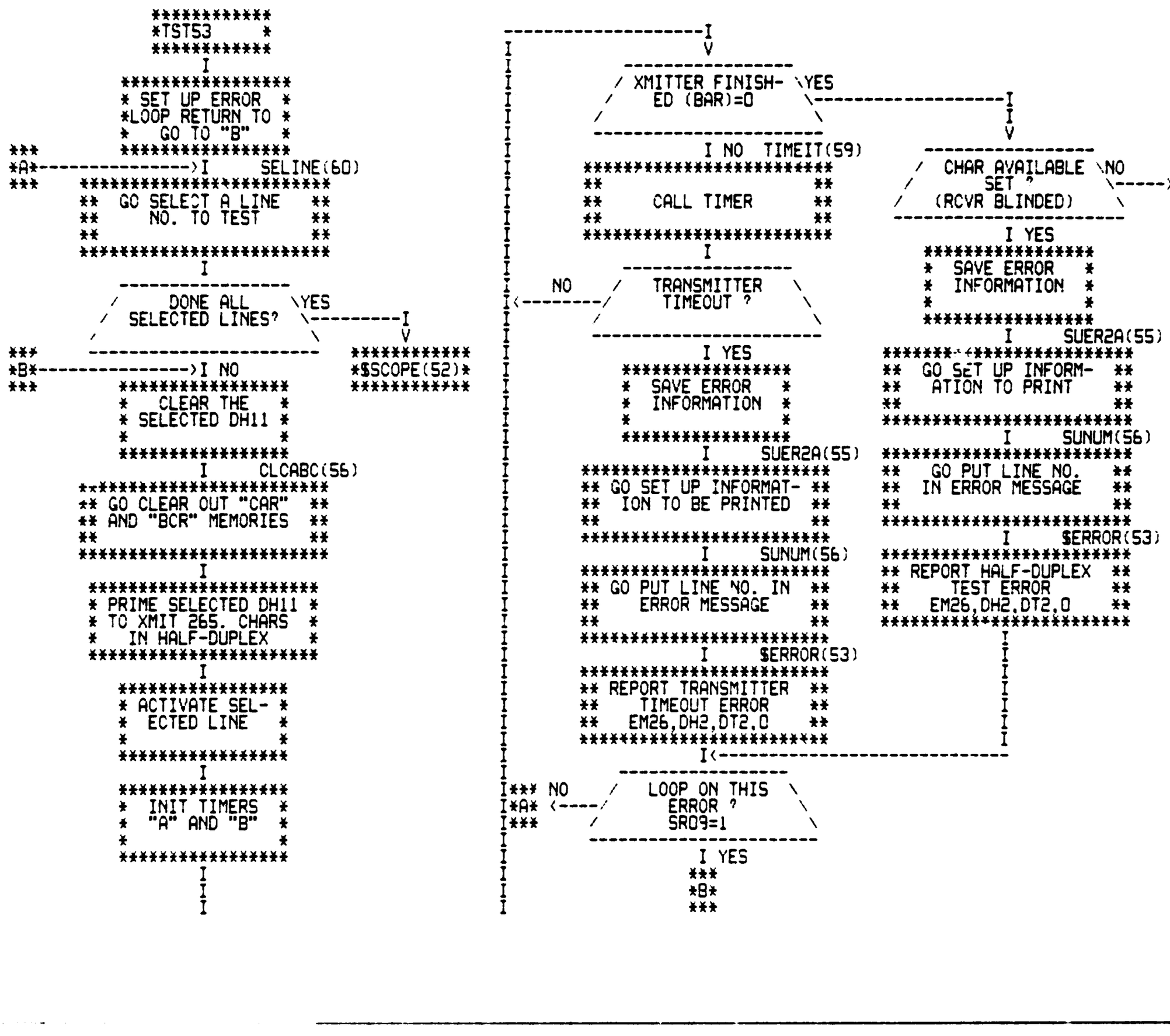

*TST50
SET UP ERROR LOOP RETURN TO GO TO "AE5"
*AE4(46)
RETRIEVE A.E. TEST DATA FROM A.E. TABLE
SELINE(60)
GO SELECT A LINE NUMBER TO TEST
DONE A.E. TEST FOR ALL LINES SELECTED ?
YES
NO
*$SCOPE(52)
CLEAR DH11-PRIME SELECTED LINE TO XMIT ONE A.E. CHAR
*AE5(45)
PRIME ALL OTHER SELECTED LINES TO XMIT BINARY PATTERN
ACTIVATE ALL LINES. AND SET LINE ACTIVE FLAGS.
*AE3(46)
INIT TIMERS "A" AND "B"
VALID DATA READ FROM SILO
YES
I NO TIMEIT(59)
CALL TIMER
TEST TIMEOUT YET
NO
YES
SAVE ERROR INFORMATION
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
$ERROR(53)
REPORT A.E. TEST TIMEOUT EM32,DH5,DT4,0
LOOP ON THIS ERROR SR09=1
YES
NO
*AE5(45)
WAS CHAR RECEIVED ON A.E. TEST LINE
YES
*AE1(46)
BINARY COUNT DATA CORRECT FOR LINE RECEIVED?
YES
*AE2(46)
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM-ATION TO PRINT
SUNUM(56)
GO PUT LINE NO. IN ERROR MESSAGE
$ERROR(53)
REPORT DATA ERROR EM31,DH4,DT2,0

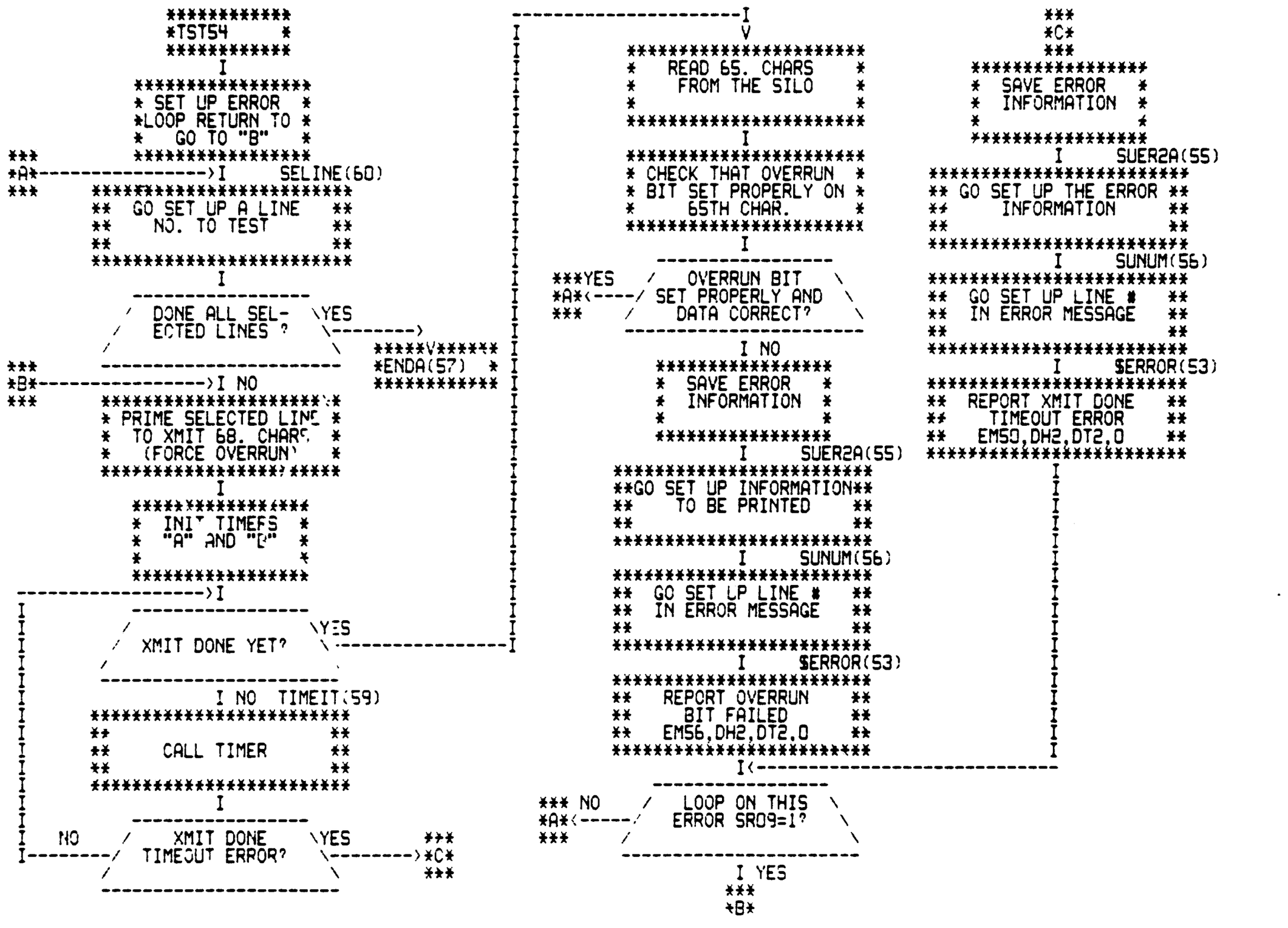

*AE1(45) *
WAS A.E. DATA CORRECT ?
I NO
* SET UP ERROR INFORMATION *
SUER2A(55)
** SET UP INFORMATION TO BE PRINTED **
SUNUM(56)
** GC PLT LINE NO. IN ERROR MESSAGE **
$ERROR(53)
** REPORT A.E. DATA ERROR EM31,DH4,DT2,0 **
LOCK ON THIS ERROR SR09=1
I NO
*AE4(45) *
YES
*AE3(45) *
ALL LINES INACTIVE ?
I NO
* TURN OFF A.E. * FOR THIS LINE *
SILO EMPTY ?
I YES
*AE4(45) *
*AE5,45, *
*AE2(45) *
*UPDATE EXPECTED BINARY DATA *
RECEIVED ALL 256. CHARS ON THIS LINE ?
NO
I YES
* CLEAR LINE ACTIVE FLAG FOR THIS LINE *
*AE3(45) *

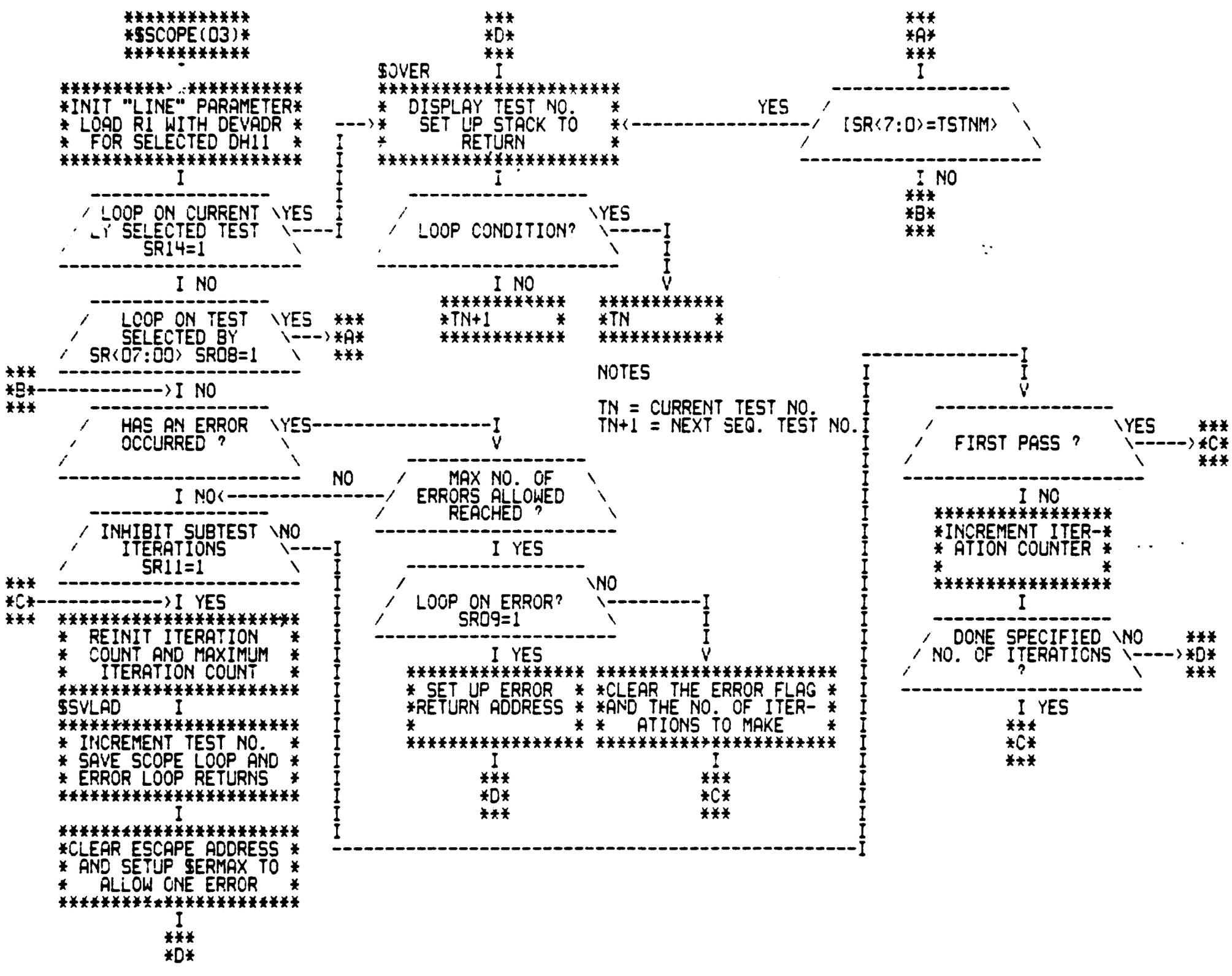

*TST51
SET UP ERROR LOOP RETURN
B
CLEAR THE SEL- ECTED DH11
SET UP XMIT ONE CHAR IN A.E. MODE ON ALL SELECTED LINES
ACTIVATE ALL SELECT- ED LINES-SET ACT- IVE FLAGS
A
INIT TIMERS "A" AND "B"
CHAR AVAILABLE SET YET ?
YES
NO
CALL TIMER
TIMEOUT ?
YES
NO
TIMEIT(59)
CORRECT DATA FOR THIS LINE?
YES
NO
AUTO ECHOED 64. CHARS ON THIS LINE
YES
NO
A
TURN OFF A.E. BIT FOR THIS LINE
CLEAR LINE ACTIVE FLAG FOR THIS LINE
ALL LINES INACTIVE ?
YES
NO
A
SAVE ERROR INFORMATION
$ERROR(53)
REPORT DATA AVAIL- ABLE TIMEOUT EM36,DH5,DT4,0
SAVE EPROR INFORMATION
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
$ERROR(53)
REPORT AUTO ECHO DATA ERROR EM31,DH4,DT2,0
LOOP ON THIS ERROR ?
NO
YES
B
$SCOPE(52)


*TST52
SET UP ERROR RETURN TO GO TO "BRK5"
INIT R5 TO POINT TO BREAK TABLE DATA
*BRK4(49)
GET TEST DATA FOR SELECTED LINE
SELINE(60)
GO SET UP LINE NO. TO TEST
DONE ALL SELECTED LINES ?
YES
NO
*$SCOPE(52)
*BRK5(49)
CLEAR THE SELECTED DH11
PRIME SELECTED LINE TO XMIT TWO "NULL" CHARS.
ACTIVATE THE SELECTED LINE
INIT TIMERS "A" AND "B"
HAVE WE RECEIVED TWO CHARS YET?
YES
NO TIMEIT(59)
CALL TIMER
NO
TIMEOUT YET ?
YES
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN ERROR MESSAGE
$ERROR(53)
REPORT TIMEOUT AWAITING TWO NULLS EM25,
*BRK1(49)
PRIME SELECTED LINE TO XMIT 256. CHARS SET THE "BREAK" BIT
ACTIVATE THE SELECTED LINE
INIT TIMERS "A" AND "B"
BUFFER ACTIVE BIT CLEARED YET ?
YES
NO TIMEIT(59)
CALL THE TIMER
NO
TIMEOUT YET?
YES
*BRK2(49)
*BRK3(49)

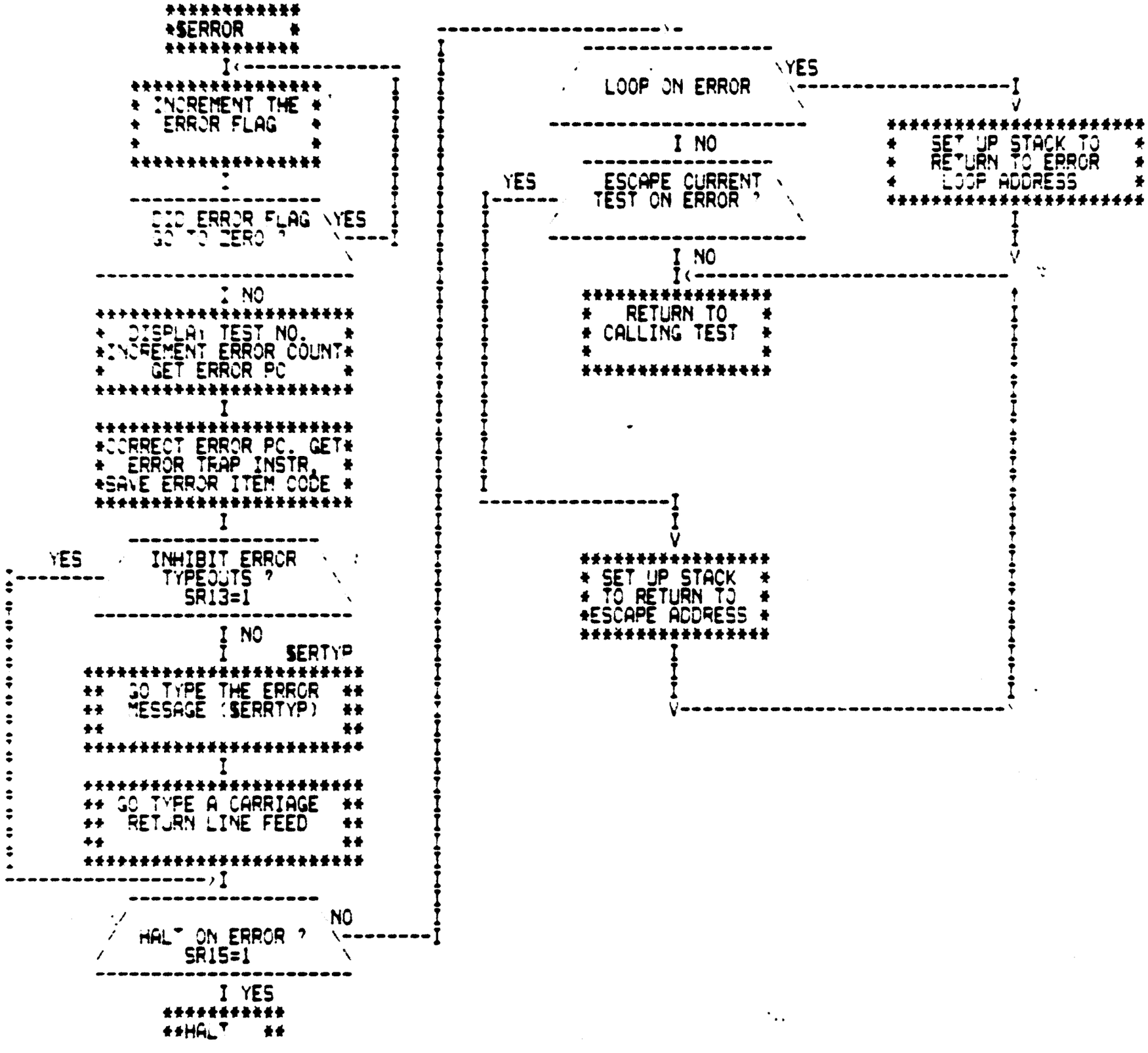

*BRK3(48) *
WAS ONLY ONE CHARACTER RCVD?
NO
I YES
WAS IT A "BREAK" CHAR
YES
*BRK4(48) *
I NO
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM- ATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
$ERROR(53)
REPORT INCORRECT DATA RECEIVED
EM25,DH2,DT2,0
*BRK1(48) *
LOOP ON THIS ERROR ? SR09=1
NO
I YES
*BRK4(48) *
*BRK5(48) *
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM- ATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
$ERROR(53)
REPORT MORE THAN ONE CHAR RCVD
EM25,DH2,DT2,0
*BRK2(48) *
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM- ATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
$ERROR(53)
REPORT TRANSMIT- TER TIMEOUT
EM25,DH2,DT2,0

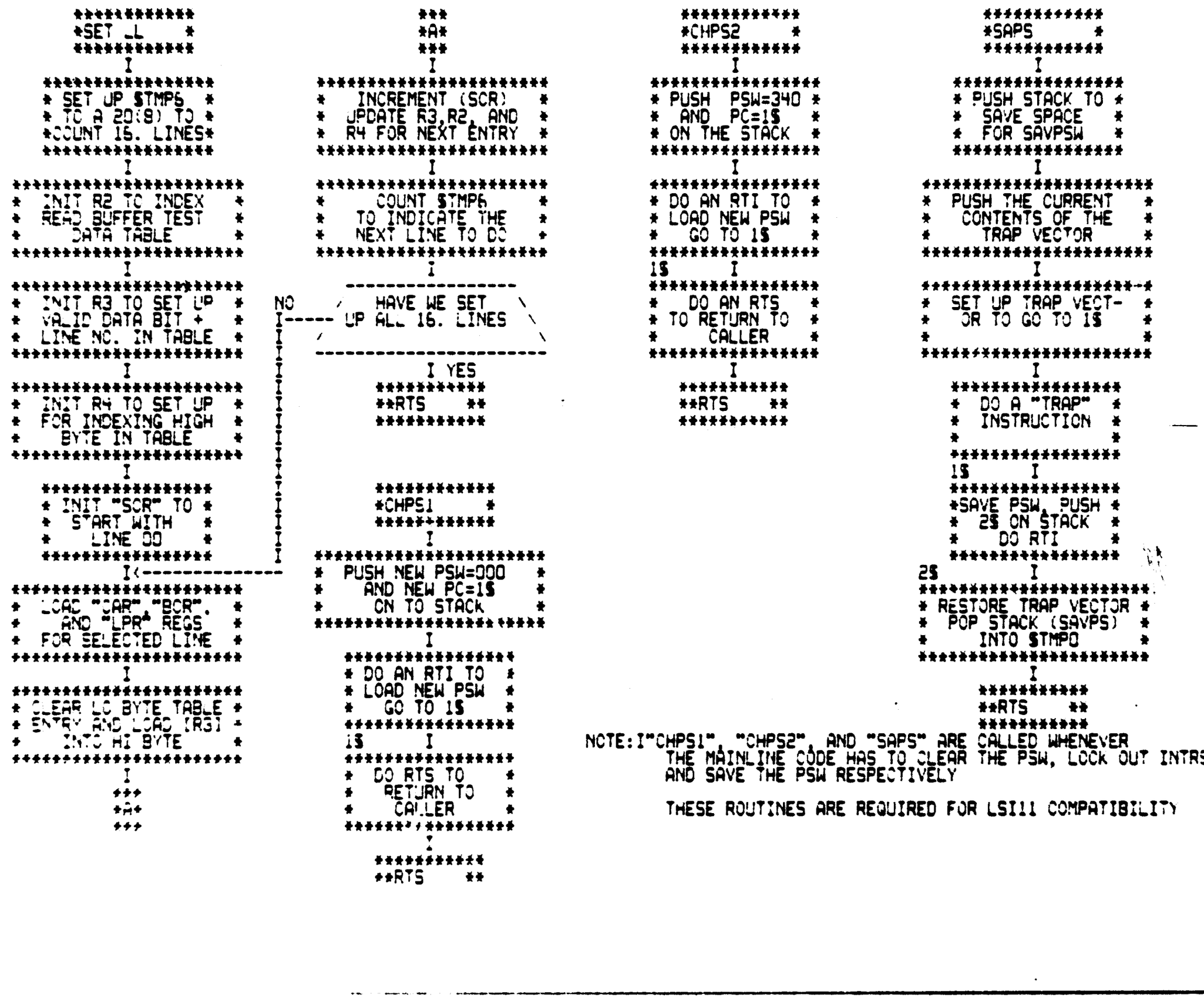

*TST53
SET UP ERROR LOOP RETURN TO GO TO "B"
SELINE(60)
GO SELECT A LINE NO. TO TEST
DONE ALL SELECTED LINES?
$SCOPE(52)
CLEAR THE SELECTED DH11
CLCABC(56)
GO CLEAR OUT "CAR" AND "BCR" MEMORIES
PRIME SELECTED DH11 TO XMIT 265. CHARS IN HALF-DUPLEX
ACTIVATE SEL- ECTED LINE
INIT TIMERS "A" AND "B"
XMITTER FINISH- ED (BAR)=0
I NO TIMEIT(59)
CALL TIMER
TRANSMITTER TIMEOUT ?
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORMAT- ION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN ERROR MESSAGE
$ERROR(53)
REPORT TRANSMITTER TIMEOUT ERROR EM26,DH2,DT2,0
LOOP ON THIS ERROR ? SR09=1
CHAR AVAILABLE SET ? (RCVR BLINDED)
GO SET UP INFORM- ATION TO PRINT
REPORT HALF-DUPLEX TEST ERROR EM26,DH2,DT2,0
YES
NO


TST54
SET UP ERROR LOOP RETURN TO "B"
SELINE(60)
GO SET UP A LINE NO. TO TEST
DONE ALL SELECTED LINES ?
YES
ENDA(57)
NO
PRIME SELECTED LINE TO XMIT 68. CHARS (FORCE OVERRUN)
INIT TIMERS "A" AND "B"
XMIT DONE YET?
YES
NO TIMEIT(59)
CALL TIMER
XMIT DONE TIMEOUT ERROR?
NO
YES
C
READ 65. CHARS FROM THE SILO
CHECK THAT OVERRUN BIT SET PROPERLY ON 65TH CHAR.
OVERRUN BIT SET PROPERLY AND DATA CORRECT?
YES
NO
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE # IN ERROR MESSAGE
$ERROR(53)
REPORT OVERRUN BIT FAILED
EM56,DH2,DT2,0
LOOP ON THIS ERROR SR09=1?
NO
YES
B
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP THE ERROR INFORMATION
SUNUM(56)
GO SET UP LINE # IN ERROR MESSAGE
$ERROR(53)
REPORT XMIT DONE TIMEOUT ERROR
EM50,DH2,DT2,0


$SCOPE(03)
$OVER
INIT "LINE" PARAMETER
LOAD R1 WITH DEVADR
FOR SELECTED DH11
LOOP ON CURRENT SELECTED TEST
SR14=1
LOOP ON TEST SELECTED BY SR<07:00> SR08=1
HAS AN ERROR OCCURRED ?
INHIBIT SUBTEST ITERATIONS
SR11=1
REINIT ITERATION COUNT AND MAXIMUM ITERATION COUNT
SVLAD
INCREMENT TEST NO.
SAVE SCOPE LOOP AND ERROR LOOP RETURNS
CLEAR ESCAPE ADDRESS AND SETUP $ERMAX TO ALLOW ONE ERROR
DISPLAY TEST NO.
SET UP STACK TO RETURN
LOOP CONDITION?
TN+1
TN
NOTES
TN = CURRENT TEST NO.
TN+1 = NEXT SEQ. TEST NO.
MAX NO. OF ERRORS ALLOWED REACHED ?
LOOP ON ERROR? SR09=1
SET UP ERROR RETURN ADDRESS
CLEAR THE ERROR FLAG AND THE NO. OF ITERATIONS TO MAKE
A
B
C
D
YES
NO

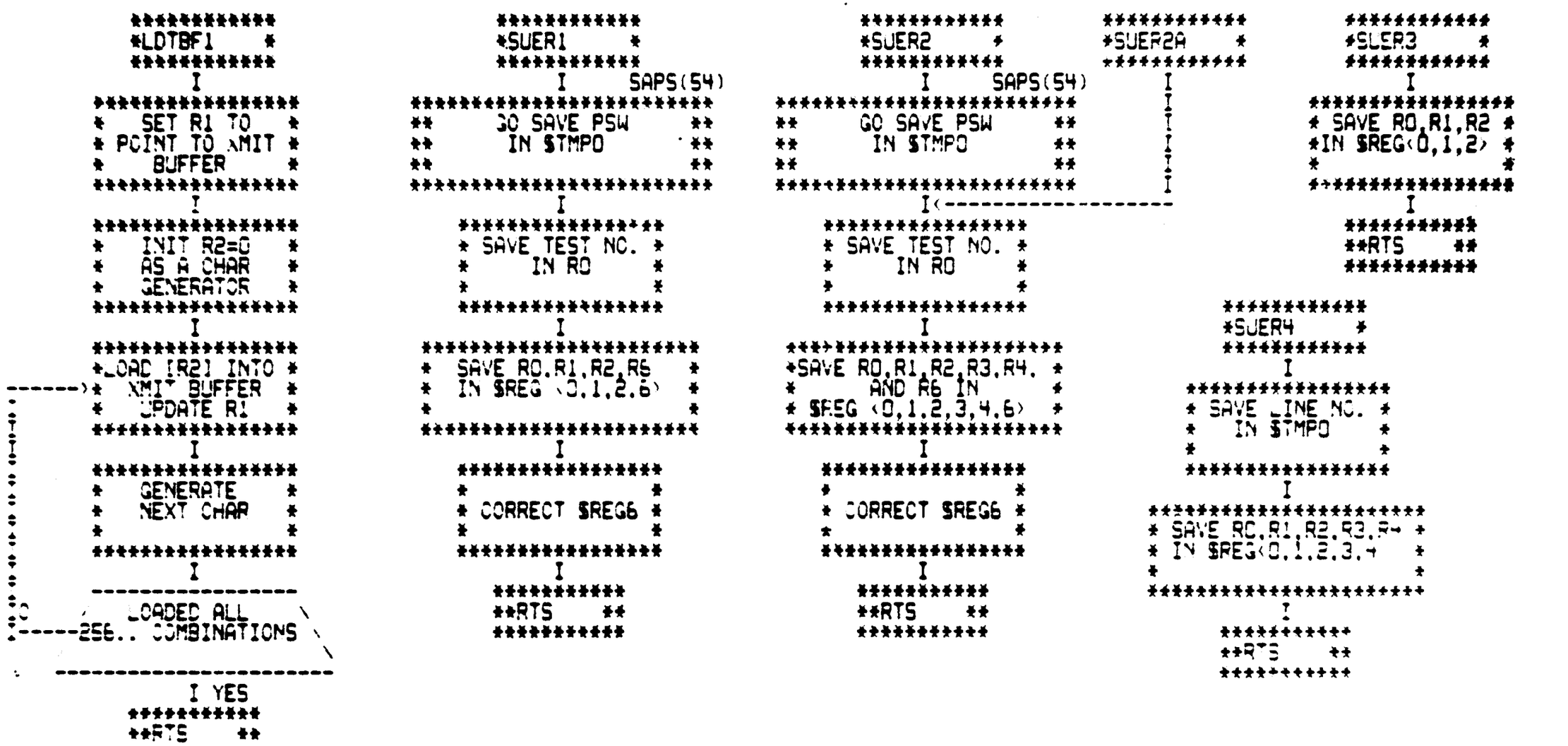
A
[SR<7:0>=TSTNM]
B
FIRST PASS ?
INCREMENT ITER-ATION COUNTER
DONE SPECIFIED NO. OF ITERATIONS ?
C
D
YES
NO

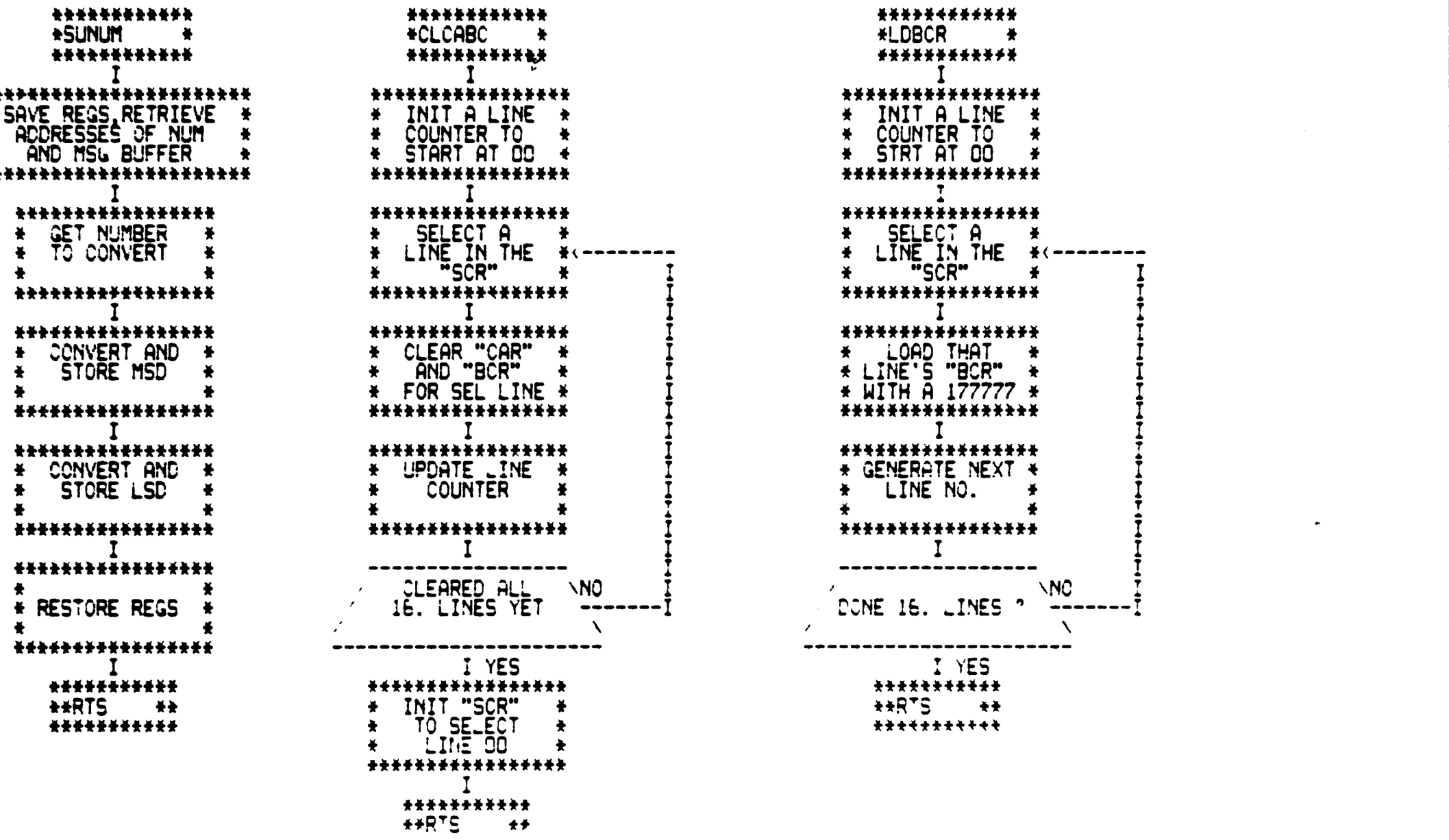

$ERROR
INCREMENT THE ERROR FLAG
DID ERROR FLAG GO TO ZERO ?
YES
NO
DISPLAY TEST NO. INCREMENT ERROR COUNT GET ERROR PC
CORRECT ERROR PC. GET ERROR TRAP INSTR. SAVE ERROR ITEM CODE
INHIBIT ERROR TYPEOUTS ? SR13=1
YES
NO
$ERRTYP
GO TYPE THE ERROR MESSAGE ($ERRTYP)
GO TYPE A CARRIAGE RETURN LINE FEED
HALT ON ERROR ? SR15=1
YES
NO
HALT
LOOP ON ERROR
YES
NO
ESCAPE CURRENT TEST ON ERROR ?
YES
NO
RETURN TO CALLING TEST
SET UP STACK TO RETURN TO ERROR LOOP ADDRESS
SET UP STACK TO RETURN TO ESCAPE ADDRESS

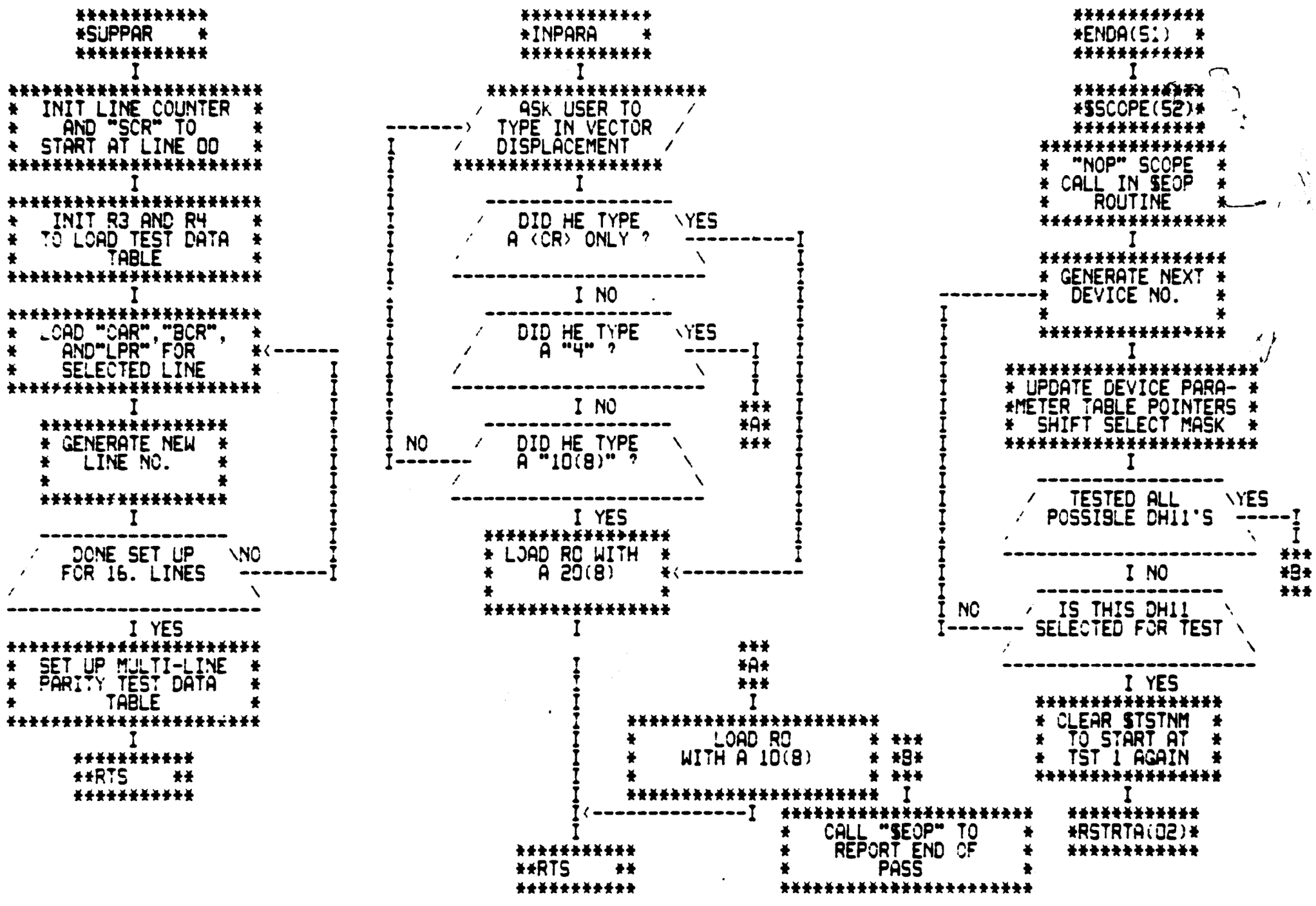


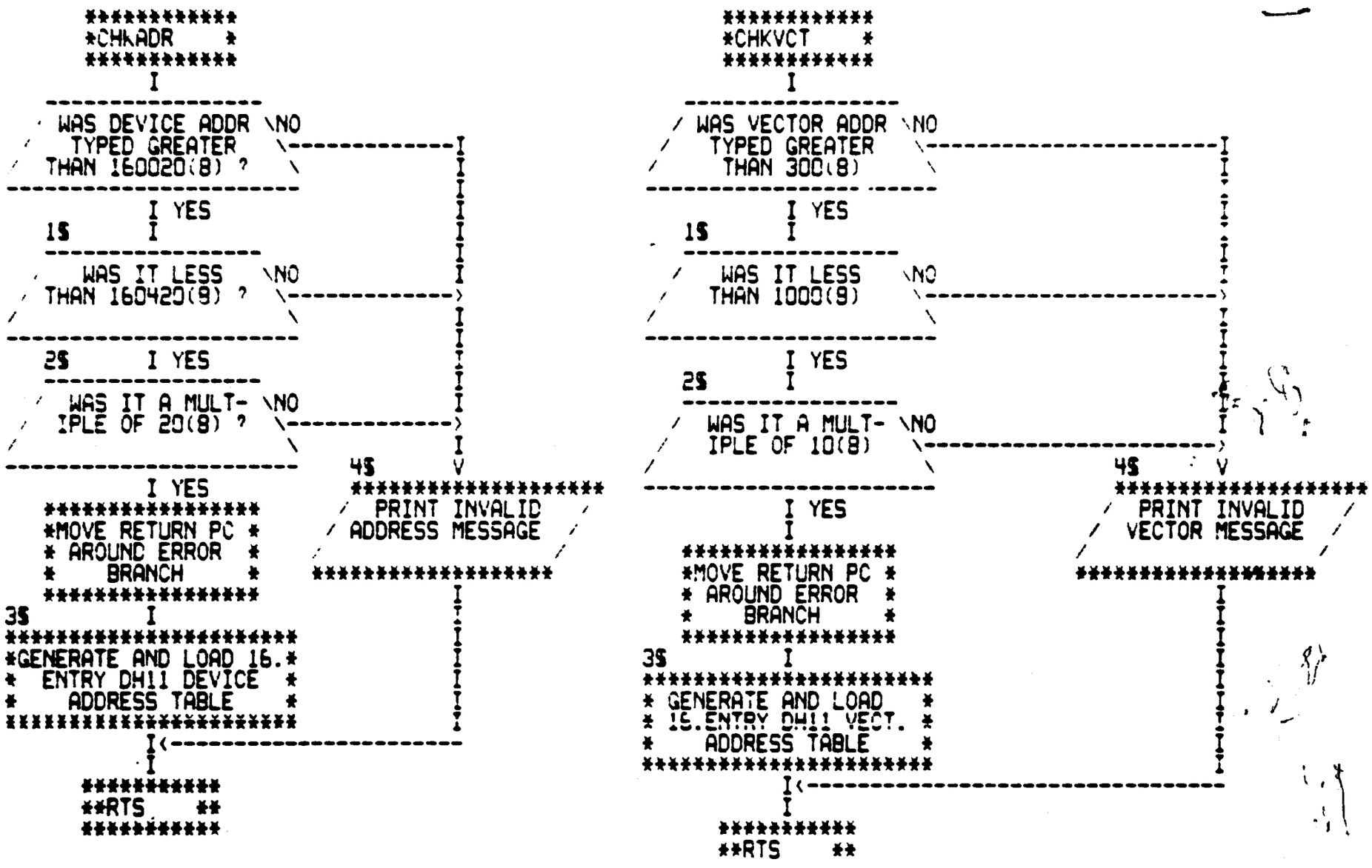
NOTE: I"CHPS1", "CHPS2", AND "SAPS" ARE CALLED WHENEVER
THE MAINLINE CODE HAS TO CLEAR THE PSW, LOCK OUT INTRS,
AND SAVE THE PSW RESPECTIVELY

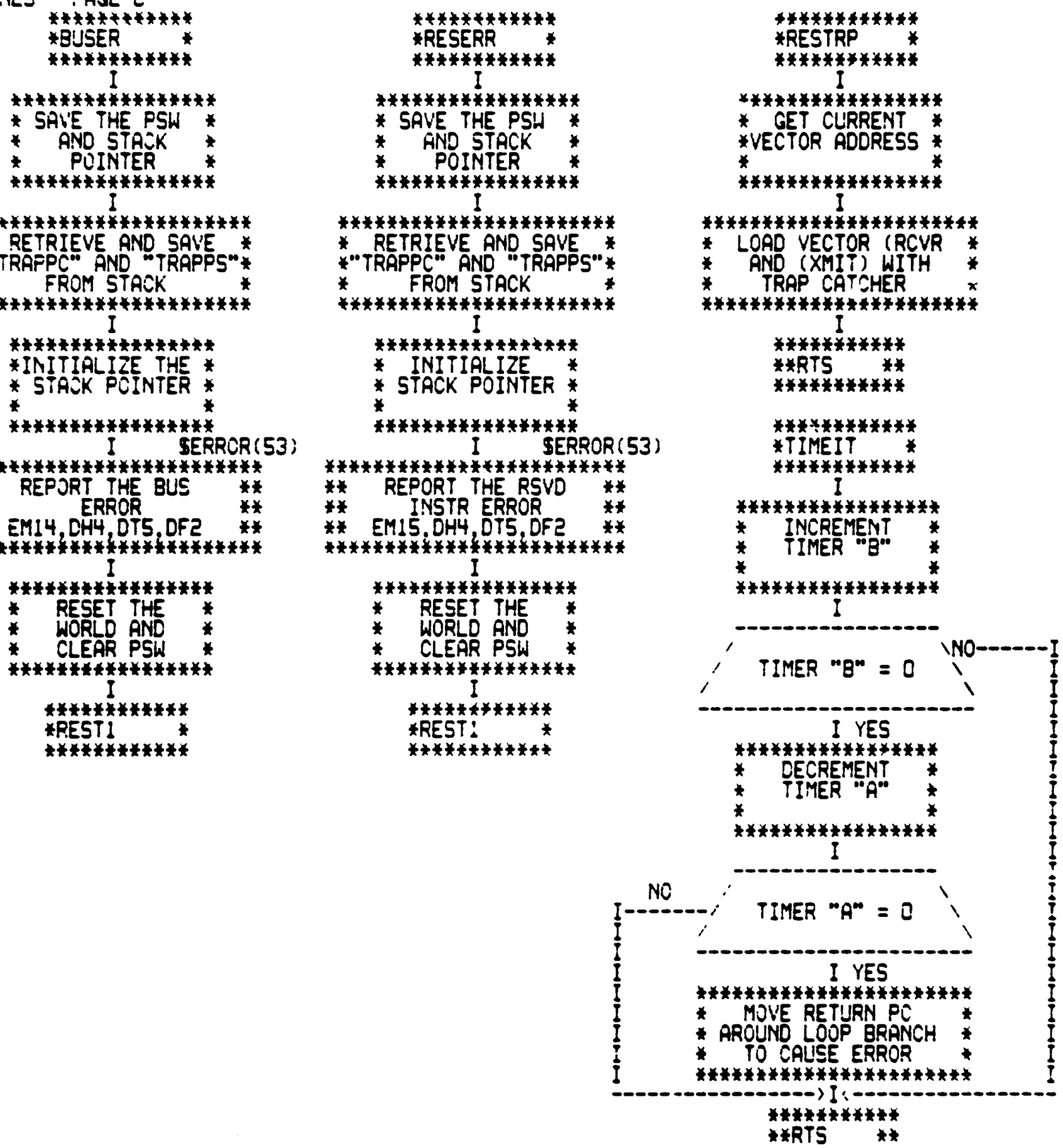
THESE ROUTINES ARE REQUIRED FOR LSI11 COMPATIBILITY

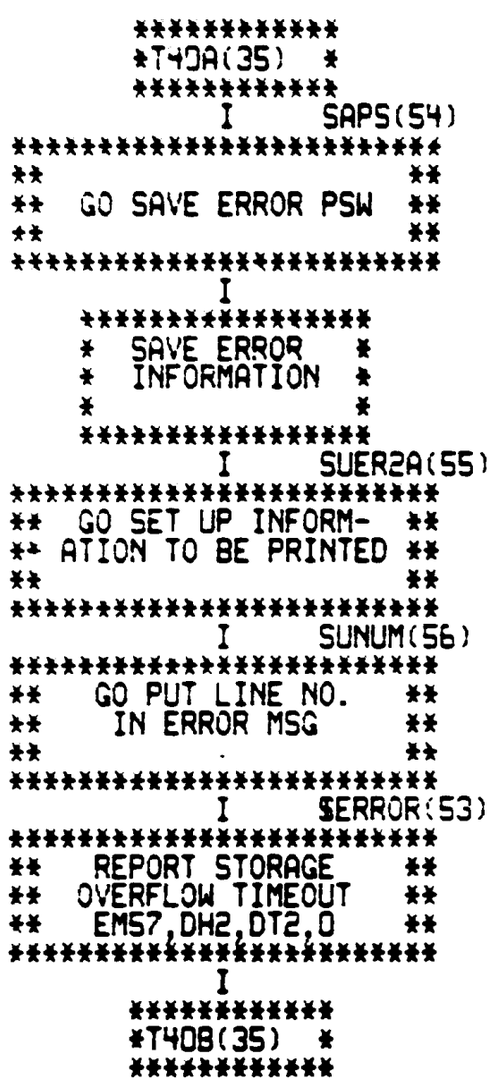

*LDTBF1
SET R1 TO POINT TO XMIT BUFFER
INIT R2=0 AS A CHAR GENERATOR
LOAD [R2] INTO XMIT BUFFER UPDATE R1
GENERATE NEXT CHAR
LOADED ALL 256.. COMBINATIONS
YES
**RTS
*SUER1
SAPS(54)
GO SAVE PSW IN $TMPO
SAVE TEST NO. IN R0
SAVE R0,R1,R2,R5 IN $REG <0,1,2,5>
CORRECT $REG6
**RTS
*SUER2
SAPS(54)
GO SAVE PSW IN $TMPO
SAVE TEST NO. IN R0
SAVE R0,R1,R2,R3,R4, AND R6 IN $REG <0,1,2,3,4,6>
CORRECT $REG6
**RTS
*SUER2A
*SLER3
SAVE R0,R1,R2 IN $REG<0,1,2>
**RTS
*SUER4
SAVE LINE NO. IN $TMPO
SAVE R0,R1,R2,R3,R4 IN $REG<0,1,2,3,4
**RTS

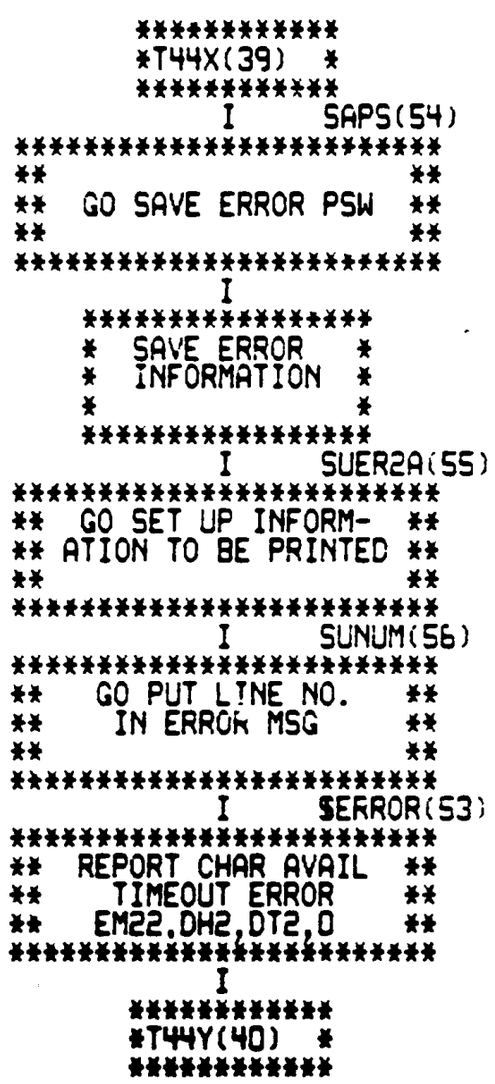

*SUNUM
SAVE REGS,RETRIEVE ADDRESSES OF NUM AND MSG BUFFER
GET NUMBER TO CONVERT
CONVERT AND STORE MSD
CONVERT AND STORE LSD
RESTORE REGS
**RTS
*CLCABC
INIT A LINE COUNTER TO START AT 00
SELECT A LINE IN THE "SCR"
CLEAR "CAR" AND "BCR" FOR SEL LINE
UPDATE LINE COUNTER
CLEARED ALL 16. LINES YET
NO
YES
INIT "SCR" TO SELECT LINE 00
**RTS
*LDBCR
INIT A LINE COUNTER TO STRT AT 00
SELECT A LINE IN THE "SCR"
LOAD THAT LINE'S "BCR" WITH A 177777
GENERATE NEXT LINE NO.
DONE 16. LINES ?
NO
YES
**RTS

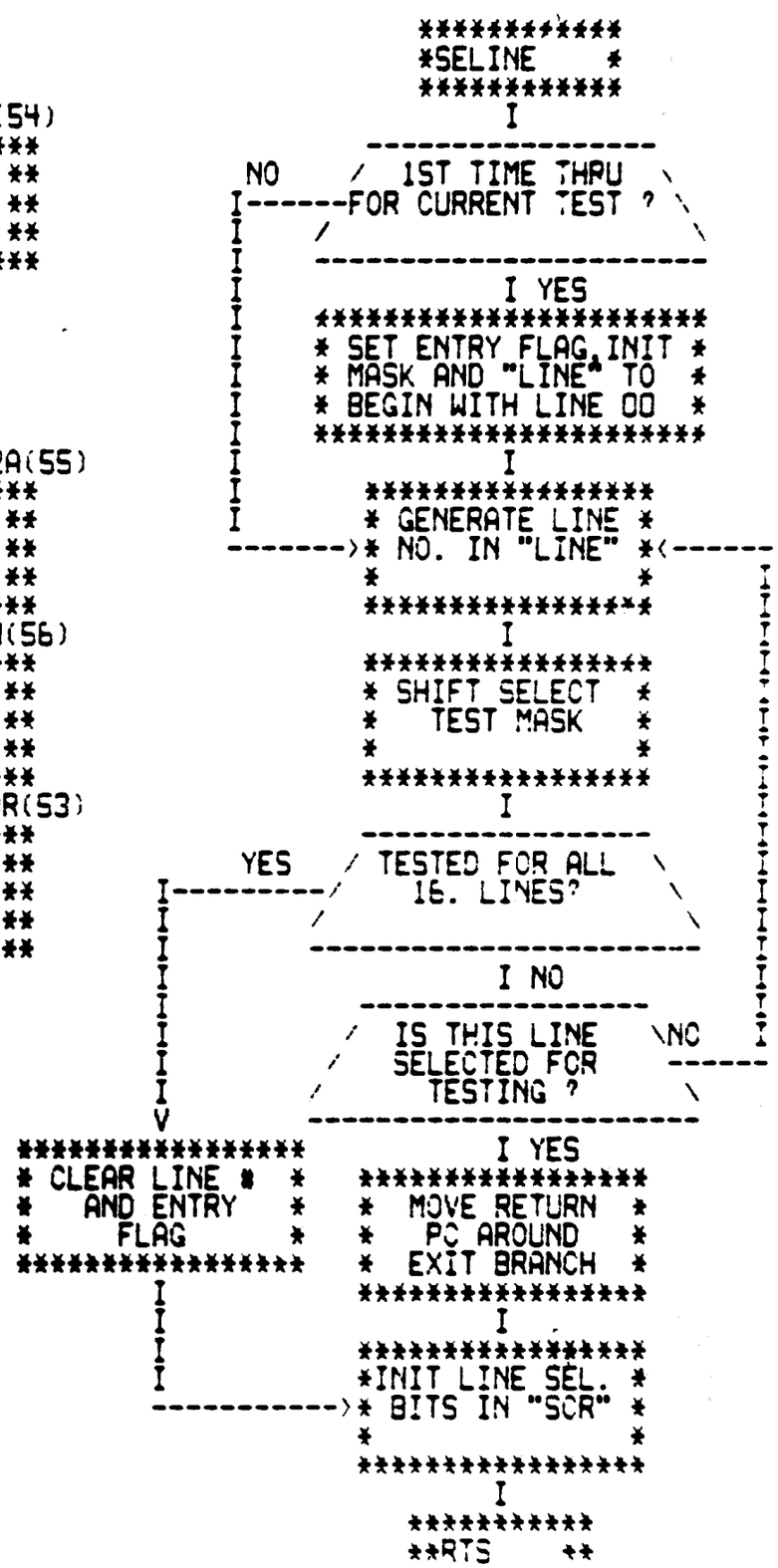

*SUPPAR
INIT LINE COUNTER AND "SCR" TO START AT LINE 00
INIT R3 AND R4 TO LOAD TEST DATA TABLE
LOAD "CAR","BCR", AND"LPR" FOR SELECTED LINE
GENERATE NEW LINE NO.
DONE SET UP FOR 16. LINES
NO
YES
SET UP MULTI-LINE PARITY TEST DATA TABLE
**RTS
*INPARA
ASK USER TO TYPE IN VECTOR DISPLACEMENT
DID HE TYPE A <CR> ONLY ?
YES
NO
DID HE TYPE A "4" ?
DID HE TYPE A "10(8)" ?
LOAD R0 WITH A 20(8)
LOAD R0 WITH A 10(8)
CALL "$EOP" TO REPORT END OF PASS
*ENDA(S:)
$SCOPE(52)
"NOP" SCOPE CALL IN $EOP ROUTINE
GENERATE NEXT DEVICE NO.
UPDATE DEVICE PARA- METER TABLE POINTERS SHIFT SELECT MASK
TESTED ALL POSSIBLE DH11'S
IS THIS DH11 SELECTED FOR TEST
CLEAR $TSTNM TO START AT TST 1 AGAIN
RSTRTA(02)
A
B


*CHKADR
WAS DEVICE ADDR TYPED GREATER THAN 160020(8) ?
NO
1$
YES
WAS IT LESS THAN 160420(8) ?
NO
2$
YES
WAS IT A MULT- IPLE OF 20(8) ?
NO
4$
PRINT INVALID ADDRESS MESSAGE
YES
*MOVE RETURN PC * AROUND ERROR * BRANCH *
3$
GENERATE AND LOAD 16. * ENTRY DH11 DEVICE * * ADDRESS TABLE *
**RTS **
*CHKVCT
WAS VECTOR ADDR TYPED GREATER THAN 300(8)
NO
1$
YES
WAS IT LESS THAN 1000(8)
NO
2$
YES
WAS IT A MULT- IPLE OF 10(8)
NO
4$
PRINT INVALID VECTOR MESSAGE
YES
*MOVE RETURN PC * AROUND ERROR * BRANCH *
3$
* GENERATE AND LOAD * * 1S.ENTRY DH11 VECT. * * ADDRESS TABLE *
**RTS **


*BUSER *
SAVE THE PSW AND STACK POINTER
RETRIEVE AND SAVE "TRAPPC" AND "TRAPPS" FROM STACK
INITIALIZE THE STACK POINTER
$ERROR(53)
REPORT THE BUS ERROR
EM14,DH4,DT5,DF2
RESET THE WORLD AND CLEAR PSW
*REST1 *
*RESERR *
SAVE THE PSW AND STACK POINTER
RETRIEVE AND SAVE "TRAPPC" AND "TRAPPS" FROM STACK
INITIALIZE STACK POINTER
$ERROR(53)
REPORT THE RSVD INSTR ERROR
EM15,DH4,DT5,DF2
RESET THE WORLD AND CLEAR PSW
*REST1 *
*RESTRP *
GET CURRENT VECTOR ADDRESS
LOAD VECTOR (RCVR AND (XMIT) WITH TRAP CATCHER
**RTS **
*TIMEIT *
INCREMENT TIMER "B"
TIMER "B" = 0
NO
I YES
DECREMENT TIMER "A"
TIMER "A" = 0
NO
I YES
MOVE RETURN PC AROUND LOOP BRANCH TO CAUSE ERROR
**RTS **


*T40A(35) *
SAPS(54)
** GO SAVE ERROR PSW **
SAVE ERROR INFORMATION
SUER2A(55)
** GO SET UP INFORM- **
** ATION TO BE PRINTED **
SUNUM(56)
** GO PUT LINE NO. **
** IN ERROR MSG **
$ERROR(53)
** REPORT STORAGE **
** OVERFLOW TIMEOUT **
** EM57,DH2,DT2,0 **
*T40B(35) *


*T44X(39) *
SAPS(54)
** GO SAVE ERROR PSW **
SAVE ERROR INFORMATION
SUER2A(55)
** GO SET UP INFORM- **
** ATION TO BE PRINTED **
SUNUM(56)
** GO PUT LINE NO. **
** IN ERROR MSG **
$ERROR(53)
** REPORT CHAR AVAIL **
** TIMEOUT ERROR **
** EM22,DH2,DT2,0 **
*T44Y(40) *


*SELINE *
1ST TIME THRU FOR CURRENT TEST ?
NO
YES
SET ENTRY FLAG, INIT MASK AND "LINE" TO BEGIN WITH LINE 00
GENERATE LINE NO. IN "LINE"
SHIFT SELECT TEST MASK
TESTED FOR ALL 16. LINES?
NO
IS THIS LINE SELECTED FOR TESTING ?
NO
YES
CLEAR LINE & ENTRY FLAG
MOVE RETURN PC AROUND EXIT BRANCH
*INIT LINE SEL. *
BITS IN "SCR"
**RTS **

AE1	45	45															
AE2	45	45															
AE3	45	46	45														
AE4	45	46	46														
AE5	45	45	46														
BEGINA	01	01															
BRK1	48	49															
BRK2	48	49															
BRK3	48	49															
BRK4	48	49	49														
BRK5	48	49															
BUSER	59																
CHKADR	02	58															
CHKVCT	02	58															
CHPS1	54																
CHPS2	54																
CLCABC	28	30	50	56													
ENDA	51	57															
HALT	02	53															
INPAR	01	02															
INPAR3	01	02	02														
INPARA	01	57															
LDBCR	30	56															
LDTBF1	02	55															
NPR1A	28	29															
NPR2A	28	29															
NPR3A	28	29															
NPR4A	28	29															
NPR5A	28	29															
NPR6A	28	29															
RESERR	59																
REST1	59	59															
RESTRP	59																
RESTRT	01																
RSTRTA	01	02	57														
RTI	43																
RTS	54	54	54	54	55	55	55	55	55	56	56	56	57	57	58	58	59	59
	60																	
SAPS	54	55	55	60	60													
SELINE	14	15	16	17	18	19	20	21	22	28	30	35	36	37	38	39	41	44
	45	48	50	51	60													
SETALL	54																	
SLD1	39	40																
SLD2	39	40	40															
SLD3	39	40	40	40														
START	01*																	
START1	01																	
START2	01	01	01	02														
SUER1	03	55																
SUER2	04	05	05	06	07	07	07	08	08	09	09	11	11	12	12	13	13	14
	15	16	16	17	17	22	32	33	38	39	55							

	23	24	24	25	25	26	26	27	27	28	29	29	29	29	30	31	31	31
	34	34	35	35	36	36	37	37	38	40	40	40	41	41	43	44	44	45
	46	47	48	49	49	49	50	50	51	51	55	60	60					
SUER3	55																	
SUER4	18	19	20	21	30	30	55											
SUNUM	02	14	15	18	19	20	21	22	28	29	30	35	35	36	36	37	37	38
	38	39	40	40	40	41	41	43	43	44	44	45	45	46	47	48	49	49
	49	50	50	51	51	56	60	60										
SUPPAR	42	57																
T40A	35	60																
T40B	36	60																
T44X	39	60																
T44Y	40	60																
T46A	42	43																
T46B	42	43	43	43														
TIMEIT	28	30	35	36	37	38	39	41	42	44	45	47	48	48	50	51	59	
TN	52	52																
TST1	02	03																
TST10	10																	
TST11	11																	
TST12	12																	
TST13	13																	
TST14	14																	
TST15	15																	
TST16	16																	
TST17	17																	
TST2	04																	
TST20	18																	
TST21	19																	
TST22	20																	
TST23	21																	
TST24	22																	
TST25	23																	
TST26	24																	
TST27	25																	
TST3	05																	
TST30	26																	
TST31	27																	
TST32	28																	
TST33	30																	
TST34	31																	
TST35	32																	
TST36	33																	
TST37	34																	
TST4	06																	
TST40	35																	
TST41	36																	
TST42	37																	
TST43	38																	
TST44	39																	
TST45	41																	
TST46	42																	

TST5	07																	
TST50	45																	
TST51	47																	
TST52	48																	
TST53	50																	
TST54	51																	
TST6	08																	
TST7	09																	
$ERROR	03	04	05	05	06	07	07	07	08	08	03	09	11	11	12	12	13	13
	14	16	16	17	17	18	19	20	21	22	23	23	24	24	25	25	26	26
	27	27	28	29	29	29	29	30	30	30	31	31	31	32	33	34	34	35
	35	36	36	37	37	38	38	39	40	40	40	41	41	42	42	43	43	43
	44	44	45	45	46	47	47	48	49	49	49	50	50	51	51	53	59	59
	60	60																
$ERTYP	53																	
$OVER	52																	
$SCOPE	03	04	05	06	07	08	09		11	12	13	14	15	16	17	18	19	20
	21	22	23	24	25	26	27	[illegible]	30	31	32	33	34	35	36	37	38	39
	41	42	44	45	47	48	50	52	57									
$SVLAD	52																	
1$	03	04	05	06	07	08	09	11	12	13	14	15	16	17	18	19	20	21
	22	24	25	26	27	28	30	31	32	33	34	36	37	38	41	54	54	54
	58	58																
11$	18	19																
2$	03	04	05	06	07	08	09	11	12	13	16	17	18	19	20	21	22	24
	25	26	27	28	31	32	34	36	37	41	42	54	58	58				
3$	03	05	06	07	08	09	11	12	13	14	16	17	24	25	26	27	28	30
	31	32	33	34	41	58	58											
4$	05	07	08	09	11	12	13	14	15	18	19	20	21	22	28	31	32	36
	37	41	58	58														
5$	07	14	15	18	20	21	22	28	31	33	34	34	34	36	37			
6$	14	15	18	19	20	21	22	28	33	34								
7$	22	28	36	37														
8$	28	36	37															

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

PRODUCT CODE: MD-11-DZDHM VERSION 000.000

PRODUCT NAME: DH11 DIAGNOSTIC

RELEASE DATE: JANUARY 1976

AUTHOR: ED CROWLEY

****

PRODUCT CODE: MD-11-DZDHM-B (REV B)

PRODUCT NAME: DH11 DIAGNOSTIC

RELEASE DATE: SEPTEMBER 1976

SUBMITTED BY: PAUL F. CANTIN

****

PROGRAM CHANGES:

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

2. WHEN A DH11 REGISTER FAILS TO RESPOND DURING TEST 1, THE PROGRAM ENTERS AN ENDLESS LOOP CAUSING ERROR MESSAGE 1 TO BE PRINTED CONTINUOUSLY. CODE HAS BEEN CORRECTED.

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

4. IN ORDER TO FACILITATE INSTALLATION CHECKOUT WHERE MODEM CONTROL IS INVOLVED, TESTS 101, 105, 106, AND 107 OF DZDHK HAVE BEEN INSERTED. THIS ALLOWS ALL THE LEVEL CONVERTERS AND CABLES TO BE CHECKED WITH JUST ONE PROGRAM USING THE H315 TURNAROUND CONNECTORS.

5. CODE HAS BEEN INSERTED TO INHIBIT FALSE ERROR REPORTING WHICH SOMETIMES OCCURS WHEN H315 TURNAROUND CONNECTORS ARE USED ON SOME LINES WHILE OTHER LINES REMAIN OPEN.

6. TEST 40 HAS BEEN RELOCATED BETWEEN TEST 42 AND TEST 43.

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

DOCUMENT CHANGES:

1. FLOW CHARTS HAVE BEEN DELETED ALONG WITH SECTIONS 6.2D AND 6.5.

2. CHANGE "WORDS BETWEEN VECTORS (4(8) OR 10(8))" REFERENCES TO "ADDRESSES BETWEEN VECTORS (10(8) OR 20(8))", WITH APPROPRIATE CODING CHANGES.

SEQ 0117

3. PARAGRAPH ON SWITCHLESS FEATURE INSERTED.

4. AUTOSIZER STARTING PROCEDURE DESCRIBED.

5. CONSOLE SWITCH REGISTER OPRIONS TABLE INSERTED.

6. SEVERAL ADDITIONS TO SECTION 6.3 WERE MADE FOR CLARITY.

****

.TITLE

```
                                .NLIST  CND,MD,MC
                                .LIST   TOC,ME,SEQ,BIN
            165400              $SWR=165400
                                .MCALL  .HEADER,.$CMTAG,.SETUP,.$ERROR
                                .MCALL  .SWRHI,.SWRLO,.$CATCH,.EQUATE,.SETUP,.$EOP,.GETSWR
                                .MCALL  .$SCOPE,.$ERRTYP,.$TYPOCT,.$TYPDEC,.$TYPE,.$POWER,.$TRAP
                                .MCALL  .$READ,.$RDOCT
                                .MCALL  .$ACT11,.$APTHDR,.$APTBLS,.$APTYPE
                                .ENABLE ABS
                                .TITLE  MAINDEC-11-DZDHM-B
                                ;*COPYRIGHT (C) 1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY ED CROWLEY
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                                ;*
            000001              $TN=1
                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>
                                .SBTTL  TRAP CATCHER

            000000                      .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
            000174                      .=174
000174  000000                  DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                  SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  026114                  JMP     @#INPARX        ;;JUMP TO STARTING ADDRESS OF PROGRAM
000204  000137  002146                  JMP     @#BEGIN         ;BEGIN EXECUTION WITH DEFAULT PARAMETERS
000210  000137  026104                  JMP     @#INPARC        ;INPUT PARAMETERS - DEVICE SELECTION ONLY

                                .SBTTL  ACT11 HOOKS

                                ;;******************************************************************
                                ;HOOKS REQUIRED BY ACT11
            000214                      $SVPC=.                 ;SAVE PC
            000046                      .=46
000046  020712                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
            000052                      .=52
000052  120000                          .WORD   120000          ;;2)SET LOC.52 TO 120000
            000214                      .=$SVPC                 ;; RESTORE PC
                                .SBTTL  APT PARAMETER BLOCK
```

```
                                ;;*****************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;*****************************************************************
                    000214              .$X=.       ;;SAVE CURRENT LOCATION
                    000024              .=24        ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                          200         ;;FOR APT START UP
                    000044              .=44        ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  000214                          $APTHDR     ;;POINT TO APT HEADER BLOCK
                    000214              .=.$X       ;;RESET LOCATION COUNTER
                                ;;*****************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

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


                                .SBTTL  BASIC DEFINITIONS

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


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


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


                                ;*PRIORITY LEVEL DEFINITIONS
                    000000      PR0=    0               ;;PRIORITY LEVEL 0
                    000040      PR1=    40              ;;PRIORITY LEVEL 1
```

```
                    
000100              PR2=    100             ;;PRIORITY LEVEL 2
000140              PR3=    140             ;;PRIORITY LEVEL 3
000200              PR4=    200             ;;PRIORITY LEVEL 4
000240              PR5=    240             ;;PRIORITY LEVEL 5
000300              PR6=    300             ;;PRIORITY LEVEL 6
000340              PR7=    340             ;;PRIORITY LEVEL 7
                    
                    ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000              SW15=   100000
040000              SW14=   40000
020000              SW13=   20000
010000              SW12=   10000
004000              SW11=   4000
002000              SW10=   2000
001000              SW09=   1000
000400              SW08=   400
000200              SW07=   200
000100              SW06=   100
000040              SW05=   40
000020              SW04=   20
000010              SW03=   10
000004              SW02=   4
000002              SW01=   2
000001              SW00=   1
                    .EQUIV  SW09,SW9
                    .EQUIV  SW08,SW8
                    .EQUIV  SW07,SW7
                    .EQUIV  SW06,SW6
                    .EQUIV  SW05,SW5
                    .EQUIV  SW04,SW4
                    .EQUIV  SW03,SW3
                    .EQUIV  SW02,SW2
                    .EQUIV  SW01,SW1
                    .EQUIV  SW00,SW0
                    
                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000              BIT15=  100000
040000              BIT14=  40000
020000              BIT13=  20000
010000              BIT12=  10000
004000              BIT11=  4000
002000              BIT10=  2000
001000              BIT09=  1000
000400              BIT08=  400
000200              BIT07=  200
000100              BIT06=  100
000040              BIT05=  40
000020              BIT04=  20
000010              BIT03=  10
000004              BIT02=  4
000002              BIT01=  2
000001              BIT00=  1
                    .EQUIV  BIT09,BIT9
                    .EQUIV  BIT08,BIT8
                    .EQUIV  BIT07,BIT7
                    .EQUIV  BIT06,BIT6
```

```
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

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

```
                                .SBTTL  COMMON TAGS

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

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

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

                                ;;**************************************************************
                                .EVEN
                                $MAIL:                          ;;APT MAILBOX
                                $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
                                $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
                                $TESTN: .WORD   ATESTN          ;;TEST NUMBER
                                $PASS:  .WORD   APASS           ;;PASS COUNT
                                $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
                                $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
                                $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
                                $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
                                $ETABLE:                        ;;APT ENVIRONMENT TABLE
                                $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
                                $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
                                $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
                                $USWR:  .WORD   AUSWR           ;;USER SWITCHES
                                $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
                                ;*                              BITS 15-11=CPU TYPE
                                ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                                      11/70=06,PDQ=07,Q=10
                                ;*                              BIT 10=REAL TIME CLOCK
                                ;*                              BIT  9=FLOATING POINT PROCESSOR
                                ;*                              BIT  8=MEMORY MANAGEMENT
                                $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
                                $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
                                ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                                      900 NSEC CORE=001
                                ;*                                      300 NSEC BIPOLAR=002
                                ;*                                      500 NSEC MOS=003
                                $MADR1: .WORD   AMADR1          ;;HIGH ADDRESS,BLK#1
                                ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
                                $MAMS2: .BYTE   AMAMS2          ;;HIGH ADDRESS,M.S. BYTE
                                $MTYP2: .BYTE   AMTYP2          ;;MEM.TYPE,BLK#2
                                $MADR2: .WORD   AMADR2          ;;MEM.LAST ADDRESS,BLK#2
                                $MAMS3: .BYTE   AMAMS3          ;;HIGH ADDRESS,M.S.BYTE
                                $MTYP3: .BYTE   AMTYP3          ;;MEM.TYPE,BLK#3
                                $MADR3: .WORD   AMADR3          ;;MEM.LAST ADDRESS,BLK#3
                                $MAMS4: .BYTE   AMAMS4          ;;HIGH ADDRESS,M.S.BYTE
                                $MTYP4: .BYTE   AMTYP4          ;;MEM.TYPE,BLK#4
                                $MADR4: .WORD   AMADR4          ;;MEM.LAST ADDRESS,BLK#4
                                $VECT1: .WORD   AVECT1          ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
                                $VECT2: .WORD   AVECT2          ;;INTERRUPT VECTOR#2BUS PRIORITY#2
                                $BASE:  .WORD   ABASE           ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
                                $DEVM:  .WORD   ADEVM           ;;DEVICE MAP
                                $CDW1:  .WORD   ACDW1           ;;CONTROLLER DESCRIPTION WORD#1
                                $CDW2:  .WORD   ACDW2           ;;CONTROLLER DESCRIPTION WORD#2
                                $DDW0:  .WORD   ADDW0           ;;DEVICE DESCRIPTOR WORD#0
                                $DDW1:  .WORD   ADDW1           ;;DEVICE DESCRIPTOR WORD#1
                                $DDW2:  .WORD   ADDW2           ;;DEVICE DESCRIPTOR WORD#2
```

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

```
                                .SBTTL  ERROR POINTER TABLE

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

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


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

001356  030250                          EM1             ;"DH11 REGISTER REFERENCE CAUSED TIMEOUT"
001360  030317                          DH1             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR "
001362  030376                          DT1             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2
001364  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 2

001366  030414                          EM2             ;"SYSTEM CONTROL REGISTER ERROR"
001370  030452                          DH2             ; (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001372  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001374  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 3

001376  030572                          EM3             ;"DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG"
001400  030452                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001402  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001404  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 4

001406  030650                          EM4             ;"LINE PARAMETER REGISTER ERROR"
001410  030452                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001412  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001414  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 5

001416  030706                          EM5             ;"BREAK CONTROL REGISTER ERROR"
001420  030452                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001422  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001424  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 6

001426  030743                          EM6             ;"SILO STATUS REGISTER ERROR"
001430  030452                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001432  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001434  000000                          0               ;PRINT ALL OCTAL
```

```
                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 7

001436  030776                          EM7             ;"CURRENT ADDRESS REGISTER ERROR - LINE #XX"
001440  030452                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001442  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001444  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 10

001446  031050                          EM10            ;"BYTE COUNTER REGISTER ERROR - LINE #XX"
001450  030452                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001452  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001454  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 11

001456  031117                          EM11            ;"UNEXPECTED DH11 RCVR INTERRUPT"
001460  030452                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001462  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001464  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 12

001466  031156                          EM12            ;"UNEXPECTED DH11 XMITTR INTERRUPT"
001470  030452                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001472  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001474  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 13

001476  031217                          EM13            ;"CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT"
001500  030452                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001502  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001504  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 14

001506  031300                          EM14            ;"TRANSMITTER NPR LOGIC ERROR - LINE #  "
001510  030452                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001512  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001514  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 15

001516  031347                          EM15            ;"XMITTR FAILED TO INTERRUPT - LINE #  "
001520  030452                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001522  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001524  000000                          0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 16

001526  031415                          EM16            ;"RCVR FAILED TO INTERRUPT"
001530  030452                          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001532  030550                          DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001534  000000                          0               ;PRINT ALL OCTAL
```

```
                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 17

001536  031446                        EM17            ;"TRANSMITTER TIMING ERROR - LINE #  "
001540  031512                        DH6             ;" (PC)  (PS)  (SP)  TEST  DEVADR  SPEED   TIMEB   TIMEC"
001542  030550                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001544  000000                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 20

001546  031610                        EM20            ;RECEIVER TIMING ERROR - LINE #  "
001550  031512                        DH6             ;" (PC)  (PS)  (SP)  TEST  DEVADR  SPEED   TIMEB   TIMEC"
001552  030550                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001554  000000                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 21

001556  031651                        EM21            ;"RCVR FAILED TO INTERRUPT - LINE #  "
001560  030452                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001562  030550                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001564  000000                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 22

001566  031715                        EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
001570  030452                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001572  030550                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001574  000000                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 23

001576  031771                        EM23            ;"BASIC DATA TEST ERROR - LINE #  "
001600  032032                        DH7             ;" (PC)  (PS)  (SP)  TEST  DEVADR  CHRLNG  WAS  S/B "
001602  030550                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001604  000000                        0               ;PRINT ALL OCTAL


                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 24

001606  032127                        EM24            ;"AUTO ECHO TEST ERROR - LINE #  "
001610  030452                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001612  030550                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001614  000000                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 25

001616  032167                        EM25            ;"BREAK BIT TEST ERROR - LINE #  "
001620  030452                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001622  030550                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001624  000000                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 26

001626  032227                        EM26            ;"HALF-DUPLEX TEST ERROR - LINE #  "
001630  030452                        DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001632  030550                        DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
```

```
001634  000000                  0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 27

001636  032271          EM27            ;"UNEXPECTED BUS ERROR TRAP"
001640  032323          DH3             ; (PC)  (PS)  (SP)  TEST  TRPPC  TRPPS
001642  032402          DT3             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2
001644  000000          0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 30

001646  032420          EM30            ;"UNEXPECTED RSVD INSTR TRAP"
001650  032323          DH3             ; (PC)  (PS)  (SP)  TEST  TRPPC  TRPPS
001652  032402          DT3             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2
001654  000000          0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 31

001656  032453          EM31            ;"AUTO ECHO DATA COMPARE ERROR - LINE #  "
001660  032523          DH4             ; (PC)   (PS)   (SP)   TEST   WASADR  SBADR   WAS    S B "
001662  030550          DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001664  000000          0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 32

001666  032620          EM32            ;"AUTO ECHO TEST TIMEOUT - LINE #  "
001670  032662          DH5             ;" (PC)  (LPRG)   TEST"
001672  032710          DT4             ;$ERRPC,$TMP0,$TMP2
001674  000000          0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 33

001676  032720          EM33            ;"PARITY LOGIC TEST ERROR - LINE #  "
001700  030452          DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S S"
001702  030550          DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001704  000000          0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 34

001706  032763          EM34            ;"MULTI-LINE PARITY DATA TEST ERROR - LINE #   - LBTEST #  "
001710  032523          DH4             ;" (PC)  (PS)  (SP)  TEST  WASADR  SBADR  WAS  S B "
001712  030550          DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001714  000000          0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 35

001716  033056          EM35            ;"MULTI-LINE PARITY DATA TEST TIMEOUT"
001720  033122          DH14            ;" (PC)  (LPRG)  LINACT "
001722  033152          DT6             ;"$ERRPC,$TMP0,$TMP3"
001724  000000          0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 36

001726  033162          EM36            ;CHAR AVAILABLE TIMEOUT"
001730  032662          DH5             ;" (PC)  (LPRG)   TEST"
```

```
001732  032710                  DT4             ;$ERRPC,$TMP0,$TMP2
001734  000000                  0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 37

001736  033211                  EM37            ;"DATA COMPARE ERROR - LINE #  "
001740  032523                  DH4             ;"  (PC)  (PS)  (SP)  TEST  WASADR  SBADR  WAS    S "
001742  030550                  DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$RE
001744  000000                  0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERRO. MESSAGE 40

001746  033247                  EM40            ;"BUFFER ACTIVE REG ERROR - LINE #  "
001750  030452                  DH2             ;"  (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
001752  030550                  DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
001754  000000                  0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 41

001756  033312                  EM41            ;"RCVR FALSE INTERRUPT"
001760  032662                  DH5             ;"  (PC)  (LPRG)  TEST"
001762  032710                  DT4             ;$ERRPC,$TMP0,$TMP2
001764  000000                  0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 42

001766  033337                  EM42            ;"SILO OVERFLOW ERROR"
001770  032662                  DH5             ;"  (PC)  (LPRG)  TEST"
001772  032710                  DT4             ;$ERRPC,$TMP0,$TMP2
001774  000000                  0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 43

001776  033363                  EM43            ;"SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT"
002000  030452                  DH2             ;"  (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S B "
002002  030550                  DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
002004  000000                  0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 44

002006  033443                  EM44            ;"NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT"
002010  030452                  DH2             ;"  (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S B "
002012  030550                  DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
002014  000000                  0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 45

002016  033525                  EM45            ;"XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT"
002020  030452                  DH2             ;"  (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S B "
002022  030550                  DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
002024  000000                  0               ;PRINT ALL OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 46

002026  033603                  EM46            ;"CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE #  "
002030  033671                  DH10            ;"  (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS     S B"
```

```
                                        DTS             ;$ERRPC,$TMPO,$TMP1,$REGO,$REG1,$REG2,$REG3,$REG4
                                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 47

                                        EM47            ;"BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE #  "
                                        DH10            ;" (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS     S/B"
                                        DTS             ;$ERRPC,$TMPO,$TMP1,$REGO,$REG1,$REG2,$REG3,$REG4
                                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 50

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

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 51

                                        EM51            ;"NPR LOGIC TEST 2 ERROR"
                                        DH11            ;" (PC)   LINACT  LINCHK   TEST   DEVADR  REGADR   WAS     S B"
                                        DTS             ;$ERRPC,$TMPO,$TMP1,$REGO,$REG1,$REG2,$REG3,$REG4"
                                        0               ;PRINT ALL OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 52

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

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 53

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

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 54

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

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 55

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

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 56

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

```
659  002132  030550                    DT2          ;$ERRPC,$TMPO,$REG6,$REGO,$REG1,$REG2,$REG3,$REG4"
660  002134  000000                    0            ;PRINT ALL OCTAL
661
662                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 57
663
664  002136  034565                    EM57         ;"STORAGE OVERFLOW BIT FAILED - LINE #  "
665  002140  030452                    DH2          ;"  (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
666  002142  030550                    DT2          ;$ERRPC,$TMPO,$REG6,$REGO,$REG1,$REG2,$REG3,$REG4"
667  002144  000000                    0            ;PRINT ALL OCTAL
668
```

```
                                BEGIN:  CLR     R0              ;INIT R0 TO INDICATE DEFAULT PARAMETERS
                                BEGINA: CLR     TITFLG          ;INIT TITLE MESSAGE FLAG
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
                                        MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
                                        CLR     (R6)+           ;;CLEAR MEMORY LOCATION
                                        CMP     #SWR,R6 ;;DONE?
                                        BNE     .-6             ;;LOOP BACK IF NO
                                        MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
                                        MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
                                        MOV     #340,@#IOTVEC+2 ;;LEVEL 7
                                        MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
                                        MOV     #340,@#EMTVEC+2 ;;LEVEL 7
                                        MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
                                        MOV     #340,@#TRAPVEC+2;LEVEL 7
                                        MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
                                        MOV     #340,@#PWRVEC+2 ;;LEVEL 7
                                        MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
                                        CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
                                        CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
                                        MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
                                        MOV     #..,$LPADR      ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
                                        MOV     #..,$LPERR      ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
                                        MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
                                        MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
                                        MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
                                        MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
                                        CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
                                        BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
                                        BR      65$             ;;BRANCH IF NO TIMEOUT
                                64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
                                        RTI
                                65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
                                        MOV     #DISPREG,DISPLAY
                                66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

                                        CLR     $PASS           ;;CLEAR PASS COUNT
                                        BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
                                        BEQ     67$             ;;YES,USE NON-APT SWITCH
                                        MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
                                67$:
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
                                        TST     @#42            ;;ARE WE RUNNING UNDER XXDP'ACT?
                                        BNE     68$             ;;BRANCH IF YES
                                        CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
                                        BEQ     68$             ;;BRANCH IF YES
                                        CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
                                        BNE     69$             ;;BRANCH IF NO
                                        GTSWR                   ;;GET SOFT-SWR SETTINGS
                                        BR      69$
                                68$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
                                69$:
```

```
002464  012767  026566  175312  START1: MOV     #BUSER,ERRVEC   ;SET UP THE BUS ERROR VECTOR
002472  012767  000340  175306          MOV     #340,ERRVEC+2
002500  012767  026642  175302          MOV     #RESERR,RESVEC  ;SET UP THE RSVD INSTR VECTOR
002506  012767  000340  175275          MOV     #340,RESVEC+2
002514  005767  025516                  TST     TITFLG          ;HAVE WE TYPED TITLE ONCE ?
002520  001012                          BNE     1$              ;BR IF YES
002522  104401                          TYPE                    ;GO TYPE PROGRAM TITLE
002524  034634                          TITLE
002526  005167  025504                  COM     TITFLG          ;SET FLAG - TYPE TITLE ONLY ONCE PER LOAD
002532  032777  000001  176400          BIT     #BIT0,@SWR      ;DO WE WANT TO AUTOSIZE?
002540  001002                          BNE     1$              ;BRANCH IF NOT.
002542  004767  022330                  JSR     PC,AUTOSZ       ;GO AUTOSIZE.
002546  005767  025116          1$:     TST     VCFLG           ;START AT 200 ??
002552  001413                          BEQ     11$             ;BR IF NOT
002554  032777  000001  176356          BIT     #BIT0,@SWR      ;ARE PARAMETERS TO BE INPUT MANUALLY?
002562  001003                          BNE     9$              ;BRANCH IF YES.
002564  016700  025406                  MOV     ADRVEC,R0       ;OTHERWISE, GET ADDRESSES BETWEEN VECTORS FROM AUTOSIZER
002570  000402                          BR      10$
002572  004767  023250          9$:     JSR     PC,INPARA       ;GO ASK FOR PARAMETERS
002576  005067  025066          10$:    CLR     VCFLG           ;RE INIT VECTOR FLAG
002602  005700                  11$:    TST     R0              ;USE DEFAULT PARAMETERS ?
002604  001407                          BEQ     START2          ;BR IF YES
002606  022700  177777                  CMP     #-1,R0          ;CHANGE DH SELECT PARAM ONLY ?
002612  001002                          BNE     2$              ;BR IF NOT
002614  000167  023375                  JMP     INPAR3          ;GO ASK FOR SELECT PARAM.
002620  000167  023302          2$:     JMP     INPAR           ;GO ASK FOR ALL PARAMETERS

002624  012767  027566  025364  START2: MOV     #DHADTB-2,ADPTR ;GET POINTER TO ADDRESS TABLE
002632  012767  027626  025360          MOV     #DHVCTB-2,VCPTR ;GET POINTER TO VECTOR TABLE
002640  012767  027670  025354          MOV     #BRLVL-2,BRPTR  ;GET POINTER TO BR LEVEL TABLE
002646  012767  177777  025334          MOV     #-1,DHNUM       ;START WITH DH #00
002654  012767  000001  024314          MOV     #1,SELMSK       ;SET UP DH11 BIT TEST MARKER

002662  005267  025322          RESTRT: INC     DHNUM           ;GENERATE DH11 DEV NUMBER
002666  062767  000002  025322          ADD     #2,ADPTR        ;UPDATE TABLE POINTERS
002674  062767  000002  025316          ADD     #2,VCPTR
002702  062767  000002  025312          ADD     #2,BRPTR
002710  036767  024262  024252          BIT     SELMSK,DHSEL    ;TEST FOR SELECTED DH11
002716  001004                          BNE     RSTRTA          ;BR IF SELECTED FOR TEST
002720  006367  024252          REST1:  ASL     SELMSK          ;SHIFT MARKER TO TEST NEXT DH11
002724  001737                          BEQ     START2          ;BR IF 16 TESTED - START OVER
002726  000755                          BR      RESTRT          ;GO TEST IF THIS ONE SELECTED
002730  017767  025262  024234  RSTRTA: MOV     @ADPTR,DHADR    ;SET UP DH11 ADDRESS
002736  017767  025256  024230          MOV     @VCPTR,DHVCT    ;SET UP THE DH11 VECTOR ENTRY
002744  017767  025252  025234          MOV     @BRPTR,DHRLVL   ;GET BR LEVEL VALUES
002752  004567  021616                  JSR     R5,SUNUM        ;GO SET DH NUMBER IN THE MESSAGE BUFFER
002756  030210                          DHNUM
002760  034724                          TITLE2+20
002762  104401                          TYPE                    ;GO PRINT "TESTING DH11 #XX"
002764  034704                          TITLE2
002766  004767  021274                  JSR     PC,LDTBF1       ;GO LOAD XMITTR OUTPUT BUFFER WITH
                                                                ;BINARY COUNT PATTERN
002772  012767  002772  176106          MOV     #.,$LPADR       ;INIT SCOPE LOOP RETURN
```

```
                        ;;*****************************************************************
                        ;*TEST 1        CHECK SSYN RESPONSE FROM ALL DH11 REGISTERS
                        ;;*****************************************************************
                        ;
003000  000004          TST1:   SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        *****************

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

                        ERRORS:
                        *******

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

                        SYNC:           (NONE)
                        *****

                        DEBUG:
                        ******

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

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

                                M7277   SH3     SSYN H                  CE2
                                                DEVICE RESPONDING L     E72-6
                                        SH4     DEVICE SELECTED H       E09-11
                        %
003002  010102                  MOV     R1,R2           ;COPY IT INTO R2
003004  010205                  MOV     R2,R5           ;ALSO R5
003006  062705  000020          ADD     #20,R5          ;R5 WILL TELL US WHEN WE'VE TESTED ALL 8
003012  016746  174766          MOV     ERRVEC,-(SP)    ;SAVE BUS ERROR VECTOR
003016  012767  003044  174760  MOV     #3$,ERRVEC      ;GO TO 3$ IF REG FAILS TO RESPOND
003024  162702  000002          SUB     #2,R2           ;SO WE START WITH FIRST REG

003030  062702  000002  1$:     ADD     #2,R2           ;POINT TO A DH11 REGISTER
003034  020205                  CMP     R2,R5           ;TESTED ALL EIGHT ??
003036  001412                  BEQ     4$              ;BR IF YES
003040  005712          2$:     TST     (R2)            ;ACCESS DH11 REG ADDR
003042  000772                  BR      1$              ;BR WHEN ALL 8 ARE DONE

003044  004767  021240  3$:     JSR     PC,SUER1        ;GO SET UP ERROR INFO
003050  022626                  CMP     (SP)+,(SP)+     ;FIX SP BECAUSE OF TRAP
```

```
  834  003052  012767  003040  176030          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
  835  003060  104001                          ERROR   1               ;DH11 REGISTER FAILED TO RESPOND TO MSYN
  836
  837  003062  000762                          BR      1$              ;GO TEST NEXT ONE
  838  003064  012667  174714          4$:     MOV     (SP)+,.ERRVEC   ;RESTORE BUS ERROR VECTOR
```

```
                        ;;*************************************************************
                        ;*TEST 2        TEST THAT "MASTER CLR" CAN CLEAR THE "SCR","LPR","BKR",AND "SSR" REGS
                        ;;*************************************************************
003070  000004          TST2:   SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

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

                        ERRORS:
                        *******

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

                        SYNC:
                        *****

                                M7289   SH6     SCR 11 H (MASTER CLEAR)         FK2

                        DEBUG:
                        ******

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

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

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

                                M7289   SH6     SCR 11 H (MASTER CLEAR) FK2

                                M7278   SH3     BUFF DATA 11 H          AA1
                        %
003072  012705  027210          MOV     #MSTCLR,R5      ;GET POINTER TO ADDRESS TABLE
003076  010125                  MOV     R1,(R5)+        ;SET UP THE TEST ADDRESS TABLE SO THAT
003100  010125                  MOV     R1,(R5)+        ;IT CONTAINS THE ADDRESSES OF THE
003102  010125                  MOV     R1,(R5)+        ;SCR,LPR,BKR, AND SSR REGISTERS
003104  010125                  MOV     R1,(R5)+
003106  062745  000016          ADD     #SSR,-(R5)      ;GENERATE SSR ADDRESS
003112  062745  000014          ADD     #BKR,-(R5)      ;GENERATE BKR ADDRESS
```

```
  895  003116  062745  000004                ADD     #LPR,-(R5)      ;GENERATE LPR ADDRESS
  896  003122  005745                        TST     -(R5)           ;POINT R5 TO FIRST ADDR ENTRY (SCR)
  897  003124  005004                        CLR     R4              ;RESULT S/B 000000 AFTER MASTER CLEAR
  898
  899  003126  012502                1$:     MOV     (R5)+,R2        ;GET REG ADDRESS
  900  003130  022705  027220                CMP     #MSTCLR+10,R5   ;DONE ALL FOUR REGS ??
  901  003134  001415                        BEQ     TST3            ;;BR IF YES
  902  003136  012712  177777        2$:     MOV     #-1,(R2)        ;SET 1'S IN REGISTER
  903  003142  052711  004000                BIS     #BIT11,(R1)     ;ISSUE MASTER CLEAR
  904  003146  011203                        MOV     (R2),R3         ;GET CONTENT OF REGISTER
  905  003150  001766                        BEQ     1$              ;;BR IF IT'S ALL ZEROES
  906
  907  003152  004767  021172                JSR     PC,SUER2        ;GO SET UP ERROR INFO
  908  003156  012767  003136  175724        MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
  909  003164  104003                        ERROR   3               ;MASTER CLR FAILED TO CLR SEL. REG.
  910  003166  000757                        BR      1$              ;GO TEST NEXT REGISTER
```

```
                        ;;*****************************************************************
                        ;*TEST 3        TEST "SCR" REG R/W BITS CAN SET/CLR (NORMAL MODE)
                        ;;*****************************************************************
003170  000004          TST3:   SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

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

                        ERRORS:
                        *******

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

                        SYNC:
                        *****

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

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

                        DEBUG:
                        ******

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

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

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

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

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

```
                                                                                
   967  003172  012767  003222  175710          MOV     #4$,$LPERR      ;SET UP ERROR LOOP RETURN
   968  003200  010102                          MOV     R1,R2           ;GET REGISTER ADDRESS
   969  003202  012705  000001                  MOV     #1,R5           ;SET UP TO START WITH BIT00
   970
   971  003206  030567  024340          1$:     BIT     R5,RGMSK1       ;SHALL WE TEST THIS BIT ?
   972  003212  001003                          BNE     4$              ;BR IF YES
   973  003214  006305                  2$:     ASL     R5              ;SHIFT TO TST NEXT BIT
   974  003216                          3$:
   975  003216  001430                          BEQ     TST4            ;;<BR IF DONE ALL R/W BITS>
   976  003220  000772                          BR      1$              ;GO TEST NEXT BIT
   977
   978  003222  010504                  4$:     MOV     R5,R4           ;RESULT S/B IN R4
   979  003224  005012                          CLR     (R2)            ;INIT REG BEING TESTED
   980  003226  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
   981  003234  010512                          MOV     R5,(R2)         ;SET THE BIT
   982  003236  011203                          MOV     (R2),R3         ;GET THE WAS DATA
   983  003240  020403                          CMP     R4,R3           ;RESULT = S/B DATA ??
   984  003242  001403                          BEQ     5$              ;BR IF YES
   985
   986  003244  004767  021100                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
   987  003250  104002                          ERROR   2               ;SELECTED BIT FAILED TO SET IN SCR
   988
   989  003252  005004                  5$:     CLR     R4              ;SET UP TO CLEAR THE BIT S/B=000000
   990  003254  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
   991  003262  040512                          BIC     R5,(R2)         ;CLR THE SELECTED BIT
   992  003264  011203                          MOV     (R2),R3         ;GET THE WAS DATA
   993  003266  001403                          BEQ     6$              ;BR IF IT CLEARED
   994
   995  003270  004767  021054                  JSR     PC,SUER2        ;GO SET UP THE ERROR INFO
   996  003274  104002                          ERROR   2               ;SELECTED BIT FAILED TO CLEAR IN SCR
   997  003276  000746                  6$:     BR      2$              ;GO SELECT NEXT BIT
```

```
                        ;;*****************************************************************
                        ;*TEST 4        TEST "SCR" REG. READ ONLY BITS (NORMAL MODE)
                        ;;*****************************************************************
003300  000004          TST4:   SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

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

                        ERRORS:
                        *******

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

                        SYNC:
                        *****

                                M7277   SH4     LOAD SSR LOW BYTE H     CR1

                        DEBUG:
                        ******

                                SAME AS FOR TEST 03

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

                                SAME AS FOR TEST 03

                        %
003302  010102                  MOV     R1,R2           ;MAKE IT THE REG. ADDR ALSO
003304  012705  000001          MOV     #1,R5           ;INIT BIT TEST MARKER
003310  030567  024240  1$:     BIT     R5,RGMSK2       ;IS IT A READ ONLY BIT ?
003314  001003                  BNE     3$              ;BR IF IT IS - GO TEST IT
003316  006305          2$:     ASL     R5              ;SHIFT BIT MARKER
003320  001420                  BEQ     TST5            ;;BR IF DONE ALL BITS
003322  000772                  BR      1$              ;GO TEST THIS BIT

003324  005004          3$:     CLR     R4              ;RESULT S/B = 000000
003326  005012                  CLR     (R2)            ;INIT REG BEING TESTED
003330  112761  000000  000016  MOVB    #0,SSR(R1)      ;SCOPE SYNC
003336  010512                  MOV     R5,(R2)         ;ATTEMPT TO SET A READ ONLY BIT
003340  011203                  MOV     (R2),R3         ;GET THE WAS DATA
003342  001765                  BEQ     2$              ;BR IF THE BIT DIDN'T SET

003344  004767  021000          JSR     PC,SUER2        ;GO SET UP ERROR INFO
003350  012767  003324  175532  MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN ADDR
003356  104002                  ERROR   2               ;READ ONLY BIT SET IN "SCR"
003360  000756                  BR      2$              ;CONTINUE WITH NEXT BIT
```

```
                        ;;*****************************************************************
                        ;*TEST 5        TEST "SCR" REG. BITS THAT CAN BE SET/CLR IN MAINT. MODE
                        ;;*****************************************************************
003362  000004          TST5:   SCOPE
                        .REM    :
                        TEST ABSTRACT:
                        ***************

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

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

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

                        ERRORS:
                        ********

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

                        SYNC:
                        ******

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

                                M7277   SH4     LOAD SSR LOW BYTE H     CR1

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

                                M7277   SH4     LOAD SSR HIGH BYTE H    CP2

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

                                M7277   SH4     LOAD LPR H              EP2

                        DEBUG:
                        ******

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

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

```
                                        SAME AS TEST 03 WITH THE FOLLOWING ADDITION

                                        M7289   SH4     74121 ONE-SHOTS E35-6, E23-6

003364  012767  003414  175516        MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
003372  010102                        MOV     R1,R2           ;MAKE IT REG ADDR TOO
003374  012705  000001                MOV     #1,R5           ;INIT BIT TEST MARKER
003400  030567  024160          1$:   BIT     R5,RGMSK6       ;IS IT A READ ONLY BIT ??
003404  001003                        BNE     3$              ;BR IF YES - TEST IT
003406  006305                  2$:   ASL     R5              ;SHIFT THE BIT MARKER
003410  001457                        BEQ     TST6            ;;BR IF DONE ALL SELECTED BITS
003412  000772                        BR      1$              ;GO TEST FOR THIS BIT

003414  010504                  3$:   MOV     R5,R4           ;SET UP S/B DATA
003416  052704  001000                BIS     #BIT09,R4       ;PUT IN THE MAINT. BIT
003422  005012                        CLR     (R2)            ;INIT REG BEING TESTED
003424  052712  001000                BIS     #BIT09,(R2)     ;TURN ON MAINT. MODE
003430  112761  000000  000016        MOVB    #0,SSR(R1)      ;SCOPE SYNC
003436  050512                        BIS     R5,(R2)         ;SET THE SELECTED BIT
003440  011203                        MOV     (R2),R3         ;GET THE WAS DATA
003442  020304                        CMP     R3,R4           ;DID SELECTED BIT GET SET ??
003444  001404                        BEQ     4$              ;BR IF IT DID

003446  004767  020676                JSR     PC,SUER2        ;GO SET UP ERROR INFO
003452  104002                        ERROR   2               ;SELECTED BIT FAILED TO SET IN MAINT MODE
003454  000754                        BR      2$              ;GO TEST NEXT BIT

003456  042712  001000          4$:   BIC     #BIT09,(R2)     ;TURN OFF MAINT. MODE
003462  042704  001000                BIC     #BIT09,R4       ;CLR MAINT BIT IN S/B DATA
003466  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
003474  040512                        BIC     R5,(R2)         ;ATTEMPT TO CLR SELECTED BIT
003476  011203                        MOV     (R2),R3         ;GET THE WAS DATA
003500  020304                        CMP     R3,R4           ;DID BIT GET CLEARED ??
003502  001404                        BEQ     5$              ;BR IF IT DIDN'T

003504  004767  020640                JSR     PC,SUER2        ;GO SET UP ERROR INFO
003510  104002                        ERROR   2               ;SELECTED BIT GOT CLEARED WITH MAINT MODE OFF
003512  000735                        BR      2$              ;GO TEST NEXT BIT

003514  012704  001000          5$:   MOV     #BIT09,R4       ;SET UP S/B DATA
003520  050412                        BIS     R4,(R2)         ;SET MAINT. MODE
003522  012761  000000  000004        MOV     #0,LPR(R1)      ;SCOPE SYNC
003530  040512                        BIC     R5,(R2)         ;NOW CLR SELECTED BIT
003532  011203                        MOV     (R2),R3         ;GET THE WAS DATA
003534  020304                        CMP     R3,R4           ;DID BIT GET CLEARED OK ??
003536  001723                        BEQ     2$              ;BR IF YES

003540  004767  020604                JSR     PC,SUER2        ;GO SET UP ERROR INFO
003544  104002                        ERROR   2               ;FAILED TO CLR SELECTED BIT IN MAINT MODE
003546  000717                        BR      2$              ;GO SELECT NEXT BIT FOR TEST
```

```
                        ;;*****************************************************************
                        ;*TEST 6        TEST THAT ALL R/W BITS IN "LPR" CAN BE SET/CLR
                        ;;*****************************************************************
003550  000004          TST6:   SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        ***************

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

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

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

                        ERRORS:
                        *******

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

                        SYNC:
                        *******

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

                        DEBUG:
                        ******

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

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

                                M7277   SH4     LOAD LPR H              EP2

                                M7278   SH5     LPR <15:12> L   (E52)
                                        SH6     LPR <11:08> L   (E37)
                                        SH7     LPR <07:04> L   (E59)
                                        SH8     LPR <03:00> L   (E61)
                                        SH5,6,7,8       OUTPUT MUX CHIPS  (74151'S PIN 3)

                        %
003552  012767  003606  175330        MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
003560  010102                        MOV     R1,R2           ;COPY IT IN R2
003562  062702  000004                ADD     #LPR,R2         ;GENERATE REGADR IN R2
```

```
 1217  003566  012705  000001          MOV     #1,R5           ;INIT BIT TEST MARKER
 1218  003572  030567  023760    1$:   BIT     R5,RGMSK3       ;TEST THIS BIT ??
 1219  003576  001003                  BNE     3$              ;BR IF YES
 1220  003600  006305                2$:   ASL     R5              ;SHIFT THE MARKER
 1221  003602  001430                  BEQ     TST7            ;BR IF DONE ALL BITS
 1222  003604  000772                  BR      1$              ;GO TEST NXT BIT
 1223
 1224  003606  010504                3$:   MOV     R5,R4           ;SET UP S/B DATA
 1225  003610  005012                  CLR     (R2)            ;INIT REG BEING TESTED
 1226  003612  112761  000000  000016      MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
 1227  003620  010512                  MOV     R5,(R2)         ;SET LPR BIT
 1228  003622  011203                  MOV     (R2),R3         ;GET THE WAS DATA
 1229  003624  020304                  CMP     R3,R4           ;DID IT SET
 1230  003626  001403                  BEQ     4$              ;BR IF IT SET PROPERLY
 1231
 1232  003630  004767  020514          JSR     PC,SUER2        ;GO SET UP ERROR INFO
 1233  003634  104004                  ERROR   4               ;LPR BIT FAILED TO SET PROPERLY
 1234
 1235  003636  005004                4$:   CLR     R4              ;GET READY TO CLEAR SELECTED BIT
 1236  003640  112761  000000  000017      MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
 1237  003646  040512                  BIC     R5,(R2)         ;CLEAR THE BIT
 1238  003650  011203                  MOV     (R2),R3         ;GET THE WAS DATA
 1239  003652  001752                  BEQ     2$              ;BR IF BIT CLEARED PROPERLY
 1240
 1241  003654  004767  020470          JSR     PC,SUER2        ;GO SET UP ERROR INFO
 1242  003660  104004                  ERROR   4               ;LPR BIT FAILED TO CLEAR PROPERLY
 1243  003662  000746                  BR      2$              ;GO SELECT NEXT BIT
```

```
                        ;;*****************************************************************
                        ;*TEST 7        TEST THAT ALL R/W BITS IN "BKR" CAN BE SET/CLR
                        ;;*****************************************************************
003664  000004          T$T7:   SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

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

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

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

                        ERRORS:
                        ********

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

                        SYNC:
                        ******

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

                        DEBUG:
                        ******

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

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

                        3.) IF ONLY ONE OR TWO FAIL THE PROBLEM IS MOST LIKELY THE M7279.

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

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

                                M7278   SH5 - SH8 74175 REGISTER CHIPS (E51,E38,E67,E60)
                                        SH5 - SH8 74151'S MUX CHIPS INPUT PIN 13
                        %
003666  012767  003722  175214        MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
003674  010102                        MOV     R1,R2           ;GENERATE "BKR" ADDRESS IN R2
```

```
1300  003676  062702  000014            ADD     #BKR,R2
1301  003702  012705  000001            MOV     #1,R5
1302  003706  030567  023646      1$:   BIT     R5,RGMSK4       ;TEST THIS BIT ??
1303  003712  001003                    BNE     3$              ;BR IF YES
1304  003714  006305                2$: ASL     R5              ;SHIFT BIT MARKER
1305  003716  001430                    BEQ     TST10           ;;BR IF ALL BITS TESTED
1306  003720  000772                    BR      1$              ;GO TEST THE BIT
1307
1308  003722  010504                3$: MOV     R5,R4           ;SET UP S/B DATA
1309  003724  005012                    CLR     (R2)            ;INIT REG BEING TESTED
1310  003726  112761  000000  000016    MOVB    #0,SSR(R1)      ;SCOPE SYNC
1311  003734  050512                    BIS     R5,(R2)         ;SET THE SELECTED BIT IN "BKR"
1312  003736  011203                    MOV     (R2),R3         ;GET THE WAS DATA
1313  003740  020304                    CMP     R3,R4           ;DID BIT SET OK
1314  003742  001403                    BEQ     4$              ;BR IF YES
1315
1316  003744  004767  020400            JSR     PC,SUER2        ;GO SET UP ERROR INFO
1317  003750  104005                    ERROR   5               ;BKR BIT FAILED TO SET PROPERLY
1318
1319  003752  005004                4$: CLR     R4              ;SET UP S/B DATA
1320  003754  112761  000000  000017    MOVB    #0,SSR+1(R1)    ;SCOPEE SYNC
1321  003762  040512                    BIC     R5,(R2)         ;CLEAR BKR BIT
1322  003764  011203                    MOV     (R2),R3         ;GET THE BKR WAS DATA
1323  003766  001752                    BEQ     2$              ;BR IF BKR BIT CLEARED OK
1324
1325  003770  004767  020354            JSR     PC,SUER2        ;GO SET UP ERROR INFO
1326  003774  104005                    ERROR   5               ;BKR BIT FAILED TO CLR PROPERLY
1327  003776  000746                    BR      2$              ;GO SELECT NEXT BIT
```

```
                        ;;*****************************************************************
                        ;*TEST 10       TEST THAT ALL R/W BITS IN "SSR" CAN BE SET/CLR
                        ;;*****************************************************************
004000  000004          TST10:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

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

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

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

                        ERRORS:
                        ********

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

                        SYNC:
                        *****

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

                        DEBUG:
                        ******

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

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

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

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

                                M7279   SH5 - SH8       REGISTER CHIPS E53,E68, OR E69 (74175'S)
                                                        OUTPUT MUX CHIPS - (74151'S  PIN 12)
                        %
```

```
1384  004002  012767  004036  175100          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
1385  004010  010102                          MOV     R1,R2           ;GENERATE "SSR" ADDRESS IN R2
1386  004012  062702  000016                  ADD     #SSR,R2
1387  004016  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MARKER
1388  004022  030567  023534          1$:     BIT     R5,RGMSK5       ;TEST THIS BIT ??
1389  004026  001003                          BNE     3$              ;BR IF YES
1390  004030  006305                  2$:     ASL     R5              ;SHIFT BIT MARKER
1391  004032  001435                          BEQ     TST11           ;;BR IF ALL BITS TESTED
1392  004034  000772                          BR      1$              ;GO TEST THE BIT
1393
1394  004036  010504                  3$:     MOV     R5,R4           ;SET UP S/B DATA
1395  004040  005012                          CLR     (R2)            ;INIT REG BEING TESTED
1396  004042  012761  000000  000004          MOV     #0,LPR(R1)      ;SCOPE SYNC
1397  004050  050512                          BIS     R5,(R2)         ;SET THE SELECTED BIT IN "SSR"
1398  004052  011203                          MOV     (R2),R3         ;GET THE WAS DATA
1399  004054  042703  077700                  BIC     #77700,R3       ;CLEAR OUT DON'T CARE BITS
1400  004060  020304                          CMP     R3,R4           ;DID BIT SET OK
1401  004062  001403                          BEQ     4$              ;BR IF YES
1402
1403  004064  004767  020260                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
1404  004070  104006                          ERROR   6               ;SSR BIT FAILED TO SET PROPERLY
1405
1406  004072  005004                  4$:     CLR     R4              ;SET UP S/B DATA
1407  004074  012761  000000  000014          MOV     #0,BKR(R1)      ;SCOPE SYNC
1408  004102  040512                          BIC     R5,(R2)         ;CLEAR SSR BIT
1409  004104  011203                          MOV     (R2),R3         ;GET THE SSR WAS DATA
1410  004106  042703  077700                  BIC     #77700,R3       ;CLEAR JUNK BITS
1411  004112  020304                          CMP     R3,R4           ;DID THE SSR BIT GET CLEARED ??
1412  004114  001745                          BEQ     2$              ;BR IF SSR BIT CLEARED OK
1413
1414  004116  004767  020226                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
1415  004122  104006                          ERROR   6               ;SSR BIT FAILED TO CLR PROPERLY
1416  004124  000741                          BR      2$              ;GO SELECT NEXT BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 11       TEST THAT CLR/SET OF BIT "N" IN "LPR" DOES NOT CLEAR ANY OTHER BITS
                                ;;*****************************************************************
004126  000004                  TST11:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

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

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

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

                                ERRORS:
                                *******

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

                                SYNC:
                                *******

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

                                DEBUG:
                                ******

                                1.) PROBLEMS DETECTED BY THIS TEST INDICATE ADJACENT BIT INTERFERENCE
                                    CAUSED BY CROSS TALK OR NOISE. PROBLEM IS MOST LIKELY THE M7279.

                                KEY LOGIC:      (SAME AS FOR TEST 6)
                                **********
                                %
004130  012767  004164  174752          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
004136  010102                          MOV     R1,R2           ;SET UP THE REG ADDR
004140  062702  000004                  ADD     #LPR,R2
004144  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MASK
004150  030567  023402          1$:     BIT     R5,RGMSK3       ;TEST THIS BIT ??
004154  001003                          BNE     3$              ;BR IF YES
004156  006305          2$:     ASL     R5                      ;SHIFT THE BIT TEST MASK
004160  001436                          BEQ     TST12           ;;BR IF TESTED ALL BITS
004162  000772                          BR      1$              ;GO TEST THIS BIT

004164  016704  023366          3$:     MOV     RGMSK3,R4       ;SET UP S/B DATA
004170  005012                          CLR     (R2)            ;INIT REG BEING TESTED
```

```
 1473  004172  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
 1474  004200  040504                          BIC     R5,R4           ;CLR BIT "N"
 1475  004202  016712  023350                  MOV     RGMSK3,(R2)     ;SET ALL R/W BITS IN LPR
 1476  004206  040512                          BIC     R5,(R2)         ;CLEAR BIT "N" IN LPR
 1477  004210  011203                          MOV     (R2),R3         ;GET THE WAS DATA
 1478  004212  020304                          CMP     R3,R4           ;DID IT CLEAR OK ?
 1479  004214  001404                          BEQ     4$              ;BR IF YES
 1480
 1481  004216  004767  020126                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
 1482  004222  104004                          ERROR   4               ;BIT "N" FAILED TO CLR PROPERLY
 1483  004224  000754                          BR      2$              ;GO TEST NEXT BIT
 1484
 1485  004226  050504                  4$:     BIS     R5,R4           ;SET BIT "N" IN S/B DATA
 1486  004230  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
 1487  004236  050512                          BIS     R5,(R2)         ;SET BIT "N" IN LPR
 1488  004240  011203                          MOV     (R2),R3         ;GET THE WAS DATA
 1489  004242  020304                          CMP     R3,R4           ;DID BIT "N" SET PROPERLY ?
 1490  004244  001744                          BEQ     2$              ;BR IF YES
 1491
 1492  004246  004767  020076                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
 1493  004252  104004                          ERROR   4               ;BIT "N" FAILED TO SET PROPERLY
 1494  004254  000740                          BR      2$              ;GO SELECT NEXT BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 12       TEST THAT CLR/SET OF BIT "N" IN "BKR" DOES NOT CLEAR ANY OTHER BITS
                                ;;*****************************************************************
004256  000004                  TST12:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

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

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

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

                                ERRORS:
                                ********

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

                                SYNC:
                                *****

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

                                DEBUG:
                                ******

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

                                KEY LOGIC:      (SAME AS FOR TEST 7)
                                **********
                                %
004260  012767  004314  174622          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
004266  010102                          MOV     R1,R2           ;SET UP THE REG ADDR
004270  062702  000014                  ADD     #BKR,R2
004274  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MASK
004300  030567  023254          1$:     BIT     R5,RGMSK4       ;TEST THIS BIT ??
004304  001003                          BNE     3$              ;BR IF YES
004306  006305                  2$:     ASL     R5              ;SHIFT THE BIT TEST MASK
004310  001436                          BEQ     TST13           ;;BR IF TESTED ALL BITS
004312  000772                          BR      1$              ;GO TEST THIS BIT

004314  016704  023240          3$:     MOV     RGMSK4,R4       ;SET UP S/B DATA
004320  040504                          BIC     R5,R4           ;CLR BIT "N"
```

```
 1551  004322  005012                          CLR     (R2)              ;INIT REG BEING TESTED
 1552  004324  016712  023230                  MOV     RGMSK4,(R2)       ;SET ALL R/W BITS IN BKR
 1553  004330  112761  000000  000016          MOVB    #0,SSR(R1)        ;SCOPE SYNC
 1554  004336  040512                          BIC     R5,(R2)           ;CLEAR BIT "N" IN BKR
 1555  004340  011203                          MOV     (R2),R3           ;GET THE WAS DATA
 1556  004342  020304                          CMP     R3,R4             ;DID IT CLEAR OK ?
 1557  004344  001404                          BEQ     4$                ;BR IF YES
 1558
 1559  004346  004767  017776                  JSR     PC,SUER2          ;GO SET UP ERROR INFO
 1560  004352  104005                          ERROR   5                 ;BIT "N" FAILED TO CLR PROPERLY
 1561  004354  000754                          BR      2$                ;GO TEST NEXT BIT
 1562
 1563  004356  050504                  4$:     BIS     R5,R4             ;SET BIT "N" IN S/B DATA
 1564  004360  112761  000000  000017          MOVB    #0,SSR+1(R1)      ;SCOPE SYNC
 1565  004366  050512                          BIS     R5,(R2)           ;SET BIT "N" IN BKR
 1566  004370  011203                          MOV     (R2),R3           ;GET THE WAS DATA
 1567  004372  020304                          CMP     R3,R4             ;DID BIT "N" SET PROPERLY ?
 1568  004374  001744                          BEQ     2$                ;BR IF YES
 1569
 1570  004376  004767  017746                  JSR     PC,SUER2          ;GO SET UP ERROR INFO
 1571  004402  104005                          ERROR   5                 ;BIT "N" FAILED TO SET PROPERLY
 1572  004404  000740                          BR      2$                ;GO SELECT NEXT BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 13       TEST THAT CLR/SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER BITS
                                ;;*****************************************************************
004406  000004                  TST13:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

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

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

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

                                ERRORS:
                                ********

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

                                SYNC:
                                *****

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

                                DEBUG:
                                ******

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

                                KEY LOGIC:      (SAME AS FOR TEST 10)
                                **********
                                %
004410  012767  004444  174472          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
004416  010102                          MOV     R1,R2           ;SET UP THE PEG ADDR
004420  062702  000016                  ADD     #SSR,R2
004424  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MASK
004430  030567  023126          1$:     BIT     R5,RGMSK5       ;TEST THIS BIT ??
004434  001003                          BNE     3$              ;BR IF YES
004436  006305                  2$:     ASL     R5              ;SHIFT THE BIT TEST MASK
004440  001442                          BEQ     TST14           ;;BR IF TESTED ALL BITS
004442  000772                          BR      1$              ;GO TEST THIS BIT

004444  016704  023112          3$:     MOV     RGMSK5,R4       ;SET UP S/B DATA
004450  040504                          BIC     R5,R4           ;CLR BIT "N"
```

```
MAINDEC-11-D2DHM-B     MACY11 27(732)  23-SEP-76  14:33  PAGE 39
D2DHMB.P11      T13    TEST THAT CLR SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER BITS                SEQ 0157

 1629  004452  005012                          CLR     (R2)            ;INIT REG BEING TESTED
 1630  004454  016712  023102                  MOV     RGMSK5,(R2)     ;SET ALL R/W BITS IN SSR
 1631  004460  012761  000000  000004          MOV     #0,LPR(R1)      ;SCOPE SYNC
 1632  004466  040512                          BIC     R5,(R2)         ;CLEAR BIT "N" IN SSR
 1633  004470  011203                          MOV     (R2),R3         ;GET THE WAS DATA
 1634  004472  042703  077700                  BIC     #^C77700,R3     ;CLEAR JUNK BITS
 1635  004476  020304                          CMP     R3,R4           ;DID IT CLEAR OK ?
 1636  004500  001404                          BEQ     4$              ;BR IF YES
 1637
 1638  004502  004767  017642                  JSR     PC,SJER2        ;GO SET UP ERROR INFO
 1639  004506  104006                          ERROR   6               ;BIT "N" FAILED TO CLR PROPERLY
 1640  004510  000752                          BR      2$              ;GO TEST NEXT BIT
 1641
 1642  004512  050504                  4$:     BIS     R5,R4           ;SET BIT "N" IN S/B DATA
 1643  004514  012761  000000  000014          MOV     #0,BKR(R1)      ;SCOPE SYNC
 1644  004522  050512                          BIS     R5,(R2)         ;SET BIT "N" IN SSR
 1645  004524  011203                          MOV     (R2),R3         ;GET THE WAS DATA
 1646  004526  042703  077700                  BIC     #^C77700,R3     ;CLEAR JUNK BITS
 1647  004532  020304                          CMP     R3,R4           ;DID BIT "N" SET PROPERLY ?
 1648  004534  001740                          BEQ     2$              ;BR IF YES
 1649
 1650  004536  004767  017606                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
 1651  004542  104006                          ERROR   5               ;BIT "N" FAILED TO SET PROPERLY
 1652  004544  000734                          BR      2$              ;GO SELECT NEXT BIT
```

```
                        ;;*****************************************************************
                        ;*TEST 14       "CAR" MEMORY ADDRESSING TEST
                        ;;*****************************************************************
004546  000004          TST14:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        ***************

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

                                MEMORY PATTERN: LOCATION        CONTENTS        (BOTH OCTAL)

                                                00              000000
                                                01              010421
                                                02              021042
                                                03              031463
                                                04              042104
                                                05              052525
                                                06              063146
                                                07              073567
                                                10              104210
                                                        11                      11463:
                                                12              125252
                                                13              135673
                                                14              146314
                                                15              156735
                                                16              167356
                                                17              177777

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

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

                        ERRORS:
                        ********

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

                        SYNC:
                        ******

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

```
                                DEBUG:
                                ******

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

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

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

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

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

                                5.) IN MOST CASES THE FAULT IS MOST LIKELY THE M7277 OR M7279.

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

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

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

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

                                                M7278   SH5 THRU SH8    74151 DATA MUX OUTPUT CHIPS (PIN 1 INPUT)

                                %
004550  010102                  MOV     R1,R2           ;COPY IT IN R2
004552  062702  000006          ADD     #CAR,R2         ;SET UP REGADR IN R2
004556  016767  022420  174434  MOV     LINSEL,$TMP7    ;SAVE LINE SELECT PARAMETER
004564  012767  177777  022410  MOV     #-1,LINSEL      ;DO ALL LINES FOR THIS TEST
004572  004767  017704  1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
004576  000415                  BR      3$              ;;BR IF DONE ALL SELECTED LINES
004600  105767  023406          TSTB    LINE            ;DOING LINE 00 ?
004604  001001                  BNE     2$              ;BR IF NOT
004606  005004                  CLR     R4              ;INIT TEST DATA

004610  156711  023376  2$:     BISB    LINE,(R1)       ;SELECT A LINE
004614  112761  000000  000016  MOVB    #0,SSR(R1)      ;SCOPE SYNC
004622  010412                  MOV     R4,(R2)         ;LOAD THE CAR REG.
```

```
1765  004624  062704  010421              ADD     #10421,R4       ;GENERATE NEW DATA
1766  004630  000760                      BR      1$              ;GO DO NEXT LINE
1767
1768  004632  004767  017644      3$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
1769  004636  000434                      BR      7$              ;;BR IF CHECKED ALL LINES
1770  004640  105767  023346              TSTB    LINE            ;DOING LINE 00 ?
1771  004644  001001                      BNE     4$              ;BR IF NOT
1772  004646  005004                      CLR     R4              ;INIT S/B DATA
1773
1774  004650  156711  023336      4$:     BISB    LINE,(R1)       ;SELECT A LINE
1775  004654  112761  000000  000017      MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
1776  004662  011203                      MOV     (R2),R3         ;GET CONTENTS OF CAR
1777  004664  020304                      CMP     R3,R4           ;WAS DATA OK ?
1778  004666  001412                      BEQ     5$              ;BR IF YES
1779
1780  004670  004767  017454              JSR     PC,SUER2        ;GO SET UP ERROR INFO
1781  004674  004567  017674              JSR     R5,SUNUM        ;SET UP LINE NO. IN MSG BUFFER
1782  004700  030212                      LINE
1783  004702  031045                      EM7+47
1784  004704  012767  004722  174176      MOV     #6$,$LPERR      ;SET UP ERROR LOOP RETURN
1785  004712  104007                      ERROR   7               ;CAR ADDRESSING ERROR
1786
1787  004714  062704  010421      5$:     ADD     #10421,R4       ;GENERATE NEW S/B DATA
1788  004720  000744                      BR      3$              ;GO CHECK NEXT LINE
1789
1790  004722  005067  023264      6$:     CLR     LINE            ;RESTART AT LINE 00 IF LOOPING
1791  004726  000721                      BR      1$              ;GO RESTART
1792
1793  004730  016767  174264  022244  7$: MOV     $TMP7,LINSEL    ;RESTORE LINE SELECT PARAMETER
```

```
                        ;;*****************************************************************
                        ;*TEST 15       "BCR" MEMORY ADDRESSING TEST
                        ;;*****************************************************************
004736  000004          TST15:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        ***************

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


                                MEMORY PATTERN: LOCATION        CONTENTS        (BOTH OCTAL)

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

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

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

                        ERRORS:
                        ********

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

                        SYNC:
                        ******

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

```
                                DEBUG:
                                ******

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

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

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

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

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

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

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

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

                                                SH5     MEMADD SOURCE SEL H     E55-8
                                                        BC MEM WRITE ENAB L     E57-4
                                                        BUF ADDRS TO BUS H      E33-1 (SHD BE LOW)

                                        M7278   SH3     BITS<15:08>
                                                        74157 INPUT MUX CHIPS E18,E19
                                                        74193 UP COUNTER CHIPS E27,E26
                                                        7489 MEMORY CHIPS E33,E34
                                                        7404 INVERTER CHIPS E41,E42

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

                                                SH5 THRU SH8    74151 DATA MUX OUTPUT CHIPS (PIN 1 INPUT)

                                %
004740  010102          MOV     R1,R2           ;COPY IT IN R2
004742  062702  000010  ADD     #BCR,R2         ;SET UP REGADR IN R2
004746  016767  022230  174244  MOV     LINSEL,$TMP7    ;SAVE LINE SELECT PARAMETER
004754  012767  177777  022220  MOV     #-1,LINSEL      ;DO ALL LINES FOR THIS TEST
004762  004767  017514    1$:   JSR     PC,SELINE       ;GO SELECT A LINE NO.
004766  000415          BR      3$              ;;BR IF DONE ALL SELECTED LINES
004770  105767  023216  TSTB    LINE            ;DOING LINE 00 ?
004774  001001          BNE     2$              ;BR IF NOT
004776  005004          CLR     R4              ;INIT TEST DATA

005000  156711  023206    2$:   BISB    LINE,(R1)       ;SELECT A LINE
005004  112761  000000  000016  MOVB    #0,SSR(R1)      ;SCOPE SYNC
```

```
 1906  005012  010412                        MOV     R4,(R2)         ;LOAD THE BCR REG.
 1907  005014  062704  010421                ADD     #10421,R4       ;GENERATE NEW DATA
 1908  005020  000760                        BR      1$              ;GO DO NEXT LINE
 1909
 1910  005022  004767  017454          3$:   JSR     PC,SELINE       ;GO SELECT A LINE NO.
 1911  005026  000434                        BR      7$              ;;BR IF CHECKED ALL LINES
 1912  005030  105767  023156                TSTB    LINE            ;DOING LINE 00 ?
 1913  005034  001001                        BNE     4$              ;BR IF NOT
 1914  005036  005004                        CLR     R4              ;INIT S/B DATA
 1915
 1916  005040  156711  023146          4$:   BISB    LINE,(R1)       ;SELECT A LINE
 1917  005044  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
 1918  005052  011203                        MOV     (R2),R3         ;GET CONTENTS OF BCR
 1919  005054  020304                        CMP     R3,R4           ;WAS DATA OK ?
 1920  005056  001412                        BEQ     5$              ;BR IF YES
 1921
 1922  005060  004767  017264                JSR     PC,SUER2        ;GO SET UP ERROR INFO
 1923  005064  004567  017504                JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
 1924  005070  030212                        LINE
 1925  005072  031114                        EM10+44
 1926  005074  012767  005112  174006        MOV     #6$,$LPERR      ;SET UP ERROR LOOP RETURN
 1927  005102  104010                        ERROR   10              ;BCR ADDRESSING ERROR
 1928
 1929  005104  062704  010421          5$:   ADD     #10421,R4       ;GENERATE NEW S/B DATA
 1930  005110  000744                        BR      3$              ;GO CHECK NEXT LINE
 1931
 1932  005112  005067  023074          6$:   CLR     LINE            ;RESTART AT LINE 00 IF LOOPING
 1933  005116  000721                        BR      1$              ;GO RESTART
 1934
 1935  005120  016767  174074  022054  7$:   MOV     $TMP7,LINSEL    ;RESTORE THE LINE SELECT PARAMETER
```

```
                                ;;*****************************************************************
                                ;*TEST 16       "CAR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
                                ;;*****************************************************************
005126  000004                  TST16:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

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

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

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

                                ERRORS:
                                ********

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

                                SYNC:
                                *****

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

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

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

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

                                %
005130  010102                          MOV     R1,R2           ;COPY IT INTO R2
005132  062702  000006                  ADD     #CAR,R2         ;R2 GETS CAR ADDRESS
005136  004767  017340          1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
005142  000443                          BR      TST17           ;;BR IF DONE ALL SELECTED LINES
005144  012704  177777                  MOV     #-1,R4          ;RESULT IN CAR S/B = 177777
005150  156711  023036                  BISB    LINE,(R1)       ;SELECT A LINE NO.
005154  004567  017414                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
005160  030212                          LINE
005162  031045                          EM7+47
                                
005164  112761  000000  000016  2$:     MOVB    #0,SSR(R1)      ;SCOPE SYNC
005172  010412                          MOV     R4,(R2)         ;LOAD A CAR WITH ALL ONES
005174  011203                          MOV     (R2),R3         ;GET THE WAS DATA FROM THE CAR
005176  020403                          CMP     R4,R3           ;DID IT CONTAIN ALL ONES ??
```

```
1992  005200  001406                          BEQ     3$              ;;BR IF ALL 1'S
1993
1994  005202  004767  017142                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
1995  005206  012767  005164  173674          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
1996  005214  104007                          ERROR   7               ;FAILED TO SET ALL 1'S IN SELECTED CAR
1997
1998  005216  005004                  3$:     CLR     R4              ;RESULT IN CAR S/B = 000000
1999  005220  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
2000  005226  010412                          MOV     R4,(R2)         ;CLEAR SELECTED CAR
2001  005230  011203                          MOV     (R2),R3         ;GET THE WAS DATA
2002  005232  001741                          BEQ     1$              ;BR IF CAR GOT CLEARED
2003
2004  005234  004767  017110                  JSR     PC,SUER2        ;GO SET UP FOR ERROR CALL
2005  005240  012767  005216  173642          MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN
2006  005246  104007                          ERROR   7               ;FAILED TO CLR ALL BITS IN SELECTED CAR
2007  005250  000732                          BR      1$              ;GO TEST NEXT LINE
```

```
                                ;;*****************************************************************
                                ;*TEST 17       "BCR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
                                ;;*****************************************************************
005252  000004                  TST17:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

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

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

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

                                ERRORS:
                                ********

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

                                SYNC:
                                *****

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

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

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

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

                                %
005254  010102                          MOV     R1,R2           ;COPY IT INTO R2
005256  062702  000010                  ADD     #BCR,R2         ;R2 GETS BCR ADDRESS
005262  004767  017214          1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
005266  000443                          BR      TST20           ;;BR IF DONE ALL SELECTED LINES
005270  012704  177777                  MOV     #-1,R4          ;RESULT IN BCR S/B = 177777
005274  156711  022712                  BISB    LINE,(R1)       ;SELECT A LINE NO.
005300  004567  017270                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
005304  030212                          LINE
005306  031114                          EM10+44

005310  112761  000000  000016  2$:     MOVB    #0,SSR(R1)      ;SCOPE SYNC
005316  010412                          MOV     R4,(R2)         ;LOAD A BCR WITH ALL ONES
005320  011203                          MOV     (R2),R3         ;GET THE WAS DATA FROM THE BCR
005322  020403                          CMP     R4,R3           ;DID IT CONTAIN ALL ONES ??
```

```
2064  005324  001406                          BEQ     3$              ;;BR IF ALL 1'S
2065
2066  005326  004767  017016                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
2067  005332  012767  005310  173550          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
2068  005340  104010                          ERROR   10              ;FAILED TO SET ALL 1'S IN SELECTED BCR
2069
2070  005342  005004                  3$:     CLR     R4              ;RESULT IN BCR S/B = 000000
2071  005344  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
2072  005352  010412                          MOV     R4,(R2)         ;CLEAR SELECTED BCR
2073  005354  011203                          MOV     (R2),R3         ;GET THE WAS DATA
2074  005356  001741                          BEQ     1$              ;BR IF BCR GOT CLEARED
2075
2076  005360  004767  016764                  JSR     PC,SUER2        ;GO SET UP FOR ERROR CALL
2077  005364  012767  005342  173516          MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN
2078  005372  104010                          ERROR   10              ;FAILED TO CLR ALL BITS IN SELECTED BCR
2079  005374  000732                          BR      1$              ;GO TEST NEXT LINE
```

```
                                ;;*****************************************************************
                                ;*TEST 20       "CAR" MEMORY PATTERNS TEST / 0'S DISTURB
                                ;;*****************************************************************
005376  000004                  TST20:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

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

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

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

                                ERRORS:
                                *******

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

                                SYNC*
                                ******

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

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

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

                                KEY LOGIC:      (REFER TO TEST 14)
                                **********
                                %
005400  010102                          MOV     R1,R2           ;SET UP REGADR
005402  062702  000006                  ADD     #CAR,R2
005406  012705  027220                  MOV     #PATRNA,R5      ;SET UP POINTER TO DATA PATTERNS
005412  012567  173566          1$:     MOV     (R5)+,$TMP1     ;GET A DATA TEST PATTERN
005416  001472                          BEQ     TST21           ;;BR IF DONE THREE PATTERNS
005420  004767  017056          11$:    JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
005424  000772                          BR      1$              ;BR IF DONE ALL SELECTED LINES
005426  116767  022560  022560        MOVB    LINE,LINEA      ;SAVE THE LINE NO. FOR ERROR LOOPING

005434  105067  173550          2$:     CLRB    $TMP3           ;INIT LINE COUNTER
```

```
2136  005440  116711  173544          3$:     MOVB    $TMP3,(R1)      ;SELECT A LINE TO CLEAR
2137  005444  005012                          CLR     (R2)            ;CLR CAR FOR THAT LINE
2138  005446  105267  173536                  INCB    $TMP3           ;GENERATE NEW LINE NO.
2139  005452  126727  173532  000020          CMPB    $TMP3,#20       ;DONE CLEARING ALL LINES ?
2140  005460  001367                          BNE     3$              ;BR IF NOT
2141
2142  005462  116711  022526                  MOVB    LINEA,(R1)      ;SET LINE SELECT BITS
2143  005466  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
2144  005474  016712  173504                  MOV     $TMP1,(R2)      ;LOAD CAR WITH TEST PATTERN
2145
2146  005500  105067  173502                  CLRB    $TMP2           ;INIT A LINE COUNTER
2147  005504  016704  173474          4$:     MOV     $TMP1,R4        ;SET UP S/B DATA
2148  005510  116711  173472                  MOVB    $TMP2,(R1)      ;SET LINE SELECT IN SCR
2149  005514  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
2150  005522  011203                          MOV     (R2),R3         ;GET WAS DATA
2151  005524  126767  173456  022460          CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
2152  005532  001401                          BEQ     5$              ;BR IF IT IS
2153  005534  005004                          CLR     R4              ;MAKE S/B DATA = 000000
2154  005536  020304          5$:     CMP     R3,R4           ;CORRECT DATA IN CAR ?
2155  005540  001412                          BEQ     6$              ;BR IF YES
2156
2157  005542  004767  016670                  JSR     PC,SUER4        ;GO SET UP ERROR IN F0
2158  005546  004567  017022                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
2159  005552  001206                          $TMP2
2160  005554  033666                          EM46+63
2161  005556  012767  005434  173324          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
2162  005564  104046                          ERROR   46              ;INCORRECT DATA READ FROM CAR
2163
2164  005566  105267  173414          6$:     INCB    $TMP2           ;GENERATE NEXT LINE NO.
2165  005572  122767  000020  173406          CMPB    #20,$TMP2       ;DONE ALL LINES ?
2166  005600  001707                          BEQ     11$             ;BR IF YES
2167  005602  000740                          BR      4$              ;GO CHECK NEXT LINE
```

```
                        ;;*****************************************************
                        ;*TEST 21       "BCR" MEMORY PATTERNS TEST / 0'S DISTURB
                        ;;*****************************************************
005604  000004          TST21:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

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

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

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

                        ERRORS:
                        *******

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

                        SYNC#
                        ******

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

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

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

                        KEY LOGIC:      (REFER TO TEST 15)
                        **********
                        %
005606  010102                          MOV     R1,R2           ;SET UP REGADR
005610  062702  000010                  ADD     #BCR,R2
005614  012705  027220                  MOV     #PATRNA,R5      ;SET UP POINTER TO DATA PATTERNS
005620  012567  173360          1$:     MOV     (R5)+,$TMP1     ;GET A DATA TEST PATTERN
005624  001472                          BEQ     TST22           ;;BR IF DONE THREE PATTERNS
005626  004767  016650          11$:    JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
005632  000772                          BR      1$              ;BR IF SELECTED ALL LINES
005634  116767  022352  022352          MOVB    LINE,LINEA      ;SAVE THE LINE NO. FOR ERROR LOOP

005642  105067  173342          2$:     CLRB    $TMP3           ;INIT LINE COUNTER
```

```
 2224  005646  116711  173336          3$:   MOVB   $TMP3,(R1)      ;SELECT A LINE TO CLEAR
 2225  005652  005012                        CLR    (R2)            ;CLR BCR FOR THAT LINE
 2226  005654  105267  173330                INCB   $TMP3           ;GENERATE NEW LINE NO.
 2227  005660  126727  173324  000020        CMPB   $TMP3,#20       ;DONE CLEARING ALL LINES ?
 2228  005666  001367                        BNE    3$              ;BR IF NOT
 2229
 2230  005670  116711  022320                MOVB   LINEA,(R1)      ;SET LINE SELECT BITS
 2231  005674  112761  000000  000016        MOVB   #0,SSR(R1)      ;SCOPE SYNC
 2232  005702  016712  173276                MOV    $TMP1,(R2)      ;LOAD BCR WITH TEST PATTERN
 2233
 2234  005706  105067  173274                CLRB   $TMP2           ;INIT A LINE COUNTER
 2235  005712  016704  173266          4$:   MOV    $TMP1,R4        ;SET UP S/B DATA
 2236  005716  116711  173264                MOVB   $TMP2,(R1)      ;SELECT A LINE TO CHECK
 2237  005722  112761  000000  000017        MOVB   #0,SSR+1(R1)    ;SCOPE SYNC
 2238  005730  011203                        MOV    (R2),R3         ;GET WAS DATA
 2239  005732  126767  173253  022252        CMPB   $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
 2240  005740  001401                        BEQ    5$              ;BR IF IT IS
 2241  005742  005004                        CLR    R4              ;MAKE S/B DATA = 000000
 2242  005744  020304          5$:   CMP    R3,R4           ;CORRECT DATA IN BCR ?
 2243  005746  001412                        BEQ    6$              ;BR IF YES
 2244
 2245  005750  004767  016462                JSR    PC,SUER4        ;GO SET UP ERROR IN F0
 2246  005754  004567  016614                JSR    R5,SUNUM        ;GO SET UP LINE NO.IN MSG BUFFER
 2247  005760  001206                        $TMP2
 2248  005762  034066                        EM47+56
 2249  005764  012767  005642  173116        MOV    #2$,$LPERP      ;SET UP ERROR LOOP RETURN
 2250  005772  104047                        ERROR  47              ;INCORRECT DATA READ FROM BCR
 2251
 2252  005774  105267  173206          6$:   INCB   $TMP2           ;GENERATE NEXT LINE NO.
 2253  006000  122767  000020  173200        CMPB   #20,$TMP2       ;DONE ALL LINES ?
 2254  006006  001707                        BEQ    11$             ;BR IF YES
 2255  006010  000740                        BR     4$              ;GO CHECK NEXT LINE
```

```
                                ;;*****************************************************************
                                ;*TEST 22       "CAR" MEMORY PATTERNS TEST / 1'S DISTURB
                                ;;*****************************************************************
006012  000004                  TST22:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

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

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

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

                                ERRORS:
                                *******

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

                                SYNC:
                                *****

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

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

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

                                KEY LOGIC:      (REFER TO TEST 14)
                                **********
                                %
006014  010102                          MOV     R1,R2           ;SET UP REGADR
006016  062702  000006                  ADD     #CAR,R2
006022  012705  177777                  MOV     #-1,R5          ;TEST PATERRN IN R5 = 177777
006026  010567  173152                  MOV     R5,$TMP1        ;SAVE FOR ERROR REPORTING
006032  004767  016444          1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
006036  000465                  TST23   BR      TST23           ;;BR IF DONE ALL LINES
006040  116767  022146  022146          MOVB    LINE,LINEA      ;SAVE THE LINE NO. FOR ERROR LOOP

006046  105067  173136          2$:     CLRB    $TMP3           ;INIT LINE COUNTER
006052  116711  173132          3$:     MOVB    $TMP3,(R1)      ;SELECT A LINE TO CLEAR
006056  010512                  TST24   MOV     R5,(R2)         ;LOAD CAR WITH 177777
006060  105267  173124                  INCB    $TMP3           ;GENERATE NEW LINE NO.
006064  126727  173120  000020          CMPB    $TMP3,#20       ;DONE SETTING ALL LINES TO 177777
```

```
2312  006072  001367                          BNE     3$              ;BR IF NOT
2313
2314  006074  116711  022114                  MOVB    LINEA,(R1)      ;SET LINE SELECT IN SCR
2315  006100  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
2316  006106  005012                          CLR     (R2)            ;CLEAR THE CAR UNDER TEST
2317
2318  006110  105067  173072                  CLRB    $TMP2           ;INIT A LINE COUNTER
2319  006114  005004                  4$:     CLR     R4              ;MAKE S/B DATA = 000000
2320  006116  116711  173064                  MOVB    $TMP2,(R1)      ;SELECT A LINE TO CHECK
2321  006122  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
2322  006130  011203                          MOV     (R2),R3         ;GET WAS DATA
2323  006132  126767  173050  022052          CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
2324  006140  001401                          BEQ     5$              ;BR IF IT IS
2325  006142  010504                          MOV     R5,R4           ;MAKE S/B DATA = 177777
2326  006144  020304                  5$:     CMP     R3,R4           ;CORRECT DATA IN CAR ?
2327  006146  001412                          BEQ     6$              ;BR IF YES
2328
2329  006150  004767  016262                  JSR     PC,SUER4        ;GO SET UP ERROR IN FO
2330  006154  004567  016414                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
2331  006160  001206                          $TMP2
2332  006162  033666                          EM46+63
2333  006164  012767  006046  172716          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
2334  006172  104046                          ERROR   46              ;INCORRECT DATA READ FROM CAR
2335
2336  006174  105267  173006          6$:     INCB    $TMP2           ;GENERATE NEXT LINE NO.
2337  006200  122767  000020  173000          CMPB    #20,$TMP2       ;DONE ALL LINES ?
2338  006206  001711                          BEQ     1$              ;BR IF YES
2339  006210  000741                          BR      4$              ;GO CHECK NEXT LINE
```

```
                                ;;**********************************************************
                                ;*TEST 23       "BCR" MEMORY PATTERNS TEST / 1'S DISTURB
                                ;;**********************************************************
006212  000004                  TST23:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

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

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

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

                                ERRORS:
                                *******

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

                                SYNC:
                                *****

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

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

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

                                KEY LOGIC:      (REFER TO TEST 15)
                                **********
                                %
006214  010102                          MOV     R1,R2           ;SET UP REGADR
006216  062702  000010                  ADD     #BCR,R2
006222  012705  177777                  MOV     #-1,R5          ;TEST PATERRN IN R5 = 177777
006226  010567  172752                  MOV     R5,$TMP1        ;SAVE IT FOR ERROR REPORTING
006232  004767  016244          1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
006236  000465                          BR      TST24           ;;BR IF DONE ALL LINES
006240  116767  021746  021746          MOVB    LINE,LINEA      ;SAVE THE LINE NO.

006246  105067  172736          2$:     CLRB    $TMP3           ;INIT LINE COUNTER
006252  116711  172732          3$:     MOVB    $TMP3,(R1)      ;SELECT A LINE TO INIT
006256  010512                          MOV     R5,(R2)         ;LOAD BCR WITH 177777
006260  105267  172724                  INCB    $TMP3           ;GENERATE NEW LINE NO.
006264  126727  172720  000020          CMPB    $TMP3,#20       ;DONE SETTING ALL LINES TO 177777 ?
```

```
2396  006272  001367                          BNE     3$              ;BR IF NOT
2397
2398  006274  116711  021714                  MOVB    LINEA,(R1)      ;SET LINE SELECT BITS
2399  006300  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
2400  006306  005012                          CLR     (R2)            ;CLEAR THE BCR UNDER TEST
2401
2402  006310  105067  172672                  CLRB    $TMP2           ;INIT A LINE COUNTER
2403  006314  005004                  4$:     CLR     R4              ;MAKE S/B DATA = 000000
2404  006316  116711  172664                  MOVB    $TMP2,(R1)      ;SELECT A LINE TO CHECK
2405  006322  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
2406  006330  011203                          MOV     (R2),R3         ;GET WAS DATA
2407  006332  126767  172650  021652          CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
2408  006340  001401                          BEQ     5$              ;BR IF IT IS
2409  006342  010504                          MOV     R5,R4           ;MAKE S/B DATA = 177777
2410  006344  020304                  5$:     CMP     R3,R4           ;CORRECT DATA IN BCR ?
2411  006346  001412                          BEQ     6$              ;BR IF YES
2412
2413  006350  004767  016062                  JSR     PC,SUER4        ;GO SET UP ERROR IN F0
2414  006354  004567  016214                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
2415  006360  001206                          $TMP2
2416  006362  034066                          EM47+56
2417  006364  012767  006246  172516          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
2418  006372  104047                          ERROR   47              ;INCORRECT DATA READ FROM BCR
2419
2420  006374  105267  172606          6$:     INCB    $TMP2           ;GENERATE NEXT LINE NO.
2421  006400  122767  000020  172600          CMPB    #20.,$TMP2      ;DONE ALL LINES ?
2422  006406  001711                          BEQ     1$              ;BR IF YES
2423  006410  000741                          BR      4$              ;GO CHECK NEXT LINE
```

```
                        ;;*****************************************************************
                        ;;*TEST 24       TEST THAT "CAR" MEMORY EXT BITS SET/CLR PROPERLY
                        ;;*****************************************************************
002427  006412  000004  TST24:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

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

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

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

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

                        ERRORS:
                        *******

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

                        SYNC:
                        ******

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

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

                        DEBUG:
                        ******

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

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

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

```
                                                        SSR06 H         CH1

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

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

                                        %
006414  010102                        MOV     R1,R2           ;SET UP REGADR
006416  062702  000016                ADD     #SSR,R2
006422  012705  027230                MOV     #PATRNB,R5      ;SET UP POINTER TO DATA PATTERNS
006426  012567  172552        1$:     MOV     (R5)+,$TMP1     ;GET THE PATTERNS
006432  012567  172550                MOV     (R5)+,$TMP2
006436  001505                        BEQ     TST25           ;;BR IF DONE ALL PATTERNS

006440  105067  172544        2$:     CLRB    $TMP3           ;INIT A LINE COUNTER
006444  142711  000017        3$:     BICB    #17,(R1)        ;INIT LINE SELECT BITS IN "SCR"
006450  156711  172534                BISB    $TMP3,(R1)      ;SELECT A LINE IN SCR
006454  142711  000060                BICB    #60,(R1)        ;SO WE CLEAR ALL THE MEM EXT BITS
006460  005061  000006                CLR     CAR(R1) ;CLEAR A CAR
006464  105267  172520                INCB    $TMP3           ;GENERATE NXT LINE NO.
006470  122767  000020  172512        CMPB    #20,$TMP3       ;CLEARED THE WHOLE THING ?
006476  001362                        BNE     3$              ;BR IF NOT

006500  004767  015776        4$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
006504  000750                        BR      1$              ;BR IF DONE ALL LINES
006506  156711  021500                BISB    LINE,(R1)       ;SET UP LINE SELECT BITS
006512  156711  172466                BISB    $TMP1,(R1)      ;SET UP MEM EXT BIT PATTERN
006516  012761  000000  000004        MOV     #0,LPR(R1)      ;SCOPE SYNC
006524  012761  000000  000006        MOV     #0,CAR(R1)      ;WRITE EXT BITS IN THIS LOCATION

006532  105067  172454                CLRB    $TMP4           ;INIT A LINE COUNTER
006536  016704  172444        5$:     MOV     $TMP2,R4        ;SET UP S/B DATA
006542  142711  000017                BICB    #17,(R1)        ;INIT SELECT BITS IN "SCR"
006546  156711  172440                BISB    $TMP4,(R1)      ;SET SELECT BITS IN SCR
006552  012761  000000  000014        MOV     #0,BKR(R1)      ;SCOPE SYNC
006560  016103  000006                MOV     CAR(R1),R3      ;READ THE SELECTED "CAR"
006564  011203                        MOV     (R2),R3         ;GET THE WAS DATA
006566  042703  177477                BIC     #177477,R3      ;CLEAR JUNK BITS
006572  126767  021414  172412        CMPB    LINE,$TMP4      ;LINE UNDER TEST ??
006600  001401                        BEQ     6$              ;BR IF YES
006602  005004                        CLR     R4              ;MAKE S/B DATA = 000000
006604  020304                6$:     CMP     R3,R4           ;WERE MEM EXT BITS CORRECT ?
006606  001412                        BEQ     7$              ;BR IF YES

006610  004767  015534                JSR     PC,SUER2        ;GO SET UP ERROR INFO
006614  004567  015754                JSR     R5,SUBJM        ;GO SET LINE NO. IN MSG BUFFER
006620  001212                        $TMP4
006622  031045                        EM7+47
006624  012767  006440  172256        MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
006632  104007                        ERROR   7               ;MEM EXT BITS READ INCORRECTLY

006634  105267  172352        7$:     INCB    $TMP4           ;GENERATE NXT LINE NO.
006640  122767  000020  172344        CMPB    #20,$TMP4       ;DONE ALL LINES
006646  001674                        BEQ     2$              ;BR IF YES
```

J14


```
 2536  006650  000732                        BR      5$              ;GO CHECk NEXT LINE
```

```
                                ;;*****************************************************************
                                ;*TEST 25       TEST INTR. ENAB. BITS - INTR. CONDITION DISABLED
                                ;;*****************************************************************
006652  000004                  TST25:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

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

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

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

                                ERRORS:
                                *******

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

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

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

                                DEBUG:
                                ******

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

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

                                        M7289   SH6     XMIT INT REQ H  FM1
                                                        RCV INT REQ H   DP1
                                %
006654  012767  006732  172226           MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
006662  010102                           MOV     R1,R2           ;MAKE IT REGADR TOO
006664  016703  020304                   MOV     DHVCT,R3        ;GET FIRST VECTOR ADDRESS
006670  012723  006762                   MOV     #4$,(R3)+       ;GO TO 3$ IF RCVR INTRS
006674  116723  021306                   MOVB    DHRLVL,(R3)+
006700  105723                           TSTB    (R3)+           ;UPDATE POINTER
006702  012723  007004                   MOV     #5$,(R3)+       ;GO TO 5$ IF XMITTR INTRS
```

```
2593  006706  116713  021275              MOVB    DHTLVL,(R3)
2594  006712  012705  000001              MOV     #1,R5           ;INIT BIT TEST MARKER
2595  006716  030567  020644      1$:     BIT     R5,INTMSK       ;TEST THIS BIT ??
2596  006722  001003                      BNE     3$              ;BR IF YES
2597  006724  006305              2$:     ASL     R5              ;SHIFT THE MARKER
2598  006726  001437                      BEQ     6$              ;BR IF TESTED ALL REQUIRED BITS
2599  006730  000772                      BR      1$              ;GO TEST FOR THIS ONE
2600
2601  006732  012706  001100      3$:     MOV     #STACK,SP       ;RESET SP FOR ERROR LOOPING
2602  006736  004767  020150              JSR     PC,CHPS2        ;GO LOCK OUT INTRS
2603  006742  012711  004000              MOV     #BIT11,(R1)     ;CLEAR THE DH11 INTERFACE
2604  006746  010504                      MOV     R5,R4           ;SET UP S/B DATA
2605  006750  050511                      BIS     R5,(R1)         ;SET THE TEST I.E. BIT
2606  006752  004767  020120              JSR     PC,CHPS1        ;GO CLEAR PSW
2607  006756  000240                      NOP                     ;WAIT A BIT TO ALLOW INTR
2608  006760  000761                      BR      2$              ;OK - GO DO NEXT I.E. BIT
2609
2610  006762  004767  020140      4$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
2611  006766  011103                      MOV     (R1),R3         ;GET THE WAS DATA
2612  006770  004767  015360              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
2613  006774  104011                      ERROR   11              ;DH11 RCVR SHOULD NOT HAVE INTERRUPTED
2614  006776  012716  006724              MOV     #2$,(SP)        ;SET UP TO RETURN
2615  007002  000002                      RTI                     ;RETURN TO TEST NEXT BIT
2616
2617  007004  004767  020116      5$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
2618  007010  011103                      MOV     (R1),R3         ;GET THE WAS DATA
2619  007012  004767  015336              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
2620  007016  104012                      ERROR   12              ;XMITTER SHOULD NOT HAVE INTERRUPTED
2621  007020  012716  006724              MOV     #2$,(SP)        ;SET UP TO RETURN
2622  007024  000002                      RTI                     ;RETURN TO TEST NEXT BIT
2623
2624  007026  012705  001100      6$:     MOV     #STACK,SP       ;RESET THE SP JUST IN CASE
2625  007032  004767  017660              JSR     PC,RES+RP       ;GO RESTORE TRAP CATCHER IN VECTOR
```

```
                        ;;*****************************************************************
                        ;*TEST 26       TEST CHAR. AVAIL. I.E.  WITH INTR. CONDITION ACTIVE
                        ;;*****************************************************************
007036  000004          TST26:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        ***************

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

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

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

                        ERRORS:
                        ********

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

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

                        DEBUG:
                        *******

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

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

                                M7289   SH6     E31-12

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

                                NOTE:   REMEMBER THAT PROBLEMS IN THIS AREA COULD BE CAUSED
                                        BY ANY DEVICE IN THE SYSTEM INCLUDING THE "CPU".
                                        SYSTEM RE-CONFIGURATION TO ISOLATE THE FAULTY SUB-
                                        SYSTEM MAY BE REQUIRED.
                        %
007040  012767  007076  172042        MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
007046  010102                        MOV     R1,R2           ;MAKE IT REGADR TOO
```

```
2682  007050  016703  020120              MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
2683  007054  012723  007206              MOV     #3$,(R3)+       ;GO TO 3$ IF RCVR INTRS
2684  007060  116723  021122              MOVB    DHRLVL,(R3)+
2685  007064  105723                       TSTB    (R3)+           ;UPDATE POINTER
2686  007066  012723  007162              MOV     #2$,(R3)+       ;GO TO 3$ ON XMITTR INTRS
2687  007072  116713  021111              MOVB    DHTLVL,(R3)
2688  007076  012711  004000      1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
2689  007102  012706  001100              MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
2690  007106  004767  020000              JSR     PC,CHPS2        ;GO LOCK OUT INTRS
2691  007112  012711  001000              MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
2692  007116  052711  000100              BIS     #BIT06,(R1)     ;SET CHAR AVAILABLE I.E. BIT
2693  007122  052711  000200              BIS     #BIT07,(R1)     ;SET THE CHAR AVAIL BIT TO FORCE INTR
2694  007126  004767  017744              JSR     PC,CHPS1        ;GO CLEAR PSW
2695  007132  000240                       NOP                     ;GIVE IT A LITTLE TIME
2696
2697  007134  004767  017766              JSR     PC,SAPS         ;SAVE THE ERROR PSW
2698  007140  011103                       MOV     (R1),R3         ;GET THE WAS DATA
2699  007142  005011                       CLR     (R1)            ;CLEAR OUT THE SCR
2700  007144  005011                       CLR     (R1)
2701  007146  012704  001300              MOV     #1300,R4        ;SET UP S/B DATA
2702  007152  004767  015176              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
2703  007156  104013                       ERROR   13              ;TIMED OUT AWAITING CHAR AVAIL INTR
2704  007160  000412                       BR      3$              ;GO EXIT TEST
2705
2706  007162  004767  017740      2$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
2707  007166  011103                       MOV     (R1),R3         ;GET WAS DATA
2708  007170  012704  001300              MOV     #1300,R4        ;SET UP S/B DATA
2709  007174  005011                       CLR     (R1)            ;CLR OUT SCR REG
2710  007176  005011                       CLR     (R1)
2711  007200  004767  015150              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
2712  007204  104012                       ERROR   12              ;UNEXPECTED XMITTR INTR
2713
2714  007206  012706  001100      3$:     MOV     #STACK,SP       ;RESET THE SP
2715  007212  004767  017500              JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER
```

```
                        ;;*****************************************************************
                        ;*TEST 27       TEST SILO OVFLW.  I.E.  WITH INTR. CONDITION ACTIVE
                        ;;*****************************************************************
007216  000004          TST27:  SCOPE
                        .REM    %
                        TEST ABSTRACT
                        *************

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

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

                        ERRORS:
                        *******

                        1.  [ERROR 43] IS CALLED TO REPORT NO RCVR INTR OCCURRED
                        2.  [ERROR 121] IS CALLED TO REPORT FALSE TRANSMITTER INTR.

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

                        DEBUG:
                        ******

                        1.  IF THE RECEIVER INTR FAILED TO INTERRUPT PROBLEM IS MOST LIKELY
                            THE M7289 MODULE
                        2.  IF A FALSE XMITTR INTR OCCURRED PROBLEM IS MOST LIKELY THE M7291 MODULE

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

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

                                M7821   "B" SECTION

                        %
007220  012767  007253  171662            MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
007226  010102                            MOV     R1,R2           ;MAKE IT REGADR TOO
007230  016703  017740                    MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
007234  012723  007366                    MOV     #3$,(R3)+       ;GO TO 3$ IF RCVR INTRS
007240  116723  020742                    MOVB    DHRLVL,(R3 +
007244  105723                            TSTB    (R3)+           ;UPDATE POINTER
007246  012723  007342                    MOV     #2$,(R3)+       ;GO TO 2$ ON XMITTR INTRS
007252  116713  020731                    MOVB    DHTLVL,(R3)
007256  012711  004000            1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
007262  012706  001100                    MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
007266  004767  017620                    JSR     PC,CHPS2        ;GO LOCK OUT INTRS
007272  012711  001000                    MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
```

```
                                                BIS     #BIT14,(R1)     ;SET SILO OVFLW I.E. BIT
                                                BIS     #BIT12,(R1)     ;SET THE SILO FULL BIT TO FORCE INTR
                                                JSR     PC,CHPS1        ;GO CLEAR PSW
                                                NOP                     ;GIVE IT A LITTLE TIME

                                                JSR     PC,SAPS         ;SAVE THE ERROR PSW
                                                MOV     (R1),R3         ;GET THE WAS DATA
                                                CLR     (R1)            ;CLEAR OUT THE SCR
                                                CLR     (R1)
                                                MOV     #51000,R4       ;SET UP S/B DATA
                                                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
                                                ERROR   43              ;TIMED OUT AWAITING SILO OVFLW INTR
                                                BR      3$              ;GO EXIT TEST

2$:                                             JSR     PC,SAPS         ;SAVE THE ERROR PSW
                                                MOV     (R1),R3         ;GET WAS DATA
                                                MOV     #51000,R4       ;SET UP S/B DATA
                                                CLR     (R1)            ;CLR OUT SCR REG
                                                CLR     (R1)
                                                JSR     PC,SLER2A       ;GO SET UP ERROR INFO
                                                ERROR   12              ;UNEXPECTED XMITTR INTR

3$:                                             MOV     #STACK,SP       ;RESET THE SP
                                                JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER
```

```
                        ;;; **********************************************************
                        ;TEST 30       TEST NON EX MEM I.E.  WITH INTR. CONDITION ACTIVE
                        ;;**********************************************************
                        ;
007376  000004          TST30:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

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

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

                        ERRORS:
                        *******

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

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

                        DEBUG:
                        ******

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

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

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

                        %
007400  012767  007436  171502            MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
007406  010102                            MOV     R1,R2           ;MAKE IT REGADR TOO
007410  016703  017560                    MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
007414  012723  007522                    MOV     #2$,(R3)+       ;GO TO 2$ IF RCVR INTRS
007420  116723  020562                    MOVB    DHRLVL,(R3)+
007424  105723                            TSTB    (R3)+           ;UPDATE POINTER
007426  012723  007546                    MOV     #3$,(R3)+       ;GO TO 3$ ON XMITTR INTRS
007432  116713  020551                    MOVB    DHTLVL,(R3)
007436  012711  004000            1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
007442  012706  001100                    MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
007446  004767  017440                    JSR     PC,CHPS2        ;GO LOCK OUT INTRS
007452  012711  001000                    MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
```

```
 2852  007456  052711  020000          BIS     #BIT13,(R1)     ;SET XMITTR I.E. BIT
 2853  007462  052711  002000          BIS     #BIT10,(R1)     ;SET THE NON EX MEM BIT TO FORCE INTR
 2854  007466  004767  017404          JSR     PC,CHPS1        ;GO CLEAR PSW
 2855  007472  000240                  NOP                     ;GIVE IT A LITTLE TIME
 2856
 2857  007474  004767  017426          JSR     PC,SAPS         ;SAVE THE ERROR PSW
 2858  007500  011103                  MOV     (R1),R3         ;GET THE WAS DATA
 2859  007502  005011                  CLR     (R1)            ;CLEAR OUT THE SCR
 2860  007504  005011                  CLR     (R1)
 2861  007506  012704  023000          MOV     #23000,R4       ;SET UP S/B DATA
 2862  007512  004767  014636          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 2863  007516  104044                  ERROR   44              ;TIMED OUT AWAITING NON EX MEM INTR
 2864  007520  000412                  BR      3$              ;GO EXIT TEST
 2865
 2866  007522  004767  017400  2$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
 2867  007526  011103                  MOV     (R1),R3         ;GET WAS DATA
 2868  007530  012704  023000          MOV     #23000,R4       ;SET UP S/B DATA
 2869  007534  005011                  CLR     (R1)            ;CLR OUT SCR REG
 2870  007536  005011                  CLR     (R1)
 2871  007540  004767  014610          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 2872  007544  104011                  ERROR   11              ;UNEXPECTED RCVR INTR
 2873
 2874  007546  012706  001100  3$:     MOV     #STACK,SP       ;RESET THE SP
 2875  007552  004767  017140          JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER
```

```
                                ;;*************************************************
                                ;*TEST 31       TEST XMITTR CONE I.E.  WITH INTR. CONDITION ACTIVE
                                ;;*************************************************
007556  000004                  TST31:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

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

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

                                ERRORS:
                                *******

                                1.  [ERROR 45] IS CALLED TO REPORT "XMIT DONE" INTR FAILED TO OCCUR
                                2.  [ERROR 111] IS CALLED TO REPORT FALSE RCVR INTRS

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

                                DEBUG:
                                ******

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


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

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

                                %
007560  012767  007616  171322          MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
007566  010102                          MOV     R1,R2           ;MAKE IT REGADR TOO
007570  016703  017400                  MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
007574  012723  007702                  MOV     #2$,(R3)+       ;GO TO 2$ IF RCVR INTRS
007600  116723  020402                  MOVB    DHRLVL,(R3)+
007604  105723                          TSTB    (R3)+           ;UPDATE POINTER
007606  012723  007726                  MOV     #3$,(R3)+       ;GO TO 3$ ON XMITTR INTRS
007612  116713  020371                  MOVB    DHTLVL,(R3)
007616  012711  004000          1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
007622  012706  001100                  MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
007626  004767  017260                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
007632  012711  001000                  MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
007636  052711  020000                  BIS     #BIT13,(R1)     ;SET XMIT DONE I.E. BIT
007642  052711  100000                  BIS     #BIT15,(R1)     ;SET THE XMITTR DONE BIT TO FORCE INTR
```

```
2932  007646  004767  017224              JSR     PC,CHPS1        ;GO CLEAR PSW
2933  007652  000240                      NOP                     ;GIVE IT A LITTLE TIME
2934
2935  007654  004767  017246              JSR     PC,SAPS         ;SAVE THE ERROR PSW
2936  007660  011103                      MOV     (R1),R3         ;GET THE WAS DATA
2937  007662  005011                      CLR     (R1)            ;CLEAR OUT THE SCR
2938  007664  005011                      CLR     (R1)
2939  007666  012704  121000              MOV     #121000,R4      ;SET UP S/B DATA
2940  007672  004767  014456              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
2941  007676  104045                      ERROR   45              ;TIMED OUT AWAITING XMIT DONE INTR
2942  007700  000412                      BR      3$              ;GO EXIT TEST
2943
2944  007702  004767  017220      2$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
2945  007706  011103                      MOV     (R1),R3         ;GET WAS DATA
2946  007710  012704  121000              MOV     #121000,R4      ;SET UP S/B DATA.
2947  007714  005011                      CLR     (R1)            ;CLR OUT SCR REG
2948  007716  005011                      CLR     (R1)
2949  007720  004767  014430              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
2950  007724  104011                      ERROR   11              ;UNEXPECTED RCVR INTR
2951
2952  007726  012706  001100      3$:     MOV     #STACK,SP       ;RESET THE SP
2953  007732  004767  016760              JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER
```

```
                                ;;**************************************************************
                                ;*TEST 32       BASIC TRANSMITTER "NPR" LOGIC TEST 1
                                ;;**************************************************************
007736  000004                  TST32:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

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

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

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

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

                                ERRORS:
                                *******

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

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

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

                                DEBUG:
                                ******

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

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


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

                    XMIT DONE FAILED TO SET:

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

                    "BAR" BIT FAILED TO CLEAR:

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

                    "CAR" REG NOT INCREMENTED:

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

                    "BCR" REG NOT INCREMENTED:

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

                    %
007740  012767  007774  171142         MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
007746  016703  017222                 MOV     DHVCT,R3        ;GET THE FIRST VECTOR ADDRESS
007752  062703  000004                 ADD     #4,R3           ;POINT TO XMITTR ENTRY
007756  012723  010140                 MOV     #4$,(R3)+       ;GO TO 4$ ON XMITTR INTR
007762  116713  020221                 MOVB    DHTLVL,(R3)
007766  004767  014510         1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO. TO TEST
007772  000552                 BR      8$                      ;BR IF TESTED ALL SELECTED LINES

007774  012706  001100         2$:     MOV     #STACK,SP       ;RESET SP FOR ERROR LOOPS
010000  010102                 MOV     R1,R2                   ;SET UP REGADR
010002  012704  120000                 MOV     #120000,R4      ;SET UP S/B DATA
010006  156704  020200                 BISB    LINE,R4
010012  004767  014636                 JSR     PC,CLCABC       ;GO CLEAR CAR AND BCR MEMORIES
010016  004767  017070                 JSR     PC,CHPS2        ;GO LOCK OUT INTRS
010022  012711  004000                 MOV     #BIT11,(R1)     ;CLEAR THE DH11 INTERFACE
010026  156711  020160                 BISB    LINE,(R1)       ;SELECT A LINE NO.
010032  012761  177777  000010         MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO -1
010040  012761  033500  000004         MOV     #33500,LPR(R1)  ;SET UP LINE PARAMETERS
010046  056761  017132  000012         BIS     LINMSK,BAR(R1)  ;ACTIVATE SELECTED LINE
```

```
3066  010054  052711  020000                BIS     #BIT13,(R1)     ;ENABLE INTERRUPT ON XMIT DONE
3067  010060  004767  017012                JSR     PC,CHPS1        ;GO CLEAR PSW
3068
3069  010064  012767  000001  020146        MOV     #1,TIMEA        ;INIT TIMER A
3070  010072  005067  020144                CLR     TIMEB           ;INIT TIMER B
3071  010076  000240                3$:     NOP                     ;DO NOTHING WAIT
3072  010100  004767  016640                JSR     PC,TIMEIT       ;CALL TIMER
3073  010104  000774                        BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC AROUND
3074                                                                ;THIS BRANCH IF TIMEOUT OCCURS
3075
3076  010106  004767  017014                JSR     PC,SAPS         ;SAVE THE ERROR PSW
3077  010112  011103                        MOV     (R1),R3         ;GET THE WAS DATA
3078  010114  042703  000200                BIC     #BIT07,R3       ;WE'RE NOT INTERESTED IN THIS BIT
3079  010120  004767  014230                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3080  010124  004567  014444                JSR     R5,SUNUM        ;GO SET LINE NO. IN ERROR MSG
3081  010130  030212                        LINE
3082  010132  031412                        EM15+43
3083  010134  104015                        ERROR   15              ;TIMEOUT WHILE AWAITING XMIT INTR
3084  010136  000713                        BR      1$              ;GO TEST NEXT LINE
3085
3086  010140  005711                4$:     TST     (R1)            ;DID XMIT DONE SET ??
3087  010142  100411                        BMI     5$              ;BR IF YES
3088
3089  010144  004767  016756                JSR     PC,SAPS         ;SAVE THE ERROR PSW
3090  010150  011103                        MOV     (R1),R3         ;GET THE WAS DATA
3091  010152  004767  014176                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3092  010156  004767  000174                JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
3093  010162  104014                        ERROR   14              ;XMIT DONE FAILED TO SET
3094  010164  000700                        BR      1$              ;GO TEST NEXT LINE
3095
3096  010166  016103  000012        5$:     MOV     BAR(R1),R3      ;GET WAS DATA FROM "BAR"
3097  010172  001413                        BEQ     6$              ;BR IF BAR BIT GOT CLEARED
3098
3099  010174  004767  016726                JSR     PC,SAPS         ;SAVE THE ERROR PSW
3100  010200  062702  000012                ADD     #BAR,R2         ;SET UP REGADR
3101  010204  005004                        CLR     R4              ;SET UP S/B DATA
3102  010206  004767  014142                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3103  010212  004767  000140                JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
3104  010216  104014                        ERROR   14              ;BAR BIT FAILED TO CLEAR
3105  010220  000662                        BR      1$              ;GO TEST NEXT LINE
3106
3107  010222  016103  000006        6$:     MOV     CAR(R1),R3      ;GET THE WAS DATA FROM CAR
3108  010226  022703  000001                CMP     #1,R3           ;DID IT GET INCREMENTED ?
3109  010232  001414                        BEQ     7$              ;BR IF YES
3110
3111  010234  004767  016666                JSR     PC,SAPS         ;SAVE THE ERROR PSW
3112  010240  012704  000001                MOV     #1,R4           ;SET UP S/B DATA
3113  010244  062702  000006                ADD     #CAR,R2         ;SET UP REGADR
3114  010250  004767  014100                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3115  010254  004767  000076                JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
3116  010260  104014                        ERROR   14              ;CAR REG NOT  INCREMENTED PROPERLY
3117  010262  000641                        BR      1$              ;GO TEST NEXT LINE.
3118
3119  010264  016103  000010        7$:     MOV     BCR(R1),R3      ;GET WAS DATA FROM BCR
3120  010270  001636                        BEQ     1$              ;BR IF BCR GOT INCREMENTED TO 000000
3121
```

```
3122  010272  004767  016630            JSR     PC,SAPS         ;SAVE THE ERROR PSW
3123  010276  005004                    CLR     R4              ;SET UP S/B DATA
3124  010300  062702  000010            ADD     #BCR,R2         ;SET UP REGADR
3125  010304  004767  014044            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3126  010310  004767  000042            JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
3127  010314  104014                    ERROR   14              ;BCR REG NOT INCREMENTED PROPERLY
3128  010316  000623                    BR      1$              ;GO TEST NEXT LINE
3129
3130  010320  012711  004000    9$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
3131  010324  016703  016644            MOV     DHVCT,R3        ;GET THE VECTOR ADDR
3132  010330  062703  000004            ADD     #4,R3           ;POINT TO XMIT VECTOR
3133  010334  010313                    MOV     R3,(R3)         ;RESTORE TRAP CATCHER
3134  010336  062723  000002            ADD     #2,(R3)+
3135  010342  005013                    CLR     (R3)
3136  010344  004767  016526            JSR     PC,CHPS1        ;GO CLEAR PSW
3137  010350  012706  001100            MOV     #STACK,SP       ;RESET THE STACK POINTER
3138  010354  000405                    BR      TST33           ;;GO TO NEXT TEST
3139
3140  010356  004567  014212    9$:     JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG.
3141  010362  030212                    LINE
3142  010364  031344                    EM14+44
3143  010366  000207                    RTS     PC              ;RETURN TO REPORT ERROR
```

```
                                ;;*****************************************************************
                                ;*TEST 33       TRANSMITTR NPR LOGIC TEST 2
                                ;;*****************************************************************
010370  000004                  TST33:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

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

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

                                ERRORS:
                                *******

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

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

                                DEBUG:
                                ******

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

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

                                %
010372  012767  010406  170510          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
010400  004767  014076          1$:     JSR     PC,$ELINE       ;GO SELECT A LINE TO TEST
010404  000544                          BR      TST34           ;;BR IF DONE ALL SELECTED LINES
010406  052711  004000          2$:     BIS     #BIT11,(R1)     ;CLEAR THE DH11
010412  004767  014236                  JSR     PC,CLCABC       ;GO CLEAR "CAR" AND "BCR" MEMORIES
010416  004767  014274                  JSR     PC,LDBCR        ;GO LOAD "BCR" MEMORY WITH ALL ONES
```

```
3200  010422  156711  017564                    BISB    LINE,(R1)          ;SELECT THE LINE
3201  010426  012761  033500  000004            MOV     #33500,LPR(R1)     ;SET UP PARAMETERS
3202  010434  016761  016544  000012            MOV     LINMSK,BAR(R1)     ;ACTIVATE XMIT ON SELECTED LINE
3203
3204  010442  012767  000001  017570            MOV     #1,TIMEA           ;INIT TIMER A
3205  010450  005067  017566                    CLR     TIMEB              ;INIT TIMER B
3206  010454  005711                    3$:     TST     (R1)               ;XMITTR DONE YET
3207  010456  100423                            BMI     4$                 ;BR IF YES
3208  010460  004767  016260                    JSR     PC,TIMEIT          ;CALL THE TIMER
3209  010464  000773                            BR      3$                 ;TIMER ROUTINE WILL MOVE RETURN PC
3210                                                                       ;AROUND THIS BRANCH IF TIME OUT OCCURS
3211
3212  010466  004767  016434                    JSR     PC,SAPS            ;SAVE THE ERROR PSW
3213  010472  011103                            MOV     (R1),R3            ;GET THE WAS DATA
3214  010474  012704  100000                    MOV     #BIT15,R4          ;SET UP S/B DATA
3215  010500  156704  017506                    BISB    LINE,R4
3216  010504  010102                            MOV     R1,R2              ;MAKE REGADR = DEVADR
3217  010506  004767  013642                    JSR     PC,SUER2A          ;GO SET UP ERROR INFO
3218  010512  004567  014056                    JSR     R5,SUNUM           ;SET LINE NO. IN MSG
3219  010516  030212                            LINE
3220  010520  034144                            EMS0+53
3221  010522  104050                            ERROR   50                 ;TIMED OUT AWAITING XMIT DONE ON SEL LINE
3222  010524  000725                            BR      1$                 ;GO TRY THE NEXT LINE
3223
3224  010526  005067  170466            4$:     CLR     $TMP7              ;INIT A LINE COUNTER
3225  010532  116711  170462            5$:     MOVB    $TMP7,(R1)         ;SELECT LINE NO. IN "SCR"
3226  010536  012704  177777                    MOV     #-1,R4             ;SET UP S/B DATA
3227  010542  016103  000010                    MOV     BCR(R1),R3         ;GET THE WAS BYTE COUNT
3228  010546  126767  017440  170444            CMPB    LINE,$TMP7         ;WAS THIS THE ACTIVE LINE ??
3229  010554  001001                            BNE     6$                 ;BR IF NOT
3230  010556  005004                            CLR     R4                 ;CHANGE S/B DATA TO 000000
3231  010560  020304                    6$:     CMP     R3,R4              ;WAS BYTE COUNT CORRECT ??
3232  010562  001416                            BEQ     7$                 ;BR IF YES
3233
3234  010564  005067  170412                    CLR     $TMP0              ;SAVE THE ACTIVE LINE NO.
3235  010570  116767  017416  170404            MOVB    LINE,$TMP0
3236  010576  016767  170416  170400            MOV     $TMP7,$TMP1        ;SAVE THE LINE NO. BEING CHECKED
3237  010604  010102                            MOV     R1,R2              ;SET UP REGADR = BCR REG ADDR
3238  010606  062702  000010                    ADD     #BCR,R2
3239  010612  004767  013620                    JSR     PC,SUER4           ;GO SET UP ERROR INFO
3240  010616  104051                            ERROR   51                 ;BYTE COUNT INCORRECT
3241
3242  010620  005004                    7$:     CLR     R4                 ;SET UP S/B DATA
3243  010622  016103  000006                    MOV     CAR(R1),R3         ;GET THE WAS DATA
3244  010626  126767  017360  170364            CMPB    LINE,$TMP7         ;IS THIS THE ACTIVE LINE
3245  010634  001001                            BNE     8$                 ;BR IF NOT
3246  010636  005204                            INC     R4                 ;BUMP THE CAR ADDRESS FOR ACTIVE LINE
3247  010640  020304                    8$:     CMP     R3,R4              ;CAR CONTENTS CORRECT ??
3248  010642  001416                            BEQ     9$                 ;BR IF YES
3249
3250  010644  005067  170332                    CLR     $TMP0              ;SET UP ACT LINE NO.
3251  010650  116767  017336  170324            MOVB    LINE,$TMP0
3252  010656  016767  170336  170320            MOV     $TMP7,$TMP1        ;SAVE THE LINE NO. BEING CHECKED
3253  010664  010102                            MOV     R1,R2              ;SET UP REGADR
3254  010666  062702  000006                    ADD     #CAR,R2
3255  010672  004767  013540                    JSR     PC,SUER4           ;SET UP THE ERROR INFO
```

```
3256  010676  104051                          ERROR   51              ;CAR REG INCORRECT
3257
3258  010700  005267  170314          9$:     INC     $TMP7           ;GENERATE NEW LINE NO.
3259  010704  022767  000020  170306          CMP     #20,$TMP7       ;TESTED ALL LINES
3260  010712  001307                          BNE     5$              ;BR IF NOT
3261  010714  000631                          BR      1$              ;GO SELECT NEXT ACTIVE LINE
```

```
                        ;;*************************************************************
                        ;*TEST 34       TEST THAT CHARACTER AVAILABLE CAN CAUSE RCVR INTERRUPT
                        ;;*************************************************************
010716  000034          TST34:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

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

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

                        ERRORS:
                        *******

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

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

                        DEBUG:
                        ******

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

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

                                M7279   SH1     SILO DATA MUX'S (74157'S)
                                                SSR15 H         CR1

                                        SH2     DATA READY L            DV1
                                                NPR 15 H                DL1
                                                SILO MEMORY             (E13,E17,E8,E3) (3341'S)
                                                LOAD SILO L             DJ2
                                                5.068 MHZ (CLOCK)       DN1
                                                SSR (13:00) H           (E20,E24)

                                M7289   SH6     RCV INT REQ H           DP1

                                M7278   SH8     SSR (03:00) H
```

```
                                                SH7     SSR <07:00> 4


010720  012767  010742  170162          MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
010726  016703  016242                  MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
010732  012723  011026                  MOV     #3$,(R3)+       ;GO TO 3$ ON RCVR INTERRUPT
010736  116713  017244                  MOVB    DHRLVL,(R3)
010742  004767  016144          1$:     JSR     PC,CHPS2        ;GO LOCK OUT INTRS
010746  012706  001100                  MOV     #STACK,SP       ;RESET SP FOR ERROR LOOPS
010752  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE DH11
010756  052761  100000  000016          BIS     #BIT15,SSR(R1)  ;SET SILO MAINT. BIT TO LOAD SILO
010764  012711  000100                  MOV     #BIT06,(R1)     ;ENABLE CHAR. AVAIL INTERRUPT
010770  004767  016102                  JSR     PC,CHPS1        ;GO CLEAR PSW
010774  012703  001000                  MOV     #1000,R3        ;INIT TIMER
011000  005303                  2$:     DEC     R3              ;DEC TIMER
011002  001376                  BNE     2$                      ;BR IF NO TIMEOUT

011004  004767  016116                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
011010  011103                  MOV     (R1),R3                 ;GET THE WAS DATA
011012  012704  000300                  MOV     #300,R4         ;SET UP S/B DATA
011016  004767  013332                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
011022  104013                  ERROR   13                      ;CHAR AVAIL FAILED TO SET ON TIME
011024  000436                  BR      5$                      ;ESCAPE FROM THIS TEST - CATASTROPHIC ERROR

011026  016105  000016          3$:     MOV     SSR(R1),R5      ;SAVE THE SILO STATUS REG.
011032  016103  000002                  MOV     NRC(R1),R3      ;GET THE WAS DATA
011036  012704  125252                  MOV     #125252,R4      ;SET UP S/B DATA
011042  020304                  CMP     R3,R4                   ;WAS = S/B = 125252 ??
011044  001410                  BEQ     4$                      ;BR IF IT IS

011046  004767  016054                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
011052  062702  000002                  ADD     #NRC,R2         ;SET UP REGADR
011056  004767  013272                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
011062  104052                  ERROR   52                      ;DATA COMPARE ERROR
011064  000416                  BR      5$                      ;GET OUT

011066  010503          4$:     MOV     R5,R3                   ;NOW GET THW SILO STATUS REG AGAIN
011070  042703  140377                  BIC     #140377,R3      ;CLR OUT JUNK
011074  012704  000400                  MOV     #400,R4         ;SET UP S/B DATA
011100  020304                  CMP     R3,R4                   ;SSR CHAR COUNT = 1 ??
011102  001407                  BEQ     5$                      ;BR IF IT IS

011104  004767  016016                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
011110  062702  000016                  ADD     #SSR,R2         ;SET UP REGADR
011114  004767  013234                  JSR     PC,SUER2A       ;SET UP ERROR INFO
011120  104006                  ERROR   6                       ;SSR COUNT NOT CORRECT

011122  012706  001100          5$:     MOV     #STACK,SP       ;RESET THE STACK POINTER
011126  004767  015744                  JSR     PC,CHPS1        ;GO CLEAR PSW
011132  005011                  CLR     (R1)                    ;RESET I.E. BIT
011134  016703  016034                  MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
011140  010313                  MOV     R3,(R3)                 ;RESTORE TRAP CATCHER
011142  062723  000002                  ADD     #2,(R3)+
011146  005013                  CLR     (R3)
```

```
                                ;;*****************************************************************
                                ;*TEST 35       TEST THAT THE SILO STATUS REG COUNTS UP CORRECTLY
                                ;;*****************************************************************
011150  000034                  TST35:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT THE SILO FILL LEVEL COUNTS UP CORRECTLY
                                WHEN ALL COUNTS (0-77) ARE TESTED BY LOADING THE SILO USING
                                MAINT MODE. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1.  INIT "$TMP7" TO START WITH A COUNT=01
                                        2.  CLEAR THE DH11
                                        3.  LOAD THE SILO WITH 125252'S IN MAINT MODE
                                            UNTIL # OF WORDS INDICATED BY THE COUNT ARE LOADED
                                        4.  AFTER LOADING REQUIRED COUNT CHECK THAT FILL LEVEL
                                            BITS (SSR<13:08>) EQUAL COUNT - IF NOT REPORT ERROR
                                        5.  INCREMENT COUNT IN "$TMP7" AND REPEAT 1 THRU 4
                                            UNTIL ALL COUNTS (01-77) HAVE BEEN TESED

                                ERRORS:
                                *******

                                1.  [ERROR 6] IS CALLED TO REPORT SILO FILL LEVEL ERRORS

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

                                DEBUG:
                                ******

                                1.  FAILURES IN THIS TEST MOST LIKELY INDICATE A BAD M7279 MODULE

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

                                        M7279   SH2     SSR <13:08>
                                                        LOAD SILO L DJ2
                                                        5.068 MHZ (CLOCK) DN1
                                                        DATA READY L DV1

                                %
011152  010102                          MOV     R1,R2           ;MAKE REGADR = SSR
011154  062702  000016                  ADD     #SSR,R2
011160  012767  000001  170032          MOV     #1,$TMP7        ;START WITH COUNT OF 1
011166  012711  004000          1$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
011172  016705  170022                  MOV     $TMP7,R5        ;SAVE CHARACTER COUNT BEING TESTED
011176  005004                          CLR     R4              ;INIT A CHAR COUNTER
011200  052712  100000          2$:     BIS     #BIT15,(R2)     ;SET THE SILO MAINT BIT
011204  012703  001000                  MOV     #1000,R3        ;INIT A TIMER
011210  005303          3$:     DEC     R3              ;STALL TO ALLOW TIME TO LOAD SILO
011212  001376                  BNE     3$
011214  042712  100000                  BIC     #BIT15,(R2)     ;CLEAR SILO MAINT. BIT
011220  005204                          INC     R4              ;COUNT A CHAR LOADED
011222  005305                          DEC     R5              ;DECREMENT TEST COUNT
```

```
3430  011224  001365                          BNE     2$              ;BR UNTIL WE'VE LOADED THE TEST COUNT
3431
3432  011226  011203                          MOV     (R2),R3         ;SET THE WAS COUNT
3433  011230  042703  140377                  BIC     #140377,R3      ;CLR JUNK BITS
3434  011234  000304                          SWAB    R4              ;SET UP S/B DATA
3435  011236  020304                          CMP     R3,R4           ;TEST COUNT = SILO COUNTER ?
3436  011240  001406                          BEQ     4$              ;BR IF YES
3437
3438  011242  004767  013102          JSR     PC,SUER2                ;GO SET UP ERROR INFO
3439  011246  012767  011166  167634          MOV     #1$,$LPERR      ;SET UP ERROR LOOP RETURN
3440  011254  104006                          ERROR   6               ;SSR FAILED TO UP-COUNT CORRECTLY
3441
3442  011256  005267  167736          4$:     INC     $TMP7           ;INCREMENT TO NEXT COUNT TO TEST
3443  011262  022767  000100  167730          CMP     #100,$TMP7      ;MAXIMUM COUNT ??
3444  011270  001336                          BNE     1$              ;BR IF NOT
3445
```

```
                        ;;*****************************************************************
                        ;*TEST 36       TEST THAT SILO STATUS REGISTER DOWN COUNTS CORRECTLY
                        ;;*****************************************************************
011272  000004          TST36:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

                                THIS TEST VERIFIES THAT THE SILO FILL LEVEL COUNTS DOWN PROPERLY
                        WHEN WORDS ARE READ FROM THE SILO. ALL COUNTS FROM 77-00 ARE
                        TESTED. THE TEST SEQUENCE IS AS FOLLOWS:

                                1.  INIT "$TMP7" TO START WITH A COUNT OF 1
                                2.  CLEAR THE DH11 AND FILL SILO WITH 64. WORDS
                                3.  READ THE NO. OF WORDS SPECIFIED BY COUNT
                                4.  CHECK THAT FILL LEVEL=64. MINUS [COUNT] - REPORT ERRORS
                                5.  INCREMENT "$TMP7" AND REPEAT 2 THRU 4 UNTIL ALL
                                    COUNTS TESTED.

                        ERRORS:
                        *******

                        1.  [ERROR 6] IS CALLED TO REPORT SILO FILL LEVEL ERRORS

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

                        DEBUG:          (REFER TO TEST 35)
                        ******

                        KEY LOGIC:      (REFER TO TEST 35)
                        **********

                        %
011274  010102                          MOV     R1,R2           ;SET UP REGADR
011276  062702  000016                  ADD     #SSR,R2
011302  012767  000001  167710          MOV     #1,$TMP7        ;START WITH COUNT = 1
011310  012711  004000          1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
011314  012705  000100                  MOV     #100,R5         ;TEST COUNT SHOULD BE 64(10) MINUS
011320  166705  167674                  SUB     $TMP7,R5        ;THE NO. OF CHARS READ
011324  012703  000100                  MOV     #100,R3         ;COUNTER USED TO FILL SILO
011330  012704  001000          2$:     MOV     #1000,R4        ;INIT TIMER
011334  052712  100000                  BIS     #BIT15,(R2)     ;SET SILO MAINT. BIT
011340  005304          3$:     DEC     R4              ;STALL TO ALLOW SILO TO LOAD
011342  001376                  BNE     3$
011344  042712  100000                  BIC     #BIT15,(R2)     ;CLEAR THE SILO MAINT BIT
011350  005303                  DEC     R3              ;COUNT ONE CHAR LOADED
011352  001366                  BNE     2$              ;BR UNTIL ALL LOADED

011354  016703  167640                  MOV     $TMP7,R3        ;INIT COUNTER FOR READING SILO
011360  012704  001000          4$:     MOV     #1000,R4        ;INIT TIMER
011364  005761  000002                  TST     NRC(R1)         ;READ THE SILO
011370  005304          5$:     DEC     R4              ;GIVE IT TIME TO SETTLE
011372  001376                  BNE     5$
011374  005303                  DEC     R3              ;COUNT ONE READ
011376  001370                  BNE     4$              ;BR UNTIL WE'VE READ TEST COUNT
```

```
3502
3503  011400  011203                        MOV     (R2),R3         ;GET THE WAS DATA
3504  011402  042703  140377                BIC     #140377,R3      ;CLR JUNK BITS
3505  011406  010504                        MOV     R5,R4           ;SET UP S/B DATA
3506  011410  000304                        SWAB    R4
3507  011412  020304                        CMP     R3,R4           ;DID IT DOWN COUNT OK ??
3508  011414  001406                        BEQ     6$              ;BR IF YES
3509
3510  011416  004767  012726                JSR     PC,SUER2        ;GO SET UP ERROR INFO
3511  011422  012767  011310  167460        MOV     #1$,$LPERR      ;SET UP ERROR LOOP RETURN
3512  011430  104006                        ERROR   6               ;SILO STATUS REG. DOWN-COUNTED INCORRECTLY
3513
3514  011432  005267  167562          6$:   INC     $TMP7           ;UPDATE COUNT
3515  011436  022767  000101  167554        CMP     #101,$TMP7      ;TESTED ALL COUNTS ??
3516  011444  001321                        BNE     1$              ;BR IF NOT
3517
3518
```

```
                                ;;**************************************************************
                                ;*TEST 37       TEST SILO ALARM LEVEL FOR  COUNTS 0,1,2,4,8,16, AND 32
                                ;;**************************************************************
011446  000004                  T37:  SCOPE
                                .REM  %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT THE SILO ALARM LEVEL WORKS PROPERLY FOR
                                INTEGRAL POWER OF 2 COUNTS (0, 1, 2, 4, 8, 16, AND 32). THE TEST SEQUENCE
                                IS AS FOLLOWS:

                                        1.  INIT "$TMP7" TO START WITH ALARM LEVEL OF 000
                                        2.  CLEAR THE DH11 AND LOAD THEN SILO WITH THAT NO. OF WORDS
                                            THAT IS ONE GREATER THAN THE ALARM LEVEL.
                                        3.  VERIFY THAT "DATA READY" DOES NOT SET UNTIL THE FILL
                                            LEVEL EXCEEDS THE ALARM LEVEL.
                                        4.  REPORT ERRORS IF:
                                            A.  "READY" SETS TOO SOON
                                            B.  "READY" SETS TOO LATE
                                        5.  SHIFT "$TMP7" LEFT TO GENERATE NEXT POWER OF 2 LEVEL
                                        6.  REPEAT 2 THRU 5 UNTIL ALL 7 TEST LEVELS CHECKED

                                        NOTE:   FOR (A) ABOVE IF "READY" SETS JUST ONE WORK TOO
                                                SOON IT IS ALLOWED BY ANYTHING GREATER RESULTS IN
                                                AN ERROR MESSAGE.

                                ERRORS:
                                *******

                                1.  [ERROR 6] IS CALLED TO REPORT BOTH TYPES OF ERRORS OUTLINED
                                    IN 4(A,B) ABOVE

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

                                DEBUG:
                                ******

                                1.  ERRORS IN THIS TEST ONLY INDICATE BAD COMPARATOR CHIP (E23 OR E19)
                                    ON THE M7279 - SH2

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

                                        M7279   SH2     E19 - PIN 5 (COMPARATOR)
                                        ALSO SAME LOGIC AS TEST 35

                                %
011450  012767  011470  167432          MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
011456  010102                          MOV     R1,R2           ;SET UP REGADR
011460  062702  000016                  ADD     #SSR,R2
011464  005067  167530                  CLR     $TMP7           ;START WITH LEVEL 00
011470  012711  004000          1$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
011474  016705  167520                  MOV     $TMP7,R5        ;SAVE IT IN R5
011500  010512                          MOV     R5,(R2)         ;SET ALARM LEVEL IN SSR
```

```
3575  011502  005205                          INC     R5              ;LOAD ONE MORE THAN FILL LEVEL
3576  011504  052712  100000          2$:     BIS     #BIT15,(R2)     ;SET SILO MAINT. TO LOAD A CHAR
3577  011510  012703  001000                  MOV     #1000,R3        ;INIT STALL TIMER
3578  011514  005303                  3$:     DEC     R3              ;WAIT FOR SILO TO SETTLE
3579  011516  001376                          BNE     3$              ;BR TIL R3 GOES TO 000000
3580  011520  042712  100000                  BIC     #BIT15,(R2)     ;CLR THE SILO MAINT BIT
3581  011524  005305                          DEC     R5              ;COUNT ONE LOADED
3582  011526  105711                          TSTB    (R1)            ;CHAR AVAIL SET YET
3583  011530  100412                          BMI     4$              ;BR IF IT IS
3584  011532  005705                          TST     R5              ;SHOULD IT BE ??
3585  011534  001363                          BNE     2$              ;BR IF NOT
3586
3587  011536  004767  015364                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
3588  011542  004767  000042                  JSR     PC,5$           ;GO SET UP S/B DATA
3589  011546  004767  012602                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3590  011552  104006                          ERROR   6               ;SILO ALARM LEVEL FAILED AT SELECTED COUNT
3591  011554  000426                          BR      6$              ;GO CHECK NEXT COUNT
3592
3593  011556  005705                  4$:     TST     R5              ;SHOULD IT HAVE BEEN SET (CHAR AVAIL)
3594  011560  001424                          BEQ     6$              ;BR IF YES
3595
3596  011562  004767  015340                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
3597  011566  022705  000001                  CMP     #1,R5           ;IS IT OFF BY ONLY ONE ??
3598  011572  001417                          BEQ     6$              ;BR IF YES - WE'LL ALLOW HIM THIS
3599  011574  004767  000010                  JSR     PC,5$           ;GO SET UP S/B DATA
3600  011600  004767  012550                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3601  011604  104006                          ERROR   6               ;SILO ALARM LEVEL FAILED
3602  011606  000411                          BR      6$              ;GO CHECK NEXT COUNT
3603
3604  011610  011203                  5$:     MOV     (R2),R3         ;GET WAS DATA
3605  011612  016704  167402                  MOV     $TMP7,R4        ;SET UP THE S/B DATA
3606  011616  005204                          INC     R4
3607  011620  000304                          SWAB    R4
3608  011622  105004                          CLRB    R4
3609  011624  156704  167370                  BISB    $TMP7,R4
3610  011630  000207                          RTS     PC              ;RETURN TO SET UP AND REPORT ERROR
3611
3612  011632  005767  167362          6$:     TST     $TMP7           ;COUNT AT ZERO
3613  011636  001002                          BNE     7$              ;BR IF NOT
3614  011640  000261                          SEC                     ;SET THE "C" BIT
3615  011642  000401                          BR      8$              ;GO SET UP COUNT
3616  011644  000241                  7$:     CLC                     ;CLEAR THE "C" BIT
3617  011646  006167  167346          8$:     ROL     $TMP7           ;SHIFT POWER OF TWO BIT
3618  011652  032767  000100  167340          BIT     #BIT6,$TMP7     ;DONE ALL POWERS ??
3619  011660  001703                          BEQ     1$              ;BR IF NOT
```

```
                                ;;*****************************************************************
                                ;*TEST 40       TRANSMITTER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS
                                ;;*****************************************************************
011662  000004                  TST40:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST PERFORMA A "RELATIVE" TIMING TEST FOR ALL BAUD RATES
                                ON ALL SELECTED LINES. IT DOES NOT MEASURE ABSOLUTE TIMES BUT
                                SIMPLY VERIFIES THAT EACH SUCCESSIVE SPEED FROM 50 TO 9600
                                BAUD IS FASTER THAN THE PREVIOUS SPEED. THE TEST SEQUENCE IS AS
                                FOLLOWS:

                                        1.  SELECT A LINE # TO TEST (AS DEFINED BY "LINSEL:")
                                        2.  INIT "$TMP7" TO START WITH 50 BAUD AND A RELATIVE
                                            TIMER "TIMEC" TO -1 (177777)
                                        3.  CLEAR THE DH11 AND ACTIVATE SELECTED LINE TO TRANSMIT
                                            THREE CHARS.
                                        4.  ACTIVATE TIMER TO UPDATE "TIMEB" THE LINE SPEED TIMER
                                        5.  IF "XMIT DONE" FAILS TO SET ON TIME - REPORT ERROR
                                            AND REPEAT 3 THRU 4 UNTIL ALL SPEEDS CHECKED - THEN
                                            REPEAT 1 THRU 5 UNTIL ALL LINES CHECKED
                                        6.  IF "XMIT DONE" SETS VERIFY [TIMEB] LESS THAN [TIMEC]
                                            IF NOT REPORT ERROR - MAKE [TIMEC]=[TIMEB] AND
                                            REPEAT 3 THRU 5 UNTIL ALL SPEEDS CHECKED
                                        7.  REPEAT 1 THRU 6 FOR ALL SELECTED LINES.

                                ERRORS:
                                *******

                                1.  [ERROR 53] IS CALLED TO REPORT XMIT TIMEOUT ERRORS
                                2.  [ERROR 171] IS CALLED TO REPORT TIMING ERRORS

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

                                DEBUG:
                                ******

                                1.  IF ALL LINES FAIL ON ALL SPEEDS SUSPECT THE CLOCK MODULE M4540
                                2.  IF ALL LINES FAIL ON JUST ONE SPEED (THE SAME ONE) SUPECT EITHER THE
                                    CLOCK MODULE OR THE M7288 MODULE (TIMING SELECT MUXES)
                                3.  IF JUST ONE LINE FAILS SUSPECT EITHER THE UART MODULE (M7280)
                                    EITHER FOR LINES <15:08> OR <07:00> OR THE M7288 MODULE

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

                                        M4540   SH2             <9600:50> BAUD SIGNALS
                                        M7288   SH3             BOT AND TOP BUF CLOCK SIGNALS
                                                SH4,6,8,OR 10   TX CLOCK NN L SIGNALS
                                        M7280   TBMT LINE "N" SIGNALS ON UART PIN 22
                                                TX CLOCK LINE "N" SIGNALS ON UART PIN 40

                                .
```

```
3676  011664  012767  011714  167216          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
3677  011672  004767  012604          1$:      JSR     PC,$ELINE       ;GO SELECT A LINE TO TEST
3678  011676  000534                          BR      TST41           ;;BR IF TESTED ALL SELECTED LINES
3679  011700  012767  002100  167312          MOV     #2100,$TMP7     ;INIT TI START WITH LO..EST SPEED
3680  011706  012767  177777  016330          MOV     #-1,TIMEC       ;INIT RELATIVE TIME CHECKER
3681  011714  012711  004000          2$:      MOV     #BIT11,(R1)     ;CLEAR THE DH11
3682  011720  156711  016266          3$:      BISB    LINE,(R1)       ;SELECT IT IN THE SCR
3683  011724  012761  177775  000010          MOV     #-3,BCR(R1)     ;SET BYTE COUNT TO XFER 3 CHARS
3684  011732  005061  000006                  CLR     CAR(R1)         ;GET TEST DATA STARTING AT LOC. 0
3685  011736  016761  167256  000004          MOV     $TMP7,LPR(R1)   ;SELECT A XMIT SPEED
3686  011744  016761  015234  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE TRANSMITTER
3687
3688  011752  012767  000001  016260          MOV     #1,TIMEA        ;INIT TIMER A
3689  011760  005067  016256                  CLR     TIMEB           ;INIT TIMER B
3690  011764  005711                  4$:      TST     (R1)            ;XMITTR DONE SET YET ?
3691  011766  100437                          BMI     5$              ;BR IF YES
3692  011770  004767  014750                  JSR     PC,TIMEIT       ;CALL THE TIMER
3693  011774  000773                          BR      4$              ;TIMER ROUTINE WILL MOVE RETURN PC
3694                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
3695
3696  011776  016767  167216  167176          MOV     $TMP7,$TMP0     ;SAVE AND SET UP THE SPEED CODE
3697  012004  000367  167172                  SWAB    $TMP0
3698  012010  006267  167166                  ASR     $TMP0
3699  012014  006267  167162                  ASR     $TMP0
3700  012020  042767  177760  167154          BIC     #177760,$TMP0
3701  012026  011103                          MOV     (R1),R3         ;GET THE WAS DATA
3702  012030  042703  000200                  BIC     #BIT07,R3       ;CLEAR UNINTERESTING BITS
3703  012034  010102                          MOV     R1,R2           ;MAKE REGADR = DEVADR
3704  012036  012704  100000                  MOV     #BIT15,R4       ;SET UP S/B DATA
3705  012042  156704  016144                  BISB    LINE,R4
3706  012046  004767  012302                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3707  012052  004567  012516                  JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
3708  012056  030212                          LINE
3709  012060  034144                          EM50+53
3710  012062  104053                          ERROR   53              ;TIMED OUT WAITING FOR XMIT DONE
3711  012064  000426                          BR      8$              ;GO TEST NEXT SPEED
3712
3713  012066  016703  016150          5$:      MOV     TIMEB,R3        ;GET THE WAS COUNT
3714  012072  016704  016146                  MOV     TIMEC,R4        ;GET LASTR CHECK COUNT
3715  012076  020304                          CMP     R3,R4           ;COMPARE RELATIVE TIMES
3716  012100  103420                          BLO     8$              ;BR IF THIS SPEED FASTER THAN LAST
3717                                                                  ;SPEED TESTED
3718
3719  012102  004767  015020          7$:      JSR     PC,SAPS         ;SAVE THE ERROR PSW
3720  012106  016702  167106                  MOV     $TMP7,R2        ;GET SPEED CODE AND RIGHT JUSTIFY
3721  012112  000302                          SWAB    R2
3722  012114  006202                          ASR     R2
3723  012116  006202                          ASR     R2
3724  012120  042702  177760                  BIC     #177760,R2      ;STRIP AWAY ALL JUNK
3725  012124  004767  012224                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3726  012130  004567  012440                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
3727  012134  030212                          LINE
3728  012136  031507                          EM17+41
3729  012140  104017                          ERROR   17              ;TRANSMITTER SPEED INCORRECT
3730
3731  012142  016767  016074  016074  8$:      MOV     TIMEB,TIMEC     ;SET UP NEW CHECK TIMER COUNT
```

```
3732  012150  062767  002100  167042          ADD     #2100,$TMP7     ;GENERATE NEXT SPEED
3733  012156  022767  035600  167034          CMP     #35600,$TMP7    ;DONE ALL SPEEDS ?
3734  012164  001253                          BNE     2$              ;BR IF NOT
3735  012166  000641                          BR      1$              ;GO TEST NEXT LINE
3736
```

```
                                ;;*****************************************************************
                                ;*TEST 41       RECEIVER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS
                                ;;*****************************************************************
012170  000004                  TST41:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST ID IDENTICAL TO TEST 42 EXCEPT IT WAITS FOR "DATA READY"
                                TO CHECK RECEIVER TIMING. THE SEQUENCE IS SIMILAR AND THE SAME TIMERS
                                ARE USED FOR ERROR CHECKING.

                                ERRORS:
                                *******

                                1.  [ERROR 54] IS CALLED TO REPORT RCVR TIMEOUT ERRORS
                                2.  [ERROR 20] IS CALLED TO REPORT RCVR TIMING ERRORS

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

                                DEBUG:          (SAME AS TEST 42)
                                ******

                                KEY LOGIC:      (SAME AS TEST 42 PLUS)
                                **********

                                        M7288   SH5,7,9,11       RX CLOCK NN L SIGNALS
                                        M7280   BUF DA LINE "N" UART PIN 19
                                                RX CLOCK LINE "N" UART PIN 17

                                %
012172  012767  012222  166710          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
012200  004767  012276          1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
012204  000532                  BR      TST42                   ;;BR IF TESTED ALL SELECTED LINES
012206  012767  002100  167004          MOV     #2100,$TMP7     ;INIT TO START WITH LOWEST SPEED
012214  012767  177777  016022          MOV     #-1,TIMEC       ;INIT RELATIVE TIME CHECKER
012222  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
012226  156711  015760          3$:     BISB    LINE,(R1)       ;SELECT IT IN THE SCR
012232  012761  177777  000010          MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO XFER 1 CHAR
012240  005061  000036                  CLR     CAR(R1)         ;GET TEST DATA STARTING AT LOC. 0
012244  016761  166750  000004          MOV     $TMP7,LPR(R1)   ;SELECT A XMIT SPEED
012252  016761  014726  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE TRANSMITTER

012260  012767  000001  015752          MOV     #1,TIMEA        ;INIT TIMER A
012266  005067  015750                  CLR     TIMEB           ;INIT TIMER B
012272  105711                  4$:     TSTB    (R1)            ;RCVR DONE YET ??
012274  100435                  BMI     5$                      ;BR IF YES
012276  004767  014442                  JSR     PC,TIMEIT       ;CALL THE TIMER
012302  000773                  BR      4$                      ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS

012304  016767  166710  166670          MOV     $TMP7,$TMP0     ;SAVE AND SET UP THE SPEED CODE
012312  006367  166664                  ASL     $TMP0
012316  006367  166660                  ASL     $TMP0
012322  000367  166654                  SWAB    $TMP0
```

```
                                                BIC     #177760,$TMP0
                                                MOV     (R1),R3         ;GET THE WAS DATA
                                                MOV     R1,R2           ;MAKE REGADR = DEVADR
                                                MOV     #BIT15+BIT07,R4 ;SET UP S/B DATA
                                                BISB    LINE,R4
                                                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
                                                JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
                                                .WORD   LINE
                                                .WORD   EM20+51
                                                ERROR   54              ;TIMED OUT WAITING FOR CHAR AVAIL
                                                BR      8$              ;GO TEST NEXT SPEED

012372  016703  015646          5$:   MOV     TIMEB,R3        ;GET THE WAS COUNT
012374  016704  015644                MOV     TIMEC,R4        ;GET THE CHECK COUNT
012400  020304                        CMP     R3,R4           ;COMPARE RELATIVE TIMES
012402  103420                        BLO     8$              ;BR IF TIME INDICATES THIS SPEED FASTER
                                                                ;THAN LAST SPEED

012404  004767  014516          7$:   JSR     PC,SAPS         ;SAVE THE ERROR PSW
012410  016702  166604                MOV     $TMP7,R2        ;GET SPEED CODE AND RIGHT JUSTIFY
012414  006302                        ASL     R2
012416  006302                        ASL     R2
012420  000302                        SWAB    R2
012422  042702  177760                BIC     #177760,R2      ;STRIP AWAY ALL JUNK
012426  004767  011[illegible]        JSR     PC,SUER2A       ;GO SET UP ERROR INFO
012432  004567  012136                JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
012436  000000                        .WORD   LINE
012440  031646                        .WORD   EM20+36
012442  104020                        ERROR   20              ;RECEIVER SPEED INCORRECT

012444  016767  015572  015572  9$:   MOV     TIMEB,TIMEC     ;SET UP NEW CHECK TIMER COUNT
012452  062767  002100  166540        ADD     #2100,$TMP7     ;GENERATE NEXT SPEED
012460  022767  035600  166532        CMP     #35600,$TMP7    ;DONE ALL SPEEDS ?
012466  001255                        BNE     2$              ;BR IF NOT
012470  000643                        BR      1$              ;GO TEST NEXT LINE
```

3829

```
                        ;;*****************************************************************
                        ;*TEST 42       VERIFY STORAGE OVERFLOW - NON MAINT. MODE - ALL SELECTED LINES
                        ;;*****************************************************************
012472  000004          T$T42:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

                                THIS TEST VERIFIES THAT THE STORAGE OVERFLOW BIT (SCR14) SETS
                        AND CLEARS PROPERLY FOR ALL SELECTED LINES. THE TEST SEQUENCE IS AS
                        FOLLOWS:

                                1. SET UP THE ERROR RETURN
                                2. SELECT A LINE NUMBER TO TEST - GO TO NEXT TEST IF ALL
                                   SELECTED LINES HAVE BEEN TESTED.
                                3. PRIME THE SELECTED LINE TO XMIT 65(10) CHARS.
                                4. ACTIVATE THE SELECTED LINE AND WAIT FOR STORAGE OVERFLOW
                                   TO SET (SCR14=1)
                                5. IF SCR14 FAILS TO SET ON TIME - REPORT ERROR AND THEN
                                   CONTINUE WITH THE NEXT LINE (STEP 2).
                                6. IF IT SETS OK - READ THE "NRC" REG TWICE TO EMPTY TWO
                                   WORDS FROM THE SILO.
                                7. AFTER A BRIEF STALL, VERIFY THAT SCR14 HAS CLEARED -
                                   IF NOT REPORT ERROR AND CONTINUE WITH NEXT LINE (STEP2)
                                8. IF IT CLEARS OK, VERIFY THAT THE FILL COUNT (SSR1[15:08])
                                   CONTAINS A 77(8) - IF NOT REPORT ERROR AND CONTINUE
                                   WITH NEXT LINE  (STEP 2).
                                9. REPEAT STEPS 2 THRU 8 UNTIL ALL SELECTED LINES HAVE BEEN
                                   TESTED.

                        ERRORS:
                        *******

                        1. [ERROR  57] IS CALLED TO REPORT ALL ERRORS DETECTED.


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

                        DEBUG:
                        ******

                        1. PROBLEM IS MOST LIKELY ON THE M7289 MODULE (SH4) OR SOME
                           SIGNAL FEEDING THIS LOGIC.

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

                                M7289   SH4     STORAGE OVERFLOW L      E43-12
                                                READY IN PULSE H        E40-11
                                                UC1 MASTER DA H         BH2
                                                UC2 MASTER DA H         BD2

                        .
012474  012767  012510  166406          MOV     #2$,$LPERR      ;SET UP ERROR RETURN
012502  004767  011774          1$:     JSR     PC,$ELINE       ;GO SELECT A LINE NO. TO TEST
```

E01


```
3896  012536  000535                      BR      TST43           ;;BR IF DONE ALL SELECTED LINES
3897  012610  012711  004000      2$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
3898  012614  010102                      MOV     R1,R2           ;MAKE REGADR = DEVADR
3899  012616  116711  015470              MOVB    LINE,(R1)       ;SELECT A LINE
3900  012622  005061  000006              CLR     CAR(R1)         ;SET UP CURRENT ADDRESS
3901  012626  012761  177677  000010      MOV     #-65.,BCR(R1)   ;SET UP BYTE COUNT
3902  012634  012761  033500  000004      MOV     #33500,LPR(R1)  ;SET UP LPR: 9600 BAUD,5 BIT CHARS
3903  012642  016761  014436  000012      MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
3904
3905  012650  012767  000001  015462      MOV     #1,TIMEA        ;INIT TIMERS
3906  012656  005067  015460              CLR     TIMEB
3907  012662  032711  040000      3$:     BIT     #BIT14,(R1)     ;STOREGE OVERFLOW YET ??
3908  012666  001024                      BNE     4$              ;BR IF YES YOU SHOULD GET IT
3909  012670  004767  014150              JSR     PC,TIMEIT       ;CALL TIMER
3910  012674  000772                      BR      3$              ;BR IF NO TIME OUT
3911
3912  012676  004767  014324              JSR     PC,SAPS         ;GO SAVE PSW
3913  012602  012704  040000              MOV     #BIT14,R4       ;SET UP S/B DATA
3914  012606  156704  015400              BISB    LINE,R4
3915  012612  011103                      MOV     (R1),R3         ;SET UP WAS DATA
3916  012614  042703  137760              BIC     #137760,R3      ;CLEAR UNINTERESTING BITS
3917  012620  004767  011530              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3918  012624  004567  011744              JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
3919  012630  030212                      LINE
3920  012632  034631                      EMS7+44
3921  012634  104057                      ERROR   57              ;STORAGE OVERFLOW FAILED TO SET
3922  012636  000721                      BR      1$              ;GO TRY NEXT LINE
3923
3924  012640  016167  000002  166352  4$:     MOV     NRC(R1),$TMP7   ;READ THE SILO
3925  012646  016167  000002  166344          MOV     NRC(R1),$TMP7   ;READ IT AGAIN
3926  012654  012705  001000              MOV     #1000.,R5       ;INIT STALL COUNTER
3927  012660  005305              41$:    DEC     R5              ;COUNT TIMER
3928  012662  001376                      BNE     41$             ;BR IF NO TIMEOUT
3929  012664  032711  040000              BIT     #BIT14,(R1)     ;DID OVERFLOW GO AWAY ?
3930  012670  001420                      BEQ     5$              ;BR IF YES
3931
3932  012672  004767  014230              JSR     PC,SAPS         ;GO SAVE THE PSW
3933  012676  005004                      CLR     R4              ;SET UP S/B DATA
3934  012700  156704  015306              BISB    LINE,R4
3935  012704  011103                      MOV     (R1),R3         ;SET UP WAS DATA
3936  012706  042703  137760              BIC     #137760,R3      ;CLEAR UNINTERESTING BITS
3937  012712  004767  011436              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3938  012716  004567  011652              JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
3939  012722  030212                      LINE
3940  012724  034631                      EMS7+44
3941  012726  104057                      ERROR   57              ;STORAGE BIT FAILED TO CLEAR
3942  012730  000664                      BR      1$              ;GO TRY NEXT LINE
3943
3944  012732  122761  000077  000017  5$:     CMPB    #77,SSR+1(R1)   ;WAS IT REALLY 65. ??
3945  012740  001660                      BEQ     1$
3946
3947  012742  004767  014160              JSR     PC,SAPS         ;GO SAVE PSW
3948  012746  012704  037400              MOV     #37400,R4       ;SET UP S/B DATA
3949  012752  062702  000016              ADD     #SSR,R2         ;SET UP REGADR
3950  012756  016103  000016              MOV     SSR(R1),R3      ;SAVE WAS DATA
3951  012762  004767  011366              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
```

F01


```
 3942  012766  004567  011602          JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
 3943  012772  030212                  LINE
 3944  012774  034631                  EM57+44
 3945  012776  104057                  ERROR   57              ;READING SILO F ILED TO DEC SSR OR
 3946                                                          ;STORAGE OVFL SE  GT WRONG COUNT
 3947  013000  000640                  BR      1$              ;GO TRY NEXT LINE
```

```
                                ;;*****************************************************************
                                ;*TEST 43       BASIC DATA TEST - ALL SELECTED LINES/ALL CHAR LENGTHS
                                ;;*****************************************************************
013002  000004                  TST43:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT A SINGLE ALL ONES CHAR. CAN BE TRANS-
                                MITTED AND RECEIVED ON ALL SELECTED LINES AT ALL FOUR CHAR LENGTHS
                                (5, 6, 7, AND 8 BITS). THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SET UP THE ERROR LOOP RETURN
                                        2. SELECT A LINE NO. TO TEST - GO TO TEST 43 IF DONE ALL
                                           SELECTED LINES.
                                        3. GET A TEST CHARACTER FROM THE DATA TABLE AND UPDATE THE
                                           TABLE POINTER.
                                        4. CLEAR THE DH11
                                        5. PRIME THE SELECTED LINE TO XMIT ONE CHAR AT 9600 BAUD
                                        6. WAIT FOR "CHAR AVAIL" TO SET - IF TIMEOUT REPORT ERROR AND
                                           RESTART AT STEP 3.
                                        7. IF NO TIMEOUT - CHECK DATA AND REPORT ANY ERRORS
                                        8. INCREMENT "SCR" REG TO CHANGE CHAR LENGTH - IF DONE
                                           ALL FOUR GO TO STEP 2 - IF NOT THEN STEP 4.

                                ERRORS:
                                ******

                                1. [ERROR  55] IS CALLED TO REPORT RCVR TIMEOUT.
                                2. [ERROR  23] IS CALLED TO REPORT DATA COMPARE ERRORS

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

                                DEBUG:
                                ******

                                1. IF FAULT AFFECTS ONLY ONE LINE AT ALL CHAR LENGTHS, SUSPECT A
                                   BAD UART MODULE M7280.

                                2. IF FAULT AFFECTS ONLY ONE BIT ON ALL LINES, SUSPECT THE
                                    THE M7279 MODULE.
                                3. IF FAULT AFFECTS ONLY CERTAIN CHAR LENGTHS, SUSPECT EITHER THE M7279
                                   OR THE UART MODULE M7280.

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

                                        M7280'S UART CHIPS PINS (12:05)

                                        M7279   SH1     E1,E2,E6, OR E7

                                        M7278   SH8     NB2 LPR 01 H    FH1
                                                        NB1 LPR 00 H    FH2
                                %
013004  012767  013032  166076          MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN
```

H01


```
 4004  013012  004767  011464          1$:   JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
 4005  013016  000511                        BR      TST44           ;;BR IF DONE ALL SELECTED LINES
 4006  013020  012705  030226                MOV     #TDATA2,R5      ;GET POINTER TO DATA TABLE
 4007  013024  005002                        CLR     R2              ;INIT R2 TO START AT CHAR LENGTH OF 5 BITS
 4008  013026  012567  166156          2$:   MOV     (R5)+,$TMP7     ;PUT TEST CHAR IN XMIT BUFFER
 4009  013032  012711  004000          3$:   MOV     #BIT11,(R1)     ;CLEAR THE DH11
 4010  013036  156711  015150                BISB    LINE,(R1)       ;SELECT THE LINE
 4011  013042  012761  177777  000010        MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO -1
 4012  013050  012761  001220  000006        MOV     #$TMP7,CAR(R1)  ;SET CURRENT ADDRESS REG
 4013  013056  012761  033500  000004        MOV     #33500,LPR(R1)  ;SET BAUD RATE TO 9600
 4014  013064  050261  000004                BIS     R2,LPR(R1)      ;SELECT CHAR LENGTH
 4015  013070  156761  014110  000012        BISB    LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
 4016
 4017  013076  012767  000001  015134        MOV     #1,TIMEA        ;INIT TIMER A
 4018  013104  005067  015132                CLR     TIMEB           ;INIT TIMER B
 4019  013110  105711                  4$:   TSTB    (R1)            ;RCVR DONE YET ??
 4020  013112  100424                        BMI     5$              ;BR IF YES
 4021  013114  004767  013624                JSR     PC,TIMEIT       ;CALL THE TIMER
 4022  013120  000773                        BR      4$              ;TIMER ROUTINE WILL MOVE RETURN PC
 4023                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS
 4024
 4025  013122  004767  014000                JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4026  013126  011103                        MOV     (R1),R3         ;GET THE SCR
 4027  013130  042703  177560                BIC     #177560,R3      ;CLEAR UNINTERESTING BITS
 4028  013134  012704  000200                MOV     #200,R4         ;SET UP S/B DATA
 4029  013140  156704  015046                BISB    LINE,R4
 4030  013144  004767  011204                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4031  013150  004567  011420                JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
 4032  013154  030212                        LINE
 4033  013156  031766                        EM22+51
 4034  013160  104055                        ERROR   55              ;CHAR AVAIL FAILED TO SET ON TIME
 4035  013162  000422                        BR      6$              ;GO TEST NEXT CHAR LENGTH
 4036
 4037  013164  016103  000002          5$:   MOV     NRC(R1),R3      ;GET THE WAS DATA
 4038  013170  012704  000200                MOV     #200,R4         ;SET UP THE S/B DATA IN R4
 4039  013174  156704  015012                BISB    LINE,R4
 4040  013200  000304                        SWAB    R4
 4041  013202  156704  166012                BISB    $TMP7,R4
 4042  013206  020304                        CMP     R3,R4           ;WAS THE RCVD DATA CORRECT ??
 4043  013210  001407                        BEQ     6$              ;BR IF YES
 4044
 4045  013212  004767  011132                JSR     PC,SUER2        ;GO SET UP THE ERROR INFO
 4046  013216  004567  011352                JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
 4047  013222  030212                        LINE
 4048  013224  032027                        EM23+36
 4049  013226  104023                        ERROR   23              ;DATA COMPARE ERROR
 4050
 4051  013230  005202                  6$:   INC     R2              ;DO NEXT CHAR LENGTH ON SELECTED LINE
 4052  013232  022702  000004                CMP     #4,R2           ;HAVE WE DONE ALL FOUR CHAR LENGTHS ??
 4053  013236  001273                        BNE     2$              ;BR IF NOT
 4054  013240  000664                        BR      1$              ;GO DO NEXT LINE
```

```
                        ;;*****************************************************************
                        ;*TEST 44       SINGLE LINE DATA TEST - ALL SELECTED LINES
                        ;;*****************************************************************
013242  000004          TST44:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

                                THIS TEST TRANSMITS AND RECEIVES A BINARY COUNT PATTERN
                        (000 - 377) ON ALL SELECTED LINES. THE TEST SEQUENCE IS AS
                        FOLLOWS:

                                1. SET UP THE ERROR LOOP RETURN
                                2. GO SELECT A LINE NO. TO TEST - IF DONE ALL SELECTED
                                   LINES THEN GO TO TEST 44.
                                3. CLEAR THE DH11 AND PRIME THE SELECTED LINE TO XMIT
                                   TO XMIT 256. CHARS AT 9600. BAUD - 8 BIT CHARS.
                                4. SET UP R5 TO POINT TO RCVR CORE BUFFER.
                                5. ACTIVATE THE SELECTED XMITTER.
                                6. WAIT FOR "CHAR AVAIL" TO SET BEFORE READING THE SILO.
                                   IF RCVR TIMEOUT REPORT ERROR AND RESTART AT STEP 2.
                                7. IF NO TIMEOUT READ THE SILO AND STORE THE WORD IN THE RCVR
                                   CORE BUFFER - WHEN THE BUFFER IS FULL GO TO STEP8 IF NOT
                                   THEN GO TO STEP 6.
                                8. COMPARE THE XMIT AND RCVR CORE IMAGE BUFFERS AND REPORT
                                   ALL DATA COMPARE ERRORS.
                                9. CHECK THE "BAR", "BCR", AND "CAR" REGISTERS FOR CORRECT
                                   CONTENTS - REPORT ALL ERRORS.
                               10. GO TO STEP 2

                                       NOTE:   THIS TEST USES THE MAINT. BIT (SCR09=1) TO
                                               TURN THE DATA AROUND INSTEAD OF THE MAINT
                                               CONNECTORS OR MODULE.

                        ERRORS:
                        *******

                        1. [ERROR  22] IS CALLED TO REPORT "DATA AVAIL" TIMEOUT
                        2. [ERROR  37] "        "       " DATA COMPARE ERRORS
                        3. [ERROR  40] "        "       " "BAR" REG NOT CLEARED
                        4. [ERROR  10] "        "       " "BCR" REG NOT ALL ZEROES
                        5. [ERROR   7] "        "       " "CAR" REG NOT UPDATED CORRECTLY

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

                        DEBUG:
                        *******

                        1. IF THE FAULT AFFECTS ONE OR MORE LINES IN AN 8 LINE GROUP <15:08>
                           OR <07:00>, SWAP THE M7280 MODULES. IF THE FAULT SHIFTS SO THAT
                           THE ERROR INDICATES DIFFERENT LINES THE PROBLEM IS MOST LIKELY
                           THE M7280 THE SYMPTOM SHIFTED TO.

                        2. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS ON
                           ONE LINE THE FAULT IS MOST LIKELY A "UART" CHIP.
```

```
                                3. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS
                                   ON ALL LINES SUSPECT THE DATA PATHS EXTERNAL TO, THE M7280 MODULES.

                                4. IF THE FAULT CAUSES NO DATA ERRORS BUT GIVES "BAR", "BCR", OR "CAR" ERRORS
                                   SUSPECT THE M7278 OR M7277 MODULES RESPECTFULLY.

                                KEY LOGIC       (REFER TO TEST 43)
                                ***********

                                %
013244  012767  013266  165636          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
013252  004767  011224          1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
013256  000401                          BR      11$             ;;BR IF ALL SELECTED LINES DONE
013260  000402                          BR      2$              ;GO TEST IT
013262  000167  000424          11$:    JMP     8$              ;EXIT TEST
013266  016701  013700          2$:     MOV     DHADR,R1        ;RESET DEVADR
013272  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE DH11
013276  156711  014710                  BISB    LINE,(R1)       ;SET SELECT BITS IN SCR
013302  012761  037112  000006          MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDRESS REG
013310  012761  177400  000010          MOV     #-400,BCR(R1)   ;SET UP BYTE COUNT
013316  012761  033503  000004          MOV     #33503,LPR(R1)  ;SET LINE PARAMETERS
013324  012705  036112                  MOV     #RBUF,R5        ;SET UP POINTER TO INPUT DATA BUFFER
013330  052711  001000                  BIS     #BIT09,(R1)     ;SET MAINT MODE BIT
013334  016761  013644  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE

013342  012767  000002  014670          MOV     #2,TIMEA        ;INIT TIMER A
013350  005067  014666                  CLR     TIMEB           ;INIT TIMER B
013354  105711                  3$:     TSTB    (R1)            ;RCVR DONE YET ??
013356  100425                          BMI     4$              ;BR IF YES
013360  004767  013360                  JSR     PC,TIMEIT       ;CALL THE TIMER
013364  000773                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS

013366  004767  013534                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
013372  010102                          MOV     R1,R2           ;SET UP REGADR
013374  011203                          MOV     (R2),R3         ;GET THE WAS DATA
013376  042703  176400                  BIC     #176400,R3      ;CLEAR JUNK BITS
013402  012704  001200                  MOV     #1200,R4        ;SET UP S/B DATA
013406  156704  014600                  BISB    LINE,R4
013412  004767  010736                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
013416  004567  011152                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
013422  030212                          LINE
013424  031766                          EM22+51
013426  104022                          ERROR   22              ;CHAR AVAIL TIMEOUT
013430  000710                          BR      1$              ;GO TRY NEXT LINE

013432  016125  000002          4$:     MOV     NRC(R1),(R5)+   ;SAVE THE RECEIVED DATA
013436  022705  037112                  CMP     #RBUF+1000,R5   ;INPUT BUFFER FULL ??
013442  001344                          BNE     3$              ;BR IF NOT

013444  012702  037112                  MOV     #TBUF,R2        ;SET UP POINTER TO OUTPUT BUFFER
013450  012701  036112                  MOV     #RBUF,R1        ;SET UP POINTER TO INPUT BUFFER
013454  111204                  5$:     MOVB    (R2),R4         ;SET UP S/B DATA IN R4
013456  042704  177400                  BIC     #177400,R4
013462  000304                          SWAB    R4
```

```
 4167  013464  156704  014522                BISB    LINE,R4
 4168  013470  152704  000200                BISB    #200,R4
 4169  013474  000304                        SWAB    R4
 4170  013476  011103                        MOV     (R1),R3         ;GET THE WAS DATA
 4171  013500  020304                        CMP     R3,R4           ;DATA CORRECT ??
 4172  013502  001407                        BEQ     6$              ;BR IF YES
 4173
 4174  013504  004767  010640                JSR     PC,SUER2        ;GO SET UP ERROR INFO
 4175  013510  004567  011060                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
 4176  013514  030212                        LINE
 4177  013516  033244                        EM37+33
 4178  013520  104037                        ERROR   37              ;DATA COMPARE ERROR
 4179
 4180
 4181  013522  005202            6$:         INC     R2              ;UPDATE DATA BUFFER POINTERS
 4182  013524  062701  000002                ADD     #2,R1           ;
 4183  013530  022701  037112                CMP     #RBUF+1000,R1   ;COMPARED ALL 256. CHARS ??
 4184  013534  001347                        BNE     5$              ;BR IF NOT
 4185
 4186  013536  016701  013430                MOV     DHADR,R1        ;RESET DEVADR
 4187  013542  010102                        MOV     R1,R2           ;SET UP REGADR
 4188  013544  062702  000012                ADD     #BAR,R2
 4189  013550  005712                        TST     (R2)            ;WAS THE "BAR" ALL ZEROES ??
 4190  013552  001413                        BEQ     7$              ;BR IF YES
 4191
 4192  013554  004767  013346                JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4193  013560  011203                        MOV     (R2),R3         ;GET THE WAS DATA
 4194  013562  005004                        CLR     R4              ;SET UP S/B DATA
 4195  013564  004767  010564                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4196  013570  004567  011000                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
 4197  013574  030212                        LINE
 4198  013576  033307                        EM40+40
 4199  013600  104040                        ERROR   40              ;"BAR" REG NOT ALL ZEROES
 4200
 4201  013602  010102            7$:         MOV     R1,R2           ;SET UP REGADR
 4202  013604  062702  000010                ADD     #BCR,R2
 4203  013610  005712                        TST     (R2)            ;BYTE COUNT REG ALL ZEROES ?
 4204  013612  001413                        BEQ     71$             ;;BR IF BYTE COUNT ZERO
 4205
 4206  013614  004767  013306                JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4207  013620  011203                        MOV     (R2),R3         ;GET THE WAS DATA
 4208  013622  005004                        CLR     R4              ;SET UP THE S/B DATA
 4209  013624  004767  010524                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4210  013630  004567  010740                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
 4211  013634  030212                        LINE
 4212  013636  031114                        EM10+44
 4213  013640  104010                        ERROR   10              ;BYTE COUNT NOT ALL ZEROES
 4214
 4215  013642  010102            71$:        MOV     R1,R2           ;SET UP REGADR
 4216  013644  062702  000006                ADD     #CAR,R2
 4217  013650  022712  037512                CMP     #TBUF+400,(R2)  ;DID "CAR" INCREMENT PROPERLY ?
 4218  013654  001414                        BEQ     72$             ;BR IF YES
 4219
 4220  013656  004767  013244                JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4221  013662  011203                        MOV     (R2),R3         ;GET THE WAS DATA
 4222  013664  012704  037512                MOV     #TBUF+400,R4    ;SET UP S B DATA
```

```
 4223  013670  004767  010460                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4224  013674  004567  010674                JSR     R5,SUNUM        ;GO PUT LINE NO IN MESSAGE
 4225  013700  030212                        LINE
 4226  013702  031045                        EM7+47
 4227  013704  104007                        ERROR   7               ;"CAR" NOT UPDATED CORRECTLY
 4228
 4229  013706  000167  177340        72$:    JMP     1$              ;GO DO NEXT LINE
 4230
 4231  013712  000240                9$:     NOP                     ;EXIT POINT
```

```
                                ;;*****************************************************************
                                ;*TEST 45       BASIC PARITY LOGIC TEST - ALL SELECTED LINES - ODD PARITY
                                ;;*****************************************************************
013714  000004                  TST45:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ****************

                                        THIS TEST VERIFIES THE ODD PARITY FUNCTION FOR ALL SELECTED
                                LINES USING THE "BREAK" FUNCTION TO FORCE PARITY ERRORS. REFER TO
                                THE FLOW CHARTS IN THE PROGRAM DOCUMENTATION FOR TEST SEQUENCES.

                                ERRORS:
                                *******

                                1. [ERROR  22] IS CALLED TO REPORT RCVR TIMEOUT
                                2. [ERROR  33] IS CALLED TO REPORT DATA/PARITY ERRORS

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

                                DEBUG:
                                ******

                                1. IF FAULT AFFECTS ALL LINES SUSPECT THE M7278 MODULE.

                                2. IF IT AFFECTS ONLY ONE LINE SUSPECT THE "UART" MODULE FOR THAT LINE.


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

                                        M7278   SH7     PEN LPR04 L     FF2
                                                        PEV LPR05 L     FN1
                                %
013716  012767  013732  165164           MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
013724  004767  010552          1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
013730  000506                  .       BR      TST46           ;;BR IF ALL SELECTED LINES DONE
013732  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
013736  012704  000260                  MOV     #260,R4         ;SET UP THE S/B DATA IN R4
013742  156704  014244                  BISB    LINE,R4
013746  000304                          SWAB    R4
013750  156711  014236                  BISB    LINE,(R1)       ;SET LINE NO. IN SCR
013754  012767  000377  014242          MOV     #377,TDATA1     ;LOAD XMIT BUFFER WITH TEST CHARACTER
013762  012761  073563  000004          MOV     #73563,LPR(R1)  ;SET UP THE LINE PARAMETERS
013770  012761  177777  000010          MOV     #-1,BCR(R1)     ;LOAD THE BYTE COUNT REG
013776  012761  030224  000006          MOV     #TDATA1,CAR(R1) ;LOAD THE BUS ADDR REG
014004  016761  013174  000014          MOV     LINMSK,BKR(R1)  ;SET BREAK BIT FOR SELECTED LINE
014012  016761  013166  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE XMITTR

014020  012767  000001  014212          MOV     #1,TIMEA        ;INIT TIMER A
014026  005067  014210                  CLR     TIMEB           ;INIT TIMER B
014032  105711                  3$:     TSTB    (R1)            ;RCVR DONE YET ??
014034  100423                          BMI     4$              ;BR IF YES
014036  004767  012702                  JSR     PC,TIMEIT       ;CALL THE TIMER
014042  000773                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
```

```
                                                        ;AROUND THIS BRANCH IF TIME OUT OCCURS

014044  004767  013056            JSR     PC,SAPS       ;SAVE THE ERROR PSW
014050  011103                    MOV     (R1),R3       ;GET THE WAS DATA
014052  012704  100200            MOV     #100200,R4    ;SET UP THE S/B DATA
014056  156704  014130            BISB    LINE,R4
014062  010102                    MOV     R1,R2         ;SET UP REGADR
014064  004767  010264            JSR     PC,SUER2A     ;GO SET UP ERROR INFO
014070  004567  010500            JSR     R5,SUNUM      ;PUT LINE NO. IN MESSAGE
014074  030212                    LINE
014076  031766                    EM22+51
014100  104022                    ERROR   22            ;TIMED OUT WAITING FOR DATA AVAIL
014102  000710                    BR      1$            ;GO TEST NEXT LINE

014104  016103  000002      4$:   MOV     NRC(R1),R3    ;GET THE WAS DATA
014110  020304                    CMP     R3,R4         ;CORRECT DATA RECEIVED ??
014112  001704                    BEQ     1$            ;BR IF YES

014114  004767  013006            JSR     PC,SAPS       ;SAVE THE ERROR PSW
014120  010102                    MOV     R1,R2         ;SET UP THE REGADR
014122  062702  000002            ADD     #NRC,R2
014126  004767  010222            JSR     PC,SUER2A     ;GO SET UP ERROR INFO
014132  004567  010436            JSR     R5,SUNUM      ;PUT LINE NO. IN MESSAGE
014136  030212                    LINE
014140  032760                    EM33+40
014142  104033                    ERROR   33            ;INCORRECT DATA OR PARITY ERROR
014144  000667                    BR      1$            ;GO TEST NEXT LINE
```

```
                        ;;*****************************************************************
                        ;*TEST 46       MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES
                        ;;*****************************************************************
014146  000004          TST46:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        ***************

                             THIS TEST VERIFIES ALL SELECTED LINES CAN TRANSMIT AND RECEIVE
                        A BINARY COUNT PATTERN WHEN RUN CONCURRENTLY. ALL CHAR LENGTHS (5, 6, 7,
                        AND 8 BITS) ARE TESTED WITH BOTH EVEN AND ODD PARITY CHECKING SPECIFIED.
                        THE TEST ACTUALLY INCLUDES EIGHT SUB-TESTS - THE PARAMETERS FOR EACH
                        SUB-TEST RETRIEVED FROM A TABLE TAGGED "PRTYTB:".  REFER TO THIS TABLE
                        TO DETERMINE THE TEST SEQUENCE.


                        ERRORS:
                        ********

                        1. [ERROR  41] IS CALLED TO REPORT FALSE RECEIVER INTRS.
                        2. [ERROR  42] IS CALLED TO REPORT SILO OVERFLOW ERRORS
                        3. [ERROR  34] IS CALLED TO REPORT PARITY/DATA ERRORS
                        4. [ERROR  35] IS CALLED TO REPORT TEST TIMEOUTS

                        SYNC:   (NONE)
                        *****

                        DEBUG:  (REFER TO TEST 45)
                        ******

                        KEY LOGIC:      (REFER TO TEST 45)
                        **********

                        %
014150  012767  014166  164732          MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
014156  012705  027254                  MOV     #PRTYTB+4,R5    ;SET UP POINTER TO TEST PARAMETERS
014162  005067  165032                  CLR     $TMP7           ;START WITH SUB TEST #00
014166  162705  000004          1$:     SUB     #4,R5           ;RESET POINTER FOR ERROR LOOPS
014172  005367  165022                  DEC     $TMP7           ;RESET SUB TEST # FOR ERROR LOOP
014176  022705  027310          2$:     CMP     #PRTYTB+40,R5   ;DONE ALL 9. SUB TESTS ??
014202  001456                  BEQ     21$             ;BR IF YES
014204  012706  001100                  MOV     #STACK,SP       ;RESET STACK POINTER FOR ERROR LOOPS
014210  016701  012756                  MOV     DHADR,R1        ;RESET DEVADR FOR ERROR LOOPS
014214  012567  164776                  MOV     (R5)+,$TMP6     ;GET THE BYTE COUNT PARAMETER
014220  012567  164770                  MOV     (R5)+,$TMP5     ;GET THE LINE PARAMETERS
014224  005267  164770                  INC     $TMP7           ;GENERATE NEW SUB-TEST NO.
014230  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE DH11
014234  004767  010516                  JSR     PC,SUPPAR       ;GO SET UP PARAMETERS
014240  016167  000004  164734          MOV     LPR(R1),$TMP0   ;SAVE CURRENT LINE PARAMETERS
014246  016767  012730  164734          MOV     LINSEL,$TMP3    ;SAVE SELECTED LINES PARAMETER
014254  004767  012632                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
014260  016702  012710                  MOV     DHVCT,R2        ;SET UP THE VECTOR
014264  012722  014356                  MOV     #3$,(R2)+       ;GO TO 3$ ON RCVR INTERRUPT
014270  116712  013702                  MOVB    DHRLVL,(R2)
014274  012711  000100                  MOV     #100,(R1)       ;ENABLE CHAR AVAIL INTERRUPTS
014300  016767  012676  013142          MOV     LINSEL,LINACT   ;FLAG ALL SELECTED LINES ACTIVE
```

```
014306 016761 012670 000012          MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL SELECTEDLINES
014314 116761 164562 164664          MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
014322 042761 177400 164656          BIC     #177400,$TMP2
014330 004767 012542                 JSR     PC,CHPS1        ;GO CLEAR PSW
014334 000167 000232                 JMP     2$              ;GO WAIT FOR INTERRUPTS

014340 012706 001100         21$:    MOV     #STACK,SP       ;RESTORE THE SP
014344 004767 012526                 JSR     PC,CHPS1        ;GO CLEAR PSW
014350 004767 012342                 JSR     PC,RESTRP       ;RESTORE TRAP CATCHER
014354 000554                        BR      TST47           ;;GO TO NEXT TEST

                             ;RECEIVER INTERRUPT SERVICE ROUTINE

014356 005067 164630         3$:     CLR     $TMP4
014362 105711                        TSTB    (R1)            ;CHAR AVAIL SET
014364 100404                        BMI     4$              ;BR IF YES

014366 012711 004000                 MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
014372 104041                        ERROR   41              ;RCVR FALSE INTERRUPT - CHAR AVAIL NOT SET
014374 000700                        BR      2$              ;GO TRY NEXT SUB TEST

014376 032711 040000         4$:     BIT     #BIT14,(R1)     ;SILO OVERFLOW ??
014402 001404                        BEQ     5$              ;BR IF NOT

014404 012711 004000                 MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
014410 104042                        ERROR   42              ;SILO OVERFLOW ERROR
014412 000671                        BR      2$              ;GO TRY NEXT SUB TEST

014414 016103 000002         5$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
014420 010302                        MOV     R3,R2           ;EXTRACT AND SAVE LINE NO.
014422 000302                        SWAB    R2
014424 042702 177760                 BIC     #177760,R2
014430 006302                        ASL     R2              ;GENERATE TABLE OFFSETR
014432 005267 164554                 INC     $TMP4
014436 022767 000101 164546          CMP     #101,$TMP4
014444 001744                        BEQ     3$
014446 036267 030136 012526          BIT     $LNSEL(R2),LINSEL       ;IS THIS ONE OF THE SELECTED LINES?
014454 001757                        BEQ     5$              ;IF NOT, THIS IS GARBAGE, SO CHECK NEXT FILD ENTRY.
014456 010267 164530                 MOV     R2,$TMP4
014462 006267 164524                 ASR     $TMP4
014466 026203 036112                 CMP     RBUF(R2),R3     ;CORRECT DATA RECEIVED ??
014472 001426                        BEQ     6$              ;BR IF YES

014474 004767 012426                 JSR     PC,SAPS         ;SAVE THE ERROR PSW
014500 012711 004000                 MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
014504 016204 036112                 MOV     RBUF(R2),R4     ;SET UP S-B DATA
014510 062701 000002                 ADD     #NRC,R1         ;SET UP WAS ADDRESS
014514 062702 036112                 ADD     #RBUF,R2        ;SET UP S-B ADDRESS
014520 004767 007630                 JSR     PC,SUER2A       ;GO SET UP ERROR INFO
014524 004567 010044                 JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
014530 001212                        $TMP4
014532 033034                        EM34+51
014534 004567 010034                 JSR     R5,SUNUM        ;PUT SUBTEST NO. IN MESSAGE
014540 001220                        $TMP7
014542 033052                        EM34+67
014544 104034                        ERROR   34              ;PARITY DATA COMPARE ERROR
```

```
                                                BR      2$              ;GO TRY NEXT SUBTEST

014550  105262  036112          6$:     INCB    RBUF(R2)        ;GENERATE NEW RCVD DATA
014554  005262  027452                  INC     MULPTB(R2)      ;COUNT ONE BYTES RECEIVED
014560  001003                          BNE     61$             ;BR IF NOT DONE
014562  046267  027410  012660          BIC     LINBIT(R2),LINACT ;FLAG THIS LINE DONE
014570  000002          61$:    RTI                             ;RETURN TO WAIT ROUTINE

                        ;WAIT ROUTINE

014572  012767  000002  013440  7$:     MOV     #2,TIMEA        ;INIT TIMER A
014600  005067  013436                  CLR     TIMEB           ;INIT TIMER B
014604  005761  000012          8$:     TST     BAR(R1)         ;ALL LINES DONE XMITTING ??
014610  001413                          BEQ     9$              ;BR IF YES
014612  004767  012126                  JSR     PC,TIMEIT       ;CALL THE TIMER
014616  000772                          BR      9$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS

014620  016167  000012  164362          MOV     BAR(R1),$TMP3   ;SAVE THE ACTIVE LINES FLAG
014626  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
014632  104035                          ERROR   35              ;TIMED OUT WAITING FOR TRANSMITTERS TO FINISH
014634  000167  177336                  JMP     2$              ;GO TRY NEXT SUBTEST

014640  012767  000001  013372  9$:     MOV     #1,TIMEA        ;INIT TIMER A
014646  005067  013370                  CLR     TIMEB           ;INIT TIMER B
014652  005767  012572          10$:    TST     LINACT          ;ALL CHARS RECEIVED ?
014656  001411                          BEQ     11$             ;BR IF YES
014660  004767  012060                  JSR     PC,TIMEIT       ;CALL THE TIMER
014664  000772                          BR      10$             ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS

014666  016767  012556  164314          MOV     LINACT,$TMP3    ;SET UP ACTIVE LINE PARAMETER
014674  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
014700  104035                          ERROR   35              ;SILO EMPTY TIMEOUT
014702  000167  177270          11$:    JMP     2$              ;GO TRY NEXT SUB TEST
```

```
                        ;;*****************************************************************
                        ;*TEST 47       AUTO ECHO TEST 1 - ALL SELECTED LINES
                        ;;*****************************************************************
014706  000004          TST47:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

                                THIS TEST VERIFIES THAT ALL SELECTED LINES CAN TURN AROUND
                        A SINGLE TEST CHARACTER (377) AND (000) IN AUTO ECHO MODE. THE TEST SEQUENCE
                        IS AS FOLLOWS:

                                1. SET UP THE ERROR LOOP RETURN
                                2. RETRIEVE THE AUTO-ECHO TEST DATA FROM "AETAB:" AND
                                   UPDATE THE POINTER.
                                3. GO SELECT A LINE NO. TO TEST - GO TO STEP 10 IF DONE
                                   ALL SELECTED LINES.
                                4. CLEAR THE "CAR" AND "BCR" MEMORIES.
                                5. PRIME THE SELECTED LINE TO XMIT ONE CHAR WITH A.E. ENABLED.
                                6. ACTIVATE THE SELECTED TRANSMITTER.
                                7. WAIT FOR "CHAR AVAIL" - IF TIMEOUT REPORT ERROR AND RESTART
                                   AT STEP 2.
                                8. IF NO TIMEOUT - READ SILO AND COMPARE AUTO ECHO DATA RECEIVED
                                   REPORT DATA COMPARE ERRORS AND RESTART AT STEP 2
                                9. IF NO ERRORS REPEAT STEPS 7 AND 8 SIXTY-FOUR TIMES THEN TURN
                                   OFF A.E. ENABLE AND READ LAST CHAR FROM SILO.. CHECK LAST
                                   CHAR FOR DATA COMPARE ERRORS - REPORT ERRORS IF ANY - AND RE-
                                   START AT STEP 2.
                               10. CHANGE A.E. TABLE POINTER TO POINT TO "AETAB0:" (0'S DATA)
                                   AND REPEAT STEPS 2 THRU 9.

                        ERRORS:
                        ********

                        1. IERROR 241 IS CALLED TO REPORT ALL ERRORS

                        SYNC:   M7277   SH4     LOAD BAR LB*HB L        CN2
                        *****

                        DEBUG:
                        ******

                        1. IF ALL LINES FAIL, SUSPECT EITHER THE M7277 OR M7289

                        2. IF ONLY ONE LINE FAILS SUSPECT THE M7288

                        3. LOOP ON THE FAILING LINE AND TRACK BACK THROUGH THE KEY LOGIC.


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

                                M7277   SH3     AE GO L         CS2
                                                74H02 "OR" GATE CHIPS E38 OR E41
                                                74157 MUX CHIPS  E39 OR 342
                                        SH4     E35 - PIN 2 STUCK LOW
```

```
                                                M7289   SH3     AE GO L          EK1
                                                                AE SCAN MUX  E22 PIN 10
                                                        SH4     SAMPLE STATUS H E21-12

                                                M7288   SH5, 7, 9, 11
                                                                AE ENABLE "NN" H CONTROL FLOPS
                                                                74174 CHIPS PIN 15

                                                        .
014710  012767  014772  164172                MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
014716  005067  164266                        CLR     $TMP3           ;INIT I/O DATA FLAG
014722  012711  004000                        MOV     #BIT11,(R1)     ;CLEAR THE DH11
014726  012705  027310                        MOV     #AETAB,R5       ;GET POINTER TO AUTO ECHO DATA TABLE
014732  005067  012250            7$:         CLR     LMSK1           ;INIT BIT TEST MARKER
014736  000261                              SEC                     ;SET "C" BIT FOR MARKER
014740  000401                              BR      13$             ;GO SHIFT MASK
014742  000241                      1$:     CLC                     ;INIT THE "C" BIT
014744  006167  012236              13$:    ROL     LMSK1           ;SHIFT BIT MARKER
014750  001407                              BEQ     12$             ;BR IF DONE ALL LINES
014752  012504                              MOV     (R5)+,R4        ;SET UP THE S/B DATA
014754  036767  012226  012220              BIT     LMSK1,LINSEL    ;TEST THIS LINE ?
014762  001767                              BEQ     1$              ;BR IF NOT
014764  004767  007512              11$:    JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
014770                                  .   12$:
014770  000522                              BR      6$              ;;BR IF ALL SELECTED LINES TESTED
014772  004767  007656              2$:     JSR     PC,CLCABC       ;GO CLEAR "CAR" AND "BCR" MEMORIES
014776  116711  013210                      MOVB    LINE,(R1)       ;SET SELECT BITS IN SCR REG
015002  012761  177777  000010              MOV     #-1,BCR(R1)     ;SET UP TO XFER ONE CHAR
015010  010561  000006                      MOV     R5,CAR(R1)      ;SET UP THE BUS ADDRESS REG
015014  162761  000002  000006              SUB     #2,CAR(R1)      ;CORRECT BUS ADDRESS
015022  012767  000100  164170              MOV     #100,$TMP7      ;COUNT 64 CHARS TO BE RECEIVED IN AUTO ECHO
015030  005067  164162                      CLR     $TMP6           ;INIT CHAR COUNTER
015034  012761  133503  000004              MOV     #133503,LPR(R1) ;SET UP LINE PARAMETER REG
015042  016761  012136  000012              MOV     LINMSK,BAR(R1)  ;ACTIVATE THE LINE

015050  012767  000002  013162              MOV     #2,TIMEA        ;INIT TIMER A
015056  005067  013160                      CLR     TIMEB           ;INIT TIMERB
015062  105711                      3$:     TSTB    (R1)            ;CHAR AVAIL SET ??
015064  100427                              BMI     4$              ;BR IF YES
015066  004767  011652                      JSR     PC,TIMEIT       ;CALL THE TIMER
015072  000773                              BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                    ;AROUND THIS BRANCH IF TIME OUT OCCURS

015074  004767  012026                      JSR     PC,SAPS         ;SAVE THE ERROR PSW
015100  005061  000004                      CLR     LPR(R1)         ;TURN OFF AUTO ECHO MODE
015104  010102                              MOV     R1,R2           ;MAKE REGADR = DEVADR
015106  011103                              MOV     (R1),R3         ;GET THE WAS DATA
015110  042703  100000                      BIC     #BIT15,R3       ;CLEAR JUNK BIT
015114  012704  000200                      MOV     #200,R4         ;SET UP S/B DATA
015120  156704  013066                      BISB    LINE,R4
015124  004767  007224                      JSR     PC,SUER2A       ;GO SET UP ERROR INFO
015130  004567  007440                      JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
015134  030212                              LINE
015136  032164                              EM24+35
015140  104024                              ERROR   24              ;DATA AVAIL FAILED TO SET ON TIME
```

```
4575  015142  000677                          BR      1$              ;GO TRY NEXT LINE
4576
4577
4578  015144  005267  164046          4$:     INC     $TMP6           ;COUNT ONE CHAR RECVD
4579  015150  016103  000002                  MOV     NRC(R1),R3      ;GET THE WAS DATA
4580  015154  020304                          CMP     R3,R4           ;WAS CHAR AUTO ECHOED CORRECTLY ?
4581  015156  001417                          BEQ     5$              ;BR IF YES
4582
4583  015160  004767  011742                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4584  015164  005061  000004                  CLR     LPR(R1)         ;DISABLE AUTO ECHO
4585  015170  010102                          MOV     R1,R2           ;SET UP REGADR
4586  015172  062702  000002                  ADD     #NRC,R2
4587  015176  004767  007152                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4588  015202  004567  007366                  JSR     R5,SUNUM        ;PUT LINE NO. IN ERROR MSG
4589  015206  030212                          LINE
4590  015210  032164                          EM24+35
4591  015212  104024                          ERROR   24              ;CHAR AUTO ECHOED INCORRECTLY
4592  015214  000652                          BR      1$              ;GO TRY NEXT LINE
4593
4594  015216  005367  163776          5$:     DEC     $TMP7           ;COUNT ONE CHAR READ OUT OF 54
4595  015222  003317                          BGT     3$              ;BR IF NOT LAST ONE
4596  015224  100646                          BMI     1$              ;;BR IF LAST ONE READ
4597  015226  042761  100000  000004          BIC     #BIT15,LPR(R1)  ;DISABLE AUTO ECHO
4598  015234  000712                          BR      3$              ;GO READ LAST CHAR
4599
4600  015236  005167  163746          6$:     COM     $TMP3           ;TOGGLE I/O FLAG
4601  015242  001406                          BEQ     TST50           ;;BR IF DONE BOTH I/O DATA
4602  015244  005067  012742                  CLR     LINE            ;INIT LINE NO TO 00
4603  015250  012705  027350                  MOV     #AETAB0,R5      ;SET POINTER TO 0'S TABLE
4604  015254  000167  177452                  JMP     7$              ;REPEAT TEST FOR ZERO PATTERNS
```

```
                                ;;*****************************************************************
                                ;*TEST 50       AUTO ECHO TEST 2 - ALL SELECTED LINES
                                ;;*****************************************************************
                                TST50:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST IS SIMILAR TO TEST 47 EXCEPT ALL SELECTED LINES OTHER
                                THAN THE A.E. TEST LINE ARE ACTIVELY TURNING APOUND A BINARY COUNT
                                TEST PATTERN IN NON-AUTO ECHO MODE AND THE A.E. TEST LINE IS TESTED
                                FOR ALL 1'S DATA ONLY.

                                ERRORS:
                                ********

                                1. [ERROR  32] IS CALLED TO REPORT A.E. TEST TIMEOUTS
                                2. [ERROR  31] IS CALLED TO REPORT ALL DATA COMPARE ERRORS

                                SYNC:   M7277   SH4     LOAD BAR LB+HB L        CN2
                                *****

                                DEBUG:
                                ******

                                        REFER TO TEST 47

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

                                        REFER TO TEST 47

                                %
015262  012767  015336  163620            MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
015270  012705  027310                    MOV     #AETAB,R5       ;SET POINTER TO A.E. TEST DATA TABLE
015274  005067  011706                    CLR     LMSK1           ;INIT BIT TEST MASK
015300  000261                            SEC                     ;GENERATER MARKER BIT IN "C"
015302  000401                            BR      12$             ;GO SHIFT MASK
015304  000241                    1$:     CLC                     ;INIT THE "C" BIT
015306  006167  011674            12$:    ROL     LMSK1           ;SHIFT TEST BIT
015312  001410                            BEQ     11$             ;BR IF TESTED ALL LINES
015314  012567  163700                    MOV     (R5)+,$TMP7     ;GET THE A.E. TEST DATA FOR THIS LINE
015320  036767  011662  011654            BIT     LMSK1,LINSEL    ;TEST THIS LINE ?
015326  001766                            BEQ     1$              ;BR IF NOT
015330  004767  007146                    JSR     PC,SELINE       ;GO SELECT A LINE
015334                            11$:
015334  000575                            BR      TST51           ;;BR IF DONE ALL SELECTED LINES

015336  016701  011630            2$:     MOV     DHADR,R1        ;RESET DEVADR IN CASE OF ERROR LOOP
015342  012711  004000                    MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
015346  004767  011414                    JSR     PC,SETALL       ;GO SET UP FOR BINARY COUNT XFER ON
                                                                  ;ALL LINES OTHER THAN THE SELECTED ONE
015352  156711  012634                    BISB    LINE,(R1)       ;SELECT THE LINE FOR A.E. TEST
015356  010561  000006                    MOV     R5,CAR(R1)      ;SET BUS ADDR TO XMIT TEST CHAR
015362  162761  000002  000006            SUB     #2,CAR(R1)      ;CORRECT THE ADDRESS
015370  012761  177777  000010            MOV     #-1,BCR(R1)     ;XMIT ONE CHAR ON THIS LINE
```

```
4661  015376  012761  133503  000004          MOV     #133503,LPR(R1) ;DO IT AT 9600 BAUD/8 BITS
4662  015404  116767  163472  163574          MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
4663  015412  042767  177400  163566          BIC     #177400,$TMP2
4664  015420  046767  011560  012022          BIC     LINMSK,LINACT   ;MAKE THIS LINE APPEAR INACTIVE
4665  015426  016761  011550  000012          MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL SELECTED TRANSMITTERS
4666
4667  015434  012767  000002  012576  21$:    MOV     #2,TIMEA        ;INIT TIMER A
4668  015442  005067  012574          3$:     CLR     TIMEB           ;INIT TIMER B
4669  015446  016103  000002          4$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
4670  015452  100414                  BMI     4$              ;BR IF YES
4671  015454  004767  011264                  JSR     PC,TIMEIT       ;CALL THE TIMER
4672  015460  000772                  BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
4673                                                          ;AROUND THIS BRANCH IF TIME OUT OCCURS
4674
4675  015462  016167  000004  163512          MOV     LPR(R1),$TMP0   ;SAVE THE CURRENT "LPR"
4676  015470  004567  007100                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4677  015474  030212                          LINE
4678  015476  032655                          EM32+35
4679  015500  104032                  ERROR   32              ;AUTO ECHO TIMEOUT
4680  015502  000700                  BR      1$              ;GO TRY NEXT LINE
4681
4682  015504  010304          4$:     MOV     R3,R4           ;EXTRACT LINE NUMBER OF RCVD CHAR
4683  015506  000304                  SWAB    R4
4684  015510  042704  177760                  BIC     #177760,R4
4685  015514  010402                  MOV     R4,R2           ;SAVE IT IN R2
4686  015516  006302                  ASL     R2              ;GENERATE TABLE INDEX IN R2
4687  015520  126704  012466                  CMPB    LINE,R4         ;IS THIS THE A.E. TEST LINE ??
4688  015524  001432                  BEQ     5$              ;BRANCH IF YES.
4689  015526  036267  030136  011446          BIT     $LNSEL(R2),LINSEL
4690  015534  001737                  BEQ     21$
4691
4692  015536  026203  036112                  CMP     RBUF(R2),R3     ;RECVD DATA CORRECT ??
4693  015542  001447                  BEQ     6$              ;BR IF IT WAS
4694
4695  015544  004767  011356                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4696  015550  010467  163440                  MOV     R4,$TMP5        ;SAVE THE LINE NUMBER
4697  015554  016204  036112                  MOV     RBUF(R2),R4     ;SET UP S/B DATA
4698  015560  062702  036112                  ADD     #RBUF,R2        ;SET UP S/B ADDRESS
4699  015564  012701  177703                  MOV     #177703,R1      ;SET UP THE WAS ADDRESS
4700  015570  004767  006560                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4701  015574  004567  006774                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4702  015600  001214                          $TMP5
4703  015602  032520                          EM31+45
4704  015604  104031                  ERROR   31              ;NON-ECHO DATA COMPARE ERROR
4705  015606  000167  177472                  JMP     1$              ;GO TRY NEXT LINE
4706
4707
4708  015612  020367  163402          5$:     CMP     R3,$TMP7        ;CHAR ECHOED OK ??
4709  015616  001427                  BEQ     7$              ;BR IF YES
4710
4711  015620  004767  011302                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4712  015624  012702  001220                  MOV     #$TMP7,R2       ;SAVE THE S/B ADDRESS
4713  015630  016704  163364                  MOV     $TMP7,R4        ;SAVE THE S/B DATA
4714  015634  012701  177703                  MOV     #177703,R1      ;SAVE THE WAS ADDRESS
4715  015640  004767  006510                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4716  015644  004567  006724                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MESSAGE
```

```
 4717  015650  030212                          LINE
 4718  015652  032520                          EM31+45
 4719  015654  104031                          ERROR   31              ;AUTO ECHO LINE DATA ERROR
 4720  015656  000167  177422                  JMP     1$              ;GO TRY NEXT LINE
 4721
 4722  015662  105262  036112          6$:     INCB    RBUF(R2)        ;GENERATE NEXT EXPECTED DATA ON THIS LINE -
 4723  015666  001362                          BNE     21$             ;BR IF ITS NOT BACK TO 000
 4724  015670  046267  027410  011552          BIC     LINBIT(R2),LINACT       ;INDICATE THIOS LINE DONE 256 BYTES
 4725  015676  005767  011546          7$:     TST     LINACT          ;ALL LINES INACTIVE
 4726  015702  001254                          BNE     21$             ;BR IF NOT
 4727  015704  042761  100000  000004          BIC     #BIT15,LPR(R1)  ;TURN OFF THE A.E. BIT
 4728  015712  105761  000017                  TSTB    SSR+1(R1)       ;SILO EMPTY ??
 4729  015716  001002                          BNE     8$              ;BR IF NOT
 4730  015720  000167  177360                  JMP     1$              ;GO TEST NEXT LINE
 4731  015724  000167  177504          8$:     JMP     21$             ;GO EMPTY IT
```

```
                                ;;*****************************************************************
                                ;*TEST 51       AUTO ECHO TEST 3 - ALL SELECTED LINES
                                ;;*****************************************************************
015730  000004                  TST51:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST IS IDENTICAL TO TEST 47 EXCEPT ALL SELECTED LINES
                                ARE ACTIVATED CONCURRENTLY RATHER THAN ONE AT A TIME AND ONLY
                                THE ALL 1'S DATA IS USED.

                                ERRORS:
                                *******

                                1. [ERROR  36] IS CALLED TO REPORT "DATA AVAIL" TIMEOUTS
                                2. [ERROR  31] IS CALLED TO REPORT A.E. DATA ERRORS

                                SYNC:   M7277   SH4 LOAD BAR LB+HB L    CN2
                                *****

                                DEBUG:
                                ******

                                        REFER TO TEST 47

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

                                        REFER TO TEST 47

                                %
015732  012767  015740  163150          MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
015740  012711  004000          1$:     MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
015744  012705  000020                  MOV     #20,R5          ;INIT COUNTER TO SET UP 16. LINES
015750  012702  027310                  MOV     #AETAB,R2       ;SET UP POINTER TO AUTO ECXHO TEST DATA
015754  012703  036052                  MOV     #RCNT,R3        ;R3 POINTS TO TABLE OF CHAR COUNTERS
015760  010261  000006          2$:     MOV     R2,CAR(R1)      ;SET UP BUS ADDRESS REG
015764  012761  177777  000010          MOV     #-1,BCR(R1)     ;SET UP BYTE COUNT REG
015772  012761  131403  000004          MOV     #131403,LPR(R1) ;SET UP LINE PARAMETERS
016000  005023                  CLR     (R3)+           ;CLEAR A COUNTER
016002  062702  000032          ADD     #2,R2           ;UPDATE POINTERS
016006  005211                  INC     (R1)            ;SELECT NEXT LINE
016010  005305                  DEC     R5              ;COUNT ONE DONE
016012  001362                  BNE     2$              ;BR TILL 16. DONE
016014  116767  163062  163164          MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
016022  042767  177400  163156          BIC     #177400,$TMP2
016030  016767  011146  011412          MOV     LINSEL,LINACT   ;SET FLAG TO INDICATE ALL 16. ACTIVE
016036  016761  011140  000012          MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL XMITTERS

016044  012767  000002  012166          MOV     #2,TIMEA        ;INIT TIMER A
016052  005067  012164                  CLR     TIMEB           ;INIT TIMERB
016056  005067  163130          3$:     CLR     $TMP4
016062  105711                  TSTB    (R1)            ;CHAR AVAIL SET YET ?
016064  100410                  BMI     4$              ;BR IF YES
016066  004767  010652                  JSR     PC,TIMEIT       ;CALL THE TIMER
```

```
 4788  016072  000771                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
 4789                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
 4790
 4791  016074  016167  000004  162100          MOV     LPR(R1),$TMP0   ;SAVE THE "LPR" REG
 4792  016102  104036                          ERROR   36              ;DATA AVAILABLE TIMEOUT
 4793  016104  000465                          BR      TST52           ;;EXIT TEST ON ERROR
 4794
 4795  016106  016103  000002          4$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
 4796  016112  010302                          MOV     R3,R2           ;BUILD AND SAVE LINE NO.
 4797  016114  000302                          SWAB    R2
 4798  016116  042702  177760                  BIC     #177760,R2
 4799  016122  010267  163070                  MOV     R2,$TMP6        ;SAVE THE LINE NO.
 4800  016126  006302                          ASL     R2              ;GENERATE TABLE OFFSET
 4801  016130  005267  163056                  INC     $TMP4
 4802  016134  022767  000101  163050          CMP     #101,$TMP4
 4803  016142  001745                          BEQ     3$
 4804  016144  036267  030136  011030          BIT     $LNSEL(R2),LINSEL  ;IS THIS ONE O  THE SELECTED LINES?
 4805  016152  001755                          BEQ     4$              ;IF NOT, THIS IS GARBAGE, SO CHECK NEXT SILO ENTRY.
 4806  016154  005262  036052                  INC     RCNT(R2)        ;COUNT THE CHARACTER
 4807  016160  020362  027310                  CMP     R3,AETAB(R2)    ;IS THE DATA CORRECT ??
 4808  016164  001420                          BEQ     5$              ;BR IF YES
 4809
 4810  016166  004767  010734                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4811  016172  016204  027310                  MOV     AETAB(R2),R4    ;GET THE S/B DATA
 4812  016176  062702  027310                  ADD     #AETAB,R2       ;GENERATE S/B ADDRESS
 4813  016202  062701  000002                  ADD     #NRC,R1         ;GENERATE THE WAS ADDRESS
 4814  016206  004767  006142                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4815  016212  004567  006356                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
 4816  016216  001216                          $TMP6
 4817  016220  032520                          EM31+45
 4818  016222  104031                          ERROR   31              ;DATA COMPARE ERROR
 4819  016224  000415                          BR      TST52           ;;EXIT TEST ON ERROR
 4820
 4821  016226  022762  000100  036052  5$:     CMP     #100,RCNT(R2)   ;DONE 64. CHARS ON THIS LINE ?
 4822  016234  001310                          BNE     3$              ;BR IF NOT
 4823  016236  016711  162754                  MOV     $TMP6,(R1)      ;SELECT LINE IN SCR REG
 4824  016242  042761  100000  000004          BIC     #BIT15,LPR(R1)  ;TURN OFF A.E. BIT
 4825  016250  046267  027410  011172          BIC     LINBIT(R2),LINACT       ;ALL LINES INACTIVE ??
 4826  016256  001277                          BNE     3$              ;BR IF NOT
```

```
                        ;;*****************************************************************
                        ;+TEST 52       BREAK BIT TEST - ALL SELECTED LINES
                        ;;*****************************************************************
016260  000004          TST52:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

                              THIS TEST VERIFIES THAT THE "BREAK" FEATURE WORKS PROPERLY
                        FOR ALL SELECTED LINES. THE TEST SEQUENCE IS AS FOLLOWS:

                                1. SET UP THE ERROR LOOP RETURN
                                2. RETRIEVE THE CORRECT S/B DATA FROM THE "BREAK" DATA
                                   TABLE AND UPDATE THE POINTER.
                                3. GO SELECT A LINE TO TEST - GO TO TEST 52 IF DONE ALL
                                   SELECTED LINES
                                4. RESET THE DH11 AND CLEAR THE "CAR" AND "BCR" MEMORIES.
                                5. PRIME SELECTED LINE TO OUTPUT TWO "NULL" CHARS TO CLEAR UART
                                6. ACTIVATE THE SELECTED LINE
                                7. WAIT FOR SILO TO RECEIVE TWO NULLS - IF TIMEOUT REPORT ERROR
                                   AND RESTART AT STEP 2
                                8. IF NO TIMEOUT CLEAR THE SELECTED DH11 AND RESELECT LINE NO.
                                9. PRIME SELECTED LINE TO OUT PUT 256. CHARS.
                                10.SET THE SELECTED LINE'S BREAK BIT
                                11.ACTIVATE THE SELECTED LINE
                                12.WAIT FOR "BAR" REG TO CLEAR -F TIMEOUT REPORT ERROR AND
                                   RESTART AT STEP 2
                                13.IF NO TIMEOUT VERIFY THAT THE SILO RECEIVED ONLY ONE CHAR-
                                   IF NOT REPORT ERROR AND RESTART AT STEP 2
                                14.IF SILO RECEIVED ONLY ONE CHAR VERIFY THAT IT WAS A "BREAK"
                                   CHAR - IF NOT REPORT ERROR - AND RESTART AT STEP 2

                        ERRORS:
                        ********

                        1. [ERROR  25] IS CALLED TO REPORT ALL ERRORS

                        SYNC:   M7277   SH4     LOAD BCR H      FU1
                        *****

                        DEBUG:
                        *******

                        1. IF ALL LINES FAILED SUSPECT THAT THE M7277 IS NOT GENERATING THE
                           BREAK CONTROL REG LOAD SIGNAL.

                        2. IF ONLY ONE LINE FAILS SL  ECT THE BREAK CONTROL LOGIC ON THE M7278

                        KEY SIGNALS:
                        ************

                               M7277   SH4     LOAD BCR H      FU1

                               M7278   SH5 THRU SH8

                                       74175 REGISTER CHIPS E51, E38, E67, E60
```

```
                                                    7400 DRIVERS    E45, E46, E75, E76

                                        ;
016262  012767  016342  162620          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
016270  012705  027512                  MOV     #BRKTAB,R5      ;SET UP POINTER TO BREAK DATA TABLE
016274  005067  010706                  CLR     LMSK1           ;INIT BIT TEST MASK
016300  000261                          SEC                     ;SET BIT MARKER IN "C"
016302  000401                          BR      12$             ;GO SHIFT MASK
016304  000241          1$:             CLC                     ;INIT THE "C" BIT
016306  006167  010674      12$:        ROL     LMSK1           ;SHIFT TEST MARKER
016312  001411                          BEQ     11$             ;BR IF ALL LINES DONE
016314  012504                          MOV     (R5)+,R4        ;GET TEST DATA FOR THIS LINE
016316  036767  010664  010656          BIT     LMSK1,LINSEL    ;LINE SELECTED ?
016324  001767                          BEQ     1$              ;BR IF NOT
016326  004767  006150                  JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
016332  000401                          BR      11$             ;;BR IF DONE ALL SELECTED LINES
016334  000402                          BR      2$              ;GO TEST THE SELECTED LINE
016336  000167  000454      11$:        JMP     9$              ;GO EXIT TEST
016342  012711  004000      2$:         MOV     #BIT11,(R1)     ;CLEAR THE DH11
016346  004767  006302                  JSR     PC,CLCABC       ;GO CLR THE "CAR" AND "BCR" MEMORIES
016352  116711  011634                  MOVB    LINE,(R1)       ;SELECT THE LINE
                                        ;
016356  012761  030246  000006          MOV     #TNULL,CAR(R1)  ;SET UP TO OUTPUT TWO NULL CHARS
016364  012761  177776  000010          MOV     #-2,BCR(R1)     ;SET BYTE COUNT TO 2
016372  012761  033503  000004          MOV     #33503,LPR(R1)  ;SET UP LINE PARAMETERS
016400  016761  010600  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE SELECTED LINE
                                        ;
016406  012767  000001  011624          MOV     #1,TIMEA        ;INIT TIMER A
016414  005067  011622                  CLR     TIMEB           ;INIT TIMER B
016420  122761  000002  000017  3$:     CMPB    #2,SSR+1(R1)    ;TWO CHARS RECEIVED ??
016426  001432                          BEQ     4$              ;BR IF YES
016430  004767  010310                  JSR     PC,TIMEIT       ;CALL THE TIMER
016434  000771                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS
                                        ;
016436  004767  010464                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
016442  010467  162536                  MOV     R4,$TMP1        ;SAVE S/B DATA
016446  010102                          MOV     R1,R2           ;SET UP REGADR
016450  062702  000016                  ADD     #SSR,R2
016454  011203                          MOV     (R2),R3         ;GET THE WAS DATA
016456  042703  100377                  BIC     #100377,R3      ;CLEAR JUNK
016462  012704  000002                  MOV     #2,R4           ;SET UP S/B DATA
016466  000304                          SWAB    R4
016470  004767  005660                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
016474  004567  006074                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MESSAGE
016500  030212                          LINE
016502  032223                          EM25+34
016504  016704  162474                  MOV     $TMP1,R4        ;RESTORE S/B DATA
016510  104025                          ERROR   25              ;TIMED OUT WAITING FOR TWO NULLS
016512  000674                          BR      1$              ;GO TRY NEXT LINE
                                        ;
016514  012711  004000      4$:         MOV     #BIT11,(R1)     ;CLEAR THE INTERFACE
016520  116711  011466                  MOVB    LINE,(R1)       ;SELECT THE LINE
016524  012761  037112  000006          MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDRESS REG FOR XMITTR
016532  012761  177400  000010          MOV     #-400,BCR(R1)   ;SET BYTE COUNT TO XMIT 256(10) CHARS
016540  012761  033503  000004          MOV     #33503,LPR(R1)  ;SET UP LINE PARAMETERS
```

```
 4939  016546  016761  010432  000014          MOV     LINMSK,BKR(R1)  ;SET BREAK BIT FOR ACTIVE LINE
 4940  016554  016761  010424  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
 4941
 4942  016562  012767  000005  011450          MOV     #5,TIMEA        ;INIT TIMER A
 4943  016570  005067  011446                  CLR     TIMEB           ;INIT TIMER B
 4944  016574  005761  000012          5$:     TST     BAR(R1)         ;BAR BIT CLEARED ??
 4945  016600  001426                  BEQ     6$              ;BR IF YES
 4946  016602  004767  010136                  JSR     PC,TIMEIT       ;CALL THE TIMER
 4947  016606  000772                  BR      5$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                               ;AROUND THIS BRANCH IF TIME OUT OCCURS
 4948
 4949  016610  004767  010312                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4950  016614  010467  162364                  MOV     R4,$TMP1        ;SAVE THE S/B DATA
 4951  016620  010102                  MOV     R1,R2           ;SET UP REGADR
 4952  016622  062702  000012                  ADD     #SSR,R2
 4953  016626  011203                  MOV     (R2),R3         ;GET THE WAS DATA
 4954  016630  005004                  CLR     R4              ;SET UP S/B DATA
 4955  016632  004767  005516                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4956  016636  004567  005732                  JSR     R5,SUNUM        ;PUT LINE NO IN MESSAGE
 4957  016642  030212                  LINE
 4958  016644  032223                  EM25+34
 4959  016646  016704  162332                  MOV     $TMP1,R4        ;RESTORE THE S/B DATA
 4960  016652  104025                  ERROR   25              ;BAR BIT FAILED TO CLEAR
 4961  016654  000613                  BR      1$              ;GO TRY NEXT LINE
 4962
 4963  016656  122761  000001  000017  6$:     CMPB    #1,SSR+1(R1)    ;ONE CHAR RECEIVED ?
 4964  016664  001430                  BEQ     7$              ;BR IF YES
 4965
 4966  016666  004767  010234                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4967  016672  010467  162306                  MOV     R4,$TMP1        ;SAVE THE S/B DATA
 4968  016676  010102                  MOV     R1,R2           ;SET UP REGADR
 4969  016700  062702  000016                  ADD     #SSR,R2
 4970  016704  011203                  MOV     (R2),R3         ;GET THE WAS DATA
 4971  016706  042703  100377                  BIC     #100377,R3      ;CLEAR JUNK
 4972  016712  012704  000001                  MOV     #1,R4           ;SET UP S/B DATA
 4973  016716  000304                  SWAB    R4
 4974  016720  004767  005430                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4975  016724  004567  005644                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MESSAGE
 4976  016730  030212                  LINE
 4977  016732  032223                  EM25+34
 4978  016734  016704  162244                  MOV     $TMP1,R4        ;RESTORE THE S/B DATA
 4979  016740  104025                  ERROR   25              ;FAILED TO RECEIVE THE ONE CHAR
 4980  016742  000167  177336                  JMP     1$              ;GO TRY NEXT LINE
 4981
 4982  016746  016103  000002          7$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
 4983  016752  020304                  CMP     R3,R4           ;WAS IT A BREAK CHAR ?
 4984  016754  001002                  BNE     8$              ;BR IF NOT CORRECT
 4985  016756  000167  177322                  JMP     1$              ;GO TEST NEXT LINE
 4986
 4987  016762  004767  010140          8$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4988  016766  010102                  MOV     R1,R2           ;SET UP REGADR
 4989  016770  062702  000002                  ADD     #NRC,R2
 4990  016774  004767  005354                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4991  017000  004567  005570                  JSR     R5,SUNUM        ;PUT LINE NO IN MESSAGE
 4992  017004  030212                  LINE
```

```
 4995  017006  032223                    .       EM25+34
 4996  017010  104025                            ERROR   25              ;INCORRECT DATA RECEIVED
 4997  017012  000766                            JMP     15              ;GO TRY NEXT LINE
 4998  017016  000240            95:             NOP                     ;EXIT THIS TEST
 4999
```

```
                        ;;*****************************************************************
                        ;*TEST 53       HALF DUPLEX TEST - ALL SELECTED LINES
                        ;;*****************************************************************
017020  000004          TST53:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        **************

                                THIS TEST VERIFIES THAT THE RECEIVERS ON ALL SELECTED LINES ARE
                        "BLINDED" WHEN THE HALF-DUPLEX MODE IS ENABLED. THE TEST
                        SEQUENCE IS AS FOLLOWS:

                                1. SET UP THE ERROR LOOP RETURN
                                2. GO SELECT A LINE NO. TO TEST
                                3. IF DONE ALL SELECTED LINES - GO TO TEST 53
                                4. RESET THE DH11 AND CLEAR THE "CAR" AND "BCR" MEMORIES
                                5. PRIME THE SELECTED DH11 TO XMIT 256. CHARS IN HALF-
                                   DUPLEX MODE.
                                6. ACTIVATE THE SELECTED LINE AND WAIT FOR THE "BAR" REG TO CLEAR
                                7. IF TIMEOUT - REPORT ERROR AND GO TO STEP 2
                                8. IF NO TIMEOUT VERIFY THE "CHAR AVAIL" DID NOT SET (RECEIVER
                                   BLINDED) - IF ERROR REPORT IT AND GO TO STEP2 - IF NO ERROR
                                    GO TO STEP 2

                        ERRORS:
                        *******

                        1. [ERROR  26] IS CALLED TO REPORT ALL ERRORS

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

                        DEBUG:
                        ******

                        1. SUSPECT EITHER THE M7289 OR THE M7288 MODULES

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

                                M7289   SH5     HALF DUPLEX <15:00> H SIGNALS
                                                END OF CHAR <15:00> SIGNALS

                                M7288   SH5     HALF DUPLEX <03:00> H SIGNALS
                                        SH7      "      "   <07:00> H SIGNALS
                                        SH9      "      "   <11:08> H SIGNALS
                                        SH11     "      "   <15:12> H SIGNALS

                        .
017022  012767  017036  162060          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
017030  004767  005446          1$:     JSR     PC,$ELINE
017034  000477                          BR      TST54           ;;BR IF ALL LINES TESTED
017036  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR THE INTERFACE
017042  004767  005606                  JSR     PC,CLCABC       ;GO CLR THE "CAR" AND "BCR" MEMORIES
017046  156711  011140                  BISB    LINE,(R1)       ;SELECT THE LINE
017052  012761  037112  000006          MOV     #TBUF,CAR(R1)   ;POINT TO XMIT BUFFER
```

```
017160  012763  012761  177400  000010          MOV     #-400,BCR(R1)   ;XMIT 256(10) CHARS
017166  012763  012761  073503  000004          MOV     #73503,LPR(R1)  ;SET UP THE LINE PARAMETERS
017174  010461  010104  000012                  MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE

017202  012767  000001  011130                  MOV     #1,TIMEA        ;INIT TIME A
017210  005067  011126                          CLR     TIMEB           ;INIT TIME B
017214  005761  000012          3$:             TST     BAR(R1)         ;WAIT FOR XMITTR TO FINISH
017220  001403                                  BEQ     4$              ;BR IF XMITTR FINISHED
017222  004767  007516                          JSR     PC,TIMEIT       ;CALL TIMER
017226  000773                                  BR      3$              ;TIMER WILL MOVE RETURN PC AROUND
                                                                        ;THIS BRANCH IF TIMEOUT OCCURS

017230  004767  000772                          JSR     PC,SAPS         ;SAVE THE ERROR PSW
017234  016103  000012                          MOV     BAR(R1),R3      ;GET THE WAS DATA
017240  010102                                  MOV     R1,R2           ;SET UP REGADR
017242  062702  000012                          ADD     #BAR,R2
017246  005004                                  CLR     R4              ;SET UP NEW S/B DATA
017250  004767  005200                          JSR     PC,SUER2A       ;GO SET UP THE ERROR INFO
017254  004567  005414                          JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
017260  030212                                  LINE
017262  032266                                  EM26+37
017264  104026                                  ERROR   26              ;BAR BIT FAILED TO CLEAR ON TIME
017266  000720                                  BR      1$              ;GO TRY NEXT LINE

017270  105711                  4$:             TSTB    (R1)            ;CHAR AVAIL SET ??
017272  100316                                  BPL     1$              ;BR IF NOT IT SHOULDN'T BE

017274  004767  007726                          JSR     PC,SAPS         ;SAVE THE ERROR PSW
017300  010102                                  MOV     R1,R2           ;SET UP REGADR
017302  011103                                  MOV     (R1),R3         ;GET WAS DATA
017304  042703  100000                          BIC     #BIT15,R3       ;CLEAR JUNK BIT
017310  116704  010776                          MOVB    LINE,R4         ;SET UP S/B DATA
017314  004767  005134                          JSR     PC,SUER2A       ;GO SETUP ERROR INFO
017320  004567  005350                          JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
017324  030212                                  LINE
017326  032266                                  EM26+37
017330  104026                                  ERROR   26              ;HALF DUPLEX FAILED TO BLIND RECVR
017332  000676                                  BR      1$              ;GO SELECT NEXT LINE
```

```
                        ;;*****************************************************************
                        ;*TEST 54       VERIFY THAT OVERRUN CAN SET PROPERLY - ALL SELECTED LINES
                        ;;*****************************************************************
017234  000004          TST54:  SCOPE
                        .REM    %
                        TEST ABSTRACT:
                        ***************

                                THIS TEST VERIFIES THAT "OVERRUN" SETS PROPERLY FOR ALL LINES
                        THAT ARE SELECTED FOR TEST WHEN THE OVERRUN CONDITION IS FORCED BY THE
                        PROGRAM. THE TEST SEQUENCE IS AS FOLLOWS:

                                1. SET UP THE ERROR LOOP RETURN
                                2. SELECT A LINE NO. TO TEST - IF DONE ALL LINES GO TO
                                   END OF PASS HANDLER.
                                3. PRIME THE SELECTED LINE TO XMIT 68. CHARS
                                4. ACTIVATE THE SELECTED LINE
                                5. WAIT FOR "XMIT DONE" TO SET - IF TIMEOUT REPORT ERROR
                                   AND RESTART AT STEP 2
                                6. IF NO TIMEOUT READ 65. CHARS FROM THE SILO AND VERIFY THAT
                                   "OVERRUN" IS SET ON THE LAST WORD READ
                                7. IF NOT REPORT ERROR AND RESTART AT STEP 2

                        ERRORS:
                        ********

                        1. [ERROR  50] IS CALLED TO REPORT "XMIT DONE " TIMEOUTS
                        2. [ERROR  56] IS CALLEDT TO REPORT "OVERRUN" ERROR

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

                        DEBUG:
                        ******

                        1. IF FAULT APPEARS ON ONLY ONE LINE SUSPECT UART MODULE
                           FOR THE APPROPRIATE LINE IN QUESTION.

                        2. IF FAULT APPEARS ON ALL LINES SUSPECT THE M7279 MODULE

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

                                M7279   SH1     MASTER OR H     E12-9
                                        SH2     MEMORY CHIP (3341) E13-11

                                M7280   SH2     UC1 OR 2 MASTER OR      EN2
                                        SH2-5   UART PIN 15 (BUF OR LINE NN)

                        %

017236  012767  017252  161644          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
017244  004767  005232          1$:     JSR     PC,$ELINE       ;GO SELECT A LINE # TO TEST
017250  000512                  1$:     BR      TST55           ;;BR IF DONE ALL SELECTED LINES
017252  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
017256  116711  010730                  MOVB    LINE,(R1)       ;SELECT THE LINE TO TEST
```

```
5150  017262  012761  037112  000006          MOV     #TBUF,CAR(R1)   ;SET UP CURRENT ADDRESS
5151  017270  012761  177674  000010          MOV     #-68.,BCR(R1)   ;SET UP BYTE COUNT REG
5152  017276  012761  033503  000004          MOV     #33503,LPR(R1)  ;DO IT AT 9600 BAUD - 8 BITS
5153  017304  016761  007674  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
5154
5155  017312  012767  000001  010720          MOV     #1,TIMEA        ;INIT TIMERS A AND B
5156  017320  005067  010716                  CLR     TIMEB
5157  017324  005711                  3$:     TST     (R1)            ;TRANSMITTER DONE ??
5158  017326  100425                          BMI     4$              ;BR IF YES
5159  017330  004767  007410                  JSR     PC,TIMEIT       ;CALL TIMER
5160  017334  000773                          BR      3$              ;BR IF NO TIMEOUT
5161
5162  017336  004767  007564                  JSR     PC,SAPS         ;GO SAVE PSW
5163  017342  011103                          MOV     (R1),R3         ;GET THE WAS DATA
5164  017344  042703  077760                  BIC     #77760,R3       ;CLEAR UNINTERESTING BITS
5165  017350  116704  010636                  MOVB    LINE,R4         ;SET UP S/B DATA
5166  017354  052704  100000                  BIS     #BIT15,R4
5167  017360  010102                          MOV     R1,R2           ;SET UP REGADR
5168  017362  004767  004766                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5169  017366  004567  005202                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE HEADER
5170  017372  030212                          LINE
5171  017374  034144                          EM50+53
5172  017376  104050                          ERROR   50              ;REPORT XMIT DONE TIME OUT
5173  017400  000721                          BR      1$              ;GO TRY NEXT LINE
5174
5175  017402  012767  000101  161574  4$:     MOV     #65.,$TMP1      ;SET UP TO READ 65. WORDS FROM SILO
5176  017410  116704  010576                  MOVB    LINE,R4         ;SET UP S/B DATA
5177  017414  000304                          SWAB    R4
5178  017416  152704  000101                  BISB    #65.,R4
5179  017422  052704  140000                  BIS     #BIT15+BIT14,R4 ;PUT IN OVERRUN AND VALID DATA BITS
5180  017426  016103  000002          5$:     MOV     NRC(R1),R3      ;GET WAS DATA FROM SILO
5181  017432  005367  161546                  DEC     $TMP1           ;COUNT ONE WORD READ
5182  017436  001373                          BNE     5$              ;BR TIL 65. READ
5183  017440  020304                          CMP     R3,R4           ;WAS DATA AND OVERRUN CORRECT ??
5184  017442  001700                          BEQ     1$              ;BR IF YES TRY NEXT SELECTED LINE
5185
5186  017444  004767  007456                  JSR     PC,SAPS         ;GO SAVE PSW
5187  017450  010102                          MOV     R1,R2           ;SET UP REGADR
5188  017452  062702  000002                  ADD     #NRC,R2
5189  017456  004767  004672                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
5190  017462  004567  005136                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG HDR
5191  017466  030212                          LINE
5192  017470  034562                          EM56+42
5193  017472  104056                          ERROR   56              ;OVERRUN OR DATA INCORRECT
5194  017474  000663                          BR      1$              ;GO TEST NEXT SELECTED LINE
5195
```

```
                                ;;*****************************************************************
                                ;*TEST 55       ABBREVIATED DM11-BB MODEM CONTROL DIAGNOSTIC
                                ;;*****************************************************************
017476  000004                  TST55:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THE FOLLOWING 4 TESTS ARE EXTRACTED FROM DZDHK DIAGNOSTIC
                                AND ARE INSERTED HERE SO THAT ALL LEVEL CONVERTERS AND CABLES CAN
                                BE CHECKED WITH JUST ONE PROGRAM (RATHER THAN TWO) USING THE H315
                                TURNAROUND CONNECTOR.

                                        THIS TEST VERIFIES THAT THE LINE ENABLE FUNCTION FLIP-FLOP
                                CAN BE SET AND CLEARED FOR THE SELECTED LINE.

                                ERRORS
                                ******

                                        IF ANY ERRORS OCCUR IN THE FOLLOWING TESTS, THEN DZDHK,
                                THE DM11-BB MODEM CONTROL DIAGNOSTIC SHOULD BE RUN IN
                                ITS ENTIRETY FOR A MORE COMPLETE DM11-BB CHECKOUT.

                                %

017500  032777  002000  161432          BIT     #BIT10,@SWR             ;CHECK DM11-BB?
017506  001002                        BNE     2$                      ;BRANCH IF YES.
017510  000167  001024                JMP     ENDA                    ;OTHERWISE, GET OUT.
017514  012700  030034          2$:   MOV     #DMADRS,R0              ;R0 POINTS TO BEGINNING OF DM ADDRESS TABLE.
017520  016767  010464  161454        MOV     DHNJM,$TMP0
017526  006367  161450                ASL     $TMP0                   ;DOUBLE TMP0
017532  066700  161444                ADD     $TMP0,R0                ;CREATE AN OFFSET.
017536  011067  010436                MOV     (R0),DHMCSR             ;MOVE THE DM ADDRESS INTO DHMCSR.
017542  016767  010432  010432        MOV     DHMCSR,DHMLSR
017550  062767  000002  010424        ADD     #2,DHMLSR               ;SAVE LINE STATUS REGISTER ADDRESS.
017556  005767  010416                TST     DHMCSR                  ;IS THERE A DM11-BB HERE?
017562  001004                        BNE     DOIT11                  ;BRANCH IF YES.
017564  104401                        TYPE
017566  036025                        MSG5                            ;NO DM11-BB FOUND.
017570  000167  000744                JMP     ENDA                    ;GET OUTTA HERE.
017574  004767  004702          DOIT11: JSR     PC,SELINE               ;GO SELECT A LINE.
017600  000470                        BR      TST56           ;;BR IF DONE ALL SELECTED LINES

017602  005077  010372          MUX11:  CLR     @DHMCSR                 ;CLEAR CONTROL STATUS REGISTER
017606  042767  000340  160162        BIC     #340,PS                 ;ENABLE INTERRUPTS
017614  116701  010372                MOVB    LINE,R1
017620  012777  002000  010352  MUX11A: MOV     #CLRMUX,@DHMCSR
017626  012702  000020                MOV     #16.,R2
017632  010177  010342                MOV     R1,@DHMCSR              ;SELECT LINE TO BE TESTED
017636  012777  000001  010336        MOV     #LINENA,@DHMLSR         ;SET LINE ENABLE FUNCTION FLIP-FLOP
017644  005077  010330                CLR     @DHMCSR
017650  005005                  MUX11B: CLR     R5
017652  017704  010324                MOV     @DHMLSR,R4              ;READ LINE STATUS REGISTER
017656  117703  010316                MOVB    @DHMCSR,R3              ;READ CONTROL STATUS REGISTER
017662  042703  177760                BIC     #177760,R3              ;CLEAR UNWANTED BITS
```

```
005252  017666  020103                          CMP     R1,R3           ;IF LINE NUMBER=SELECTED LINE NUMBER,
005253  017670  001002                          BNE     MUX11C          ;EXCEPT LINE ENABLE FUNCTION FLIP FLOP
005254  017672  012705  000001                  MOV     #LINENA,R5
005255                                                                  ;TO BE SET
005256  017676  020504                  MUX11C: CMP     R5,R4           ;COMPARE EXPECTED AND RECEIVED
005257  017700  001403                          BEQ     MUX11D          ;RESULTS
005258  017702  104401                          TYPE
005259  017704  035755                          MSG4                    ;ERROR
005260  017706  000000                          HALT
005261  017710  052777  000400  010262  MUX11D: BIS     #STEP,@DHMCSR   ;EXAMINE NEXT LINE
005262  017716  005302                          DEC     R2
005263  017720  001353                          BNE     MUX11B
005264  017722  005005                          CLR     R5
005265  017724  010177  010250          MUX11E: MOV     R1,@DHMCSR
005266  017730  010103                          MOV     R1,R3           ;SET LINE COUNTER TO SELECTED LINE
005267  017732  005077  010244                  CLR     @DHMLSR         ;CLEAR LINE ENABLE FLIP FLOP
005268  017736  105227  000000                  INCB    #0              ;DELAY FOR CABLE
005269  017742  001375                          BNE     .-4             ;DITTO
005270  017744  017704  010232                  MOV     @DHMLSR,R4      ;READ LINE STATUS REGISTER
005271  017750  005704                          TST     R4              ;WAS LINE ENABLE FUNCTION FLIP FLOP
005272  017752  001710                          BEQ     DOIT11          ;CLEARED
005273  017754  104401                          TYPE
005274  017756  035755                          MSG4                    ;ERROR
005275  017760  000000                          HALT
005276
005277
```

```
                                ;;*****************************************************************
                                ;*TEST 56       DM11-BB DIAGNOSTIC CONTINUED
                                ;;*****************************************************************
017762  000004                  TST56:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST VERIFIES THAT CLEAR TO SEND AND CARRIER ARE SET
                                IF  "LINE ENABLE" AND TERMINAL ARE SET FOR THE SELECTED LINE.

                                ERRORS:
                                ********

                                        IF ANY ERRORS OCCUR, RUN THE DM11-BB MODEM CONTROL DIAGNOSTIC.
                                 DZDHK.

                                %
017764  005077  010212          DOIT15: CLR     @DHMLSR                 ;CLEAR LINE STATUS REGISTER
017770  004767  004506                  JSR     PC,SELINE               ;GO SELECT A LINE.
017774  000467                          BR      TST57           ;;BR IF DONE ALL SELECTED LINES
017776  005077  010176          MUX15:  CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
020002  042767  000340  157766        BIC     #340,PS                 ;ENABLE INTERRUPTS
020010  116701  010176                  MOVB    LINE,R1
020014  012702  000020          MUX15A: MOV     #16.,R2                 ;16 LINES
020020  010177  010154                  MOV     R1,@DHMCSR              ;SELECT A LINE
020024  012777  000003  010150        MOV     #LINENA+TRMRDY,@DHMLSR  ;SET LINE ENABLE +TRMRDY
020032  005077  010142                  CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
020036  005005                  MUX15B: CLR     R5                      ;CLEAR EXPECTED RESULT
020040  017704  010136                  MOV     @DHMLSR,R4              ;READ LINE STATUS
020044  117703  010130                  MOVB    @DHMCSR,R3              ;READ LINE NUMBER
020050  042703  177760                  BIC     #177760,R3              ;CLEAR UNWANTED BITS
020054  020103                          CMP     R1,R3                   ;IF RECEIVED LINE=SELECTED LINE
020056  001002                          BNE     MUX15C                  ;EXPECT LINE ENABLE AND
020060  012705  000143                  MOV     #LINENA+TRMRDY+CC+CS,R5
                                                                        ;CLEAR TO SEND AND CARRIER ARE SET
020064  020405          MUX15C: CMP     R4,R5                   ;COMPARE EXPECTED AND
020066  001403                  BEQ     MUX15D                  ;RECEIVED RESULTS
020070  104401                  TYPE
020072  035755                  MSG4                            ;ERROR.
020074  000000                  HALT
020076  052777  000400  010074  MUX15D: BIS     #STEP,@DHMCSR           ;UPDATE LINE COUNTER
020104  005302                  DEC     R2                      ;CONTINUE IF ALL CHECKS
020106  001353                  BNE     MUX15B                  ;ARE NOT DONE FOR THIS LINE
020110  012705  000001                  MOV     #LINENA,R5              ;EXPECT LINE ENABLE
020114  010103          MUX15E: MOV     R1,R3                   ;ON SELECTED LINE
020116  010177  010056                  MOV     R1,@DHMCSR              ;SELECT LINE
020122  042777  000002  010052          BIC     #TRMRDY,@DHMLSR         ;CLEAR TERMINAL
020130  105227  000000                  INCB    #0                      ;DELAY FOR CABLE
020134  001375                          BNE     .-4                     ;DITTO
020136  017704  010040                  MOV     @DHMLSR,R4              ;READ LINE STATUS REGISTER
020142  020504                          CMP     R5,R4                   ;ONLY LINE ENABLE SHOULD BE
020144  001707                          BEQ     DOIT15                  ;SET ON THIS LINE
020146  104401                          TYPE
020150  035755                          MSG4                            ;ERROR.
020152  000000                          HALT
```

```
5334
5335
```

```
                                ;;*****************************************************************
                                ;*TEST 57       DM11-BB DIAGNOSTIC CONTINUED
                                ;;*****************************************************************
000EC154  000004                TST57:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ****************

                                        THIS TEST VERIFIES THAT RING IS SET IF "LINE ENABLE" AND
                                REQUEST TO SEND ARE SET FOR THE SELECTED LINE.

                                ERRORS:
                                *********

                                        IF ANY ERRORS OCCUR, RUN THE DM11-BB MODEM CONTROL DIAGNOSTIC,
                                 DZDHK.

                                %
020156  005077  010020        DOIT16: CLR     @DHMLSR                 ;CLEAR LINE STATUS REGISTER
020162  004767  004314                JSR     PC,SELINE               ;GO SELECT A LINE.
020166  000467                       BR      TST60           ;;BR IF DONE ALL SELECTED LINES
020170  005077  010004        MUX16:  CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
020174  042767  000340  157574        BIC     #340,PS                 ;ENABLE INTERRUPTS
020202  116701  010004                MOVB    LINE,R1
020206  012702  000020        MUX16A: MOV     #16.,R2                 ;16 LINES
020212  010177  007762                MOV     R1,@DHMCSR              ;SELECT A LINE
020216  012777  000005  007756        MOV     #LINENA+RS,@DHMLSR      ;SET LINE ENABLE +RS
020224  005077  007750                CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
020230  005005                MUX16B: CLR     R5                      ;CLEAR EXPECTED RESULT
020232  017704  007744                MOV     @DHMLSR,R4              ;READ LINE STATUS
020236  117703  007736                MOVB    @DHMCSR,R3              ;READ LINE NUMBER
020242  042703  177760                BIC     #177760,R3              ;CLEAR UNWANTED BITS
020246  020103                       CMP     R1,R3                   ;IF RECEIVED LINE=SELECTED LINE
020250  001002                       BNE     MUX16C                  ;EXPECT LINE ENABLE AND
020252  012705  000205                MOV     #LINENA+RS+RING,R5
                                                                    ;RING IS SET
020256  020405                MUX16C: CMP     R4,R5                   ;COMPARE EXPECTED AND
020260  001403                       BEQ     MUX16D                  ;RECEIVED RESULTS
020262  104401                       TYPE
020264  035755                       MSG4                            ;ERROR
020266  000000                       HALT
020270  052777  000400  007702  MUX16D: BIS     #STEP,@DHMCSR           ;UPDATE LINE COUNTER
020276  005302                       DEC     R2                      ;CONTINUE IF ALL CHECKS
020300  001353                       BNE     MUX16B                  ;ARE NOT DONE FOR THIS LINE
020302  012705  000001                MOV     #LINENA,R5              ;EXPECT LINE ENABLE
020306  010103                MUX16E: MOV     R1,R3                   ;ON SELECTED LINE
020310  010177  007664                MOV     R1,@DHMCSR              ;SELECT LINE
020314  042777  000004  007660        BIC     #RS,@DHMLSR             ;CLEAR REQUEST TO SEND
020322  105227  000000                INCB    #0                      ;DELAY FOR CABLE
020326  001375                       BNE     .-4                     ;DITTO
020330  017704  007646                MOV     @DHMLSR,R4              ;READ LINE STATUS REGISTER
020334  020504                       CMP     R5,R4                   ;ONLY LINE ENABLE SHOULD BE
020336  001707                       BEQ     DOIT16                  ;SET ON THIS LINE
020340  104401                       TYPE
020342  035755                       MSG4                            ;ERROR
020344  000000                       HALT
```

```
5392
5393
```

```
                                ;;*****************************************************
                                ;*TEST 60       DM11-BB DIAGNOSTIC CONTINUED
                                ;;*****************************************************
020346  000004                  TST60:  SCOPE
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST VERIFIES THAT SECONDARY RECEIVE IS SET IF "LINE
                                ENABLE" AND SECONDARY TRANSMIT ARE SET FOR THE SELECTED LINE.

                                ERRORS:
                                *********

                                        IF ANY ERRORS OCCUR, RUN THE DM11-BB MODEM CONTROL DIAGNOSTIC,
                                 DZDHK.

                                %
020350  005077  007626          DOIT17: CLR     @DHMLSR         ;CLEAR LINE STATUS REGISTER.
020354  004767  004122                  JSR     PC,SELINE                       ;GO SELECT A LINE.
020360  000467                          BR      ENDA
020362  005077  007612          MUX17:  CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
020366  042767  000340  157402          BIC     #340,PS                 ;ENABLE INTERRUPTS
020374  116701  007612                  MOVB    LINE,R1
020400  012702  000020          MUX17A: MOV     #16.,R2                 ;16 LINES
020404  010177  007570                  MOV     R1,@DHMCSR              ;SELECT A LINE
020410  012777  000011  007564          MOV     #LINENA+SECTX,@DHMLSR   ;SET LINE ENABLE +SECTX
020416  005077  007556                  CLR     @DHMCSR                 ;CLEAR CONTROL REGISTER
020422  005005                  MUX17B: CLR     R5                      ;CLEAR EXPECTED RESULT
020424  017704  007552                  MOV     @DHMLSR,R4              ;READ LINE STATUS
020430  117703  007544                  MOVB    @DHMCSR,R3              ;READ LINE NUMBER
020434  042703  177760                  BIC     #177760,R3              ;CLEAR UNWANTED BITS
020440  020103                          CMP     R1,R3                   ;IF RECEIVED LINE=SELECTED LINE
020442  001002                          BNE     MUX17C                  ;EXPECT LINE ENABLE AND
020444  012705  000031                  MOV     #LINENA+SECTX+SECRX,R5
                                                                        ;SECONDARY RECEIVE IS SET
020450  020405          MUX17C: CMP     R4,R5                   ;COMPARE EXPECTED AND
020452  001403                  BEQ     MUX17D                  ;RECEIVED RESULTS
020454  104401                  TYPE
020456  035755                  MSG4
020460  000000                  HALT
020462  052777  000400  007510  MUX17D: BIS     #STEP,@DHMCSR           ;UPDATE LINE COUNTER
020470  005302                  DEC     R2                      ;CONTINUE IF ALL CHECKS
020472  001353                  BNE     MUX17B                  ;ARE NOT DONE FOR THIS LINE
020474  012705  000001                  MOV     #LINENA,R5              ;EXPECT LINE ENABLE
020500  010103          MUX17E: MOV     R1,R3                   ;ON SELECTED LINE
020502  010177  007472                  MOV     R1,@DHMCSR              ;SELECT LINE
020506  042777  000010  007466          BIC     #SECTX,@DHMLSR          ;CLEAR SECONDARY TRANSMIT
020514  105227  000000                  INCB    #0                      ;DELAY FOR CABLE
020520  001375                          BNE     .-4                     ;DITTO
020522  017704  007454                  MOV     @DHMLSR,R4              ;READ LINE STATUS REGISTER
020526  020504                          CMP     R5,R4                   ;ONLY LINE ENABLE SHOULD BE
020530  001707                          BEQ     DOIT17                  ;SET ON THIS LINE
020532  104401                          TYPE
020534  035755                          MSG4
020536  000000                          HALT
```

```
020540 000004                      ENCA:   SCOPE
020542 012767 000240 000054                MOV     #240,$EOP       ;NOP THE SCOPE AT THE BEGINNING OF EOP
020550 005267 007434                       INC     CHNUM           ;GENERATE NEW DH11 NUMBER
020554 062767 000002 007434                ADD     #2,ADPTR        ;UPDATE THE TABLE POINTERS
020562 062767 000002 007430                ADD     #2,VCPTR
020570 062767 000002 007424                ADD     #2,BRPTR
020576 006367 006374                       ASL     SELMSK          ;SHIFT MARKER TO TEST NEXT DH11
020602 001410                              BEQ     $EOP            ;BR IF TESTED ALL SELECTED DH11'S
020604 130567 006366 006366                BIT     SELMSK,DHSEL    ;IS THIS DH11 SELECTED ?
020612 001750                              BEQ     ENCA            ;BR IF NOT
020614 005067 160262                       CLRB    $TSTNM          ;INIT TEST NUMBER
020620 000167 162104                       JMP     RSTRTA          ;GO TEST THIS DH11
                                   .SBTTL  END OF PASS ROUTINE

                                   ;;*****************************************************************
                                   ;*INCREMENT THE PASS NUMBER ($PASS)
                                   ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                   ;*IF THERES A MONITOR GO TO IT
                                   ;*IF THERE ISN'T JUMP TO START2

020624                             $EOP:
020624 000004                              SCOPE
020626 005067 160250                       CLR     $TSTNM          ;;ZERO THE TEST NUMBER
020632 005067 160364                       CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
020636 005267 160376                       INC     $PASS           ;;INCREMENT THE PASS NUMBER
020642 042767 100000 160370                BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
020650 005327                              DEC     (PC)+           ;;LOOP?
020652 000001                      $EOPCT: .WORD   1
020654 003022                              BGT     $DOAGN          ;;YES
020656 012737                              MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
020660 000001                      $ENDCT: .WORD   1
020662 020652                              $EOPCT
020664 104401 020731                       TYPE    ,$ENDMG         ;;TYPE "END PASS #"
020670 016746 163344                       MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
020674 104405                              TYPOS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
020676 104401 020726                       TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
020702 013700 000042               $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
020706 001405                              BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
020710 000005                              RESET                   ;;CLEAR THE WORLD
020712 004710                      $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
020714 000240                              NOP                     ;;SAVE ROOM
020716 000240                              NOP                     ;;FOR
020720 000240                              NOP                     ;;ACT11
020722                             $DOAGN:
020722 000137                              JMP     @(PC)+          ;;RETURN
020724 026624                      $RTNAD: .WORD   START2
020726     377     377     000     $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
020731     215 042412 042116      $ENDMG: .ASCIZ  <15><12>/END PASS #/
020736 050040 051501 020123
020744 000043

                                   .SBTTL  SCOPE HANDLER ROUTINE

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

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

000746                                  $SCOPE:
000746  104407                                 CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
000750  005067  007236                         CLR     LINE            ;INIT THE LINE NO. TO ZERO
000754  016701  006212                         MOV     DHADR,R1        ;SET UP DEVADR IN R1
000760  032777  040000  160152         1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
000766  001114                                 BNE     $OVER           ;;YES IF SW14=1
                                        ;*****START OF CODE FOR THE XOR TESTER*****
000770  000416                         $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
000772  013746  000004                         MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
000776  012737  021000  000004                 MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
001004  005737  177060                         TST     @#177060        ;;TIME OUT ON XOR?
001010  012637  000004                         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
001014  000463                                 BR      $SVLAD          ;;GO TO THE NEXT TEST
001016  022626                         5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
001020  012637  000004                         MOV     (SP)+,@#ERR.EC  ;;RESTORE THE ERROR VECTOR
001024  000423                                 BR      7$              ;;LOOP ON THE PRESENT TEST
001026                                  6$:;*****END OF CODE FOR THE XOR TESTER*****
001026  032777  000400  160104                 BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
001034  001404                                 BEQ     2$              ;;BR IF NO
001036  127767  160076  160036                 CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
001044  001465                                 BEQ     $OVER           ;;BR IF YES
001046  105767  160031                 2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
001052  001421                                 BEQ     3$              ;;BR IF NO
001054  126767  160035  160021                 CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
001062  101015                                 BHI     3$              ;;BR IF NO
001064  032777  001000  160046                 BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
001072  001404                                 BEQ     4$              ;;BR IF NO
001074  016767  160010  160004         7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
001102  000446                                 BR      $OVER
001104  105067  157773                 4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
001110  005067  160106                         CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
001114  000415                                 BR      1$              ;;ESCAPE TO THE NEXT TEST
001116  032777  004000  160014         3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
001124  001011                                 BNE     1$              ;;BR IF YES
001126  005767  160106                         TST     $PASS           ;;IF FIRST PASS OF PROGRAM
001132  001406                                 BEQ     1$              ;;       INHIBIT ITERATIONS
001134  005267  157744                         INC     $ICNT           ;;INCREMENT ITERATION COUNT
001140  026767  160056  157736                 CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
001146  002024                                 BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
001150  012767  000001  157726         1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
001156  016767  000052  160036                 MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
001164  105267  157712                 $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
001170  116767  157706  160040                 MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
001176  011667  157704                         MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
```

```
 5562  021202  011667  157702                MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 5563  021206  005067  160012                CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 5564  021212  112767  000001  157675        MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 5565  021220  016767  157656  157714 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 5566  021226  016716  157654                MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 5567  021232  000002                        RTI                     ;;FIXES PS
 5568  021234  000010                $MXCNT: 10                      ;;MAX. NUMBER OF ITERATIONS
                                    .SBTTL  ERROR HANDLER ROUTINE

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

 5601  021236                $ERROR:
 5602  021236  104407                        CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 5603  021240  105267  157637        7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 5604  021244  001775                BEQ     7$                      ;;DON'T LET THE FLAG GO TO ZERO
 5605  021246  016777  157630  157666        MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 5606  021254  005267  157632                INC     $ERTTL          ;;INC THE ERROR COUNT
 5607  021260  011667  157632                MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 5608  021264  162767  000002  157624        SUB     #2,$ERRPC
 5609  021272  117767  157620  157614        MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 5610  021300  032777  020000  157632        BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 5611  021306  001004                        BNE     20$             ;;SKIP TYPEOUTS
 5612  021310  004767  000106                JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
 5613  021314  104401  001227                TYPE    ,$CRLF
 5614  021320                        20$:
 5615  021320  122767  000001  157724        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 5616  021326  001007                        BNE     2$              ;;NO,SKIP APT ERROR REPORT
 5617  021330  116767  157560  000004        MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
 5618  021336  004767  001166                JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
 5619  021342     000                21$:     .BYTE   0
 5620  021343     000                        .BYTE   0
 5621  021344  000777                22$:     BR      22$             ;;APT ERROR LOOP
 5622  021346  005777  157566        2$:     TST     @SWR            ;;HALT ON ERROR
 5623  021352  100002                        BPL     3$              ;;SKIP IF CONTINUE
 5624  021354  000000                        HALT                    ;;HALT ON ERROR!
 5625  021356  104407                        CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 5626  021360  032777  001000  157552  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 5627  021366  001402                        BEQ     4$              ;;BR IF NO
 5628  021370  016716  157514                MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 5629  021374  005767  157624        4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 5630  021400  001402                        BEQ     5$              ;;BR IF NONE
 5631  021402  016716  157616                MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
 5632  021406                        5$:
 5633  021406  022737  020712  000042        CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
 5634  021414  001001                        BNE     6$              ;;BRANCH IF NO
 5635  021416  000000                        HALT                    ;;YES
 5636  021420                        6$:
```

```
                        RTI                             ;;RETURN
                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

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

021422          $ERRTYP:
021422  104401  001227          TYPE    $CRLF           ;;"CARRIAGE RETURN" & "LINE FEED"
021426  010046                  MOV     R0,-(SP)        ;;SAVE R0
021430  005000                  CLR     R0              ;;PICKUP THE ITEM INDEX
021432  153700  001114          BISB    @#$ITEMB,R0
021436  001004                  BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                        ;;TYPE THE PC OF THE ERROR
021440  016746  157452          MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                        ;;ERROR ADDRESS
021444  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
021446  000426                  BR      6$              ;;GET OUT
021450  005300          1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
021452  006300                  ASL     R0              ;;      WORK FOR THE ERROR TABLE
021454  006300                  ASL     R0
021456  006300                  ASL     R0
021460  062700  001356          ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
021464  012067  000004          MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
021470  001404                  BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
021472  104401                  TYPE                    ;;TYPE THE "ERROR MESSAGE"
021474  000000          2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
021476  104401  001227          TYPE    $CRLF           ;;"CARRIAGE RETURN" & "LINE FEED"
021502  012067  000004  3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
021506  001404                  BEQ     5$              ;;SKIP TYPEOUT IF 0
021510  104401                  TYPE                    ;;TYPE THE "DATA HEADER"
021512  000000          4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
021514  104401  001227          TYPE    $CRLF           ;;"CARRIAGE RETURN" & "LINE FEED"
021520  011000          5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
021522  001004                  BNE     7$              ;;GO TYPE THE DATA
021524  012600          6$:     MOV     (SP)+,R0        ;;RESTORE R0
021526  104401  001227          TYPE    $CRLF           ;;"CARRIAGE RETURN" & "LINE FEED"
021532  000207                  RTS     PC              ;;RETURN
021534          7$:
021534  013046                  MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
021536  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
021540  005710                  TST     (R0)            ;;IS THERE ANOTHER NUMBER?
021542  001770                  BEQ     6$              ;;BR IF NO
021544  104401  021552          TYPE    ,8$             ;;TYPE TWO(2) SPACES
021550  000771                  BR      7$              ;;LOOP
021552  020040      000 8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
021556                          .EVEN
                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                ;;**************************************************************
                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                ;*CALL:
                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
```

```
                                        ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                        ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                        ;*      .BYTE   M               ;;M=1 OR 0
                                        ;*                                      ;;1=TYPE LEADING ZEROS
                                        ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                        ;*
                                        ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                        ;*$TYPOS OR $TYPOC
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                                        ;*
                                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT

021556  017646  000030          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
021562  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
021570  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
021574  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
021600  000406                          BR      $TYPON
021602  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
021610  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
021616  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
021624  010346                          MOV     R3,-(SP)        ;;SAVE R3
021626  010446                          MOV     R4,-(SP)        ;;SAVE R4
021630  010546                          MOV     R5,-(SP)        ;;SAVE R5
021632  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
021636  005404                          NEG     R4
021640  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
021644  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
021650  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
021654  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
021660  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
021662  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
021664  000404                          BR      3$              ;;GO DO MSB
021666  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
021670  006105                          ROL     R5
021672  006105                          ROL     R5
021674  010503                          MOV     R5,R3
021676  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
021700  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
021704  100016                          BPL     7$              ;;BR IF NO
021706  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
021712  001002                          BNE     4$              ;;TEST FOR 0
021714  005704                          TST     R4              ;;SUPPRESS THIS 0?
021716  001403                          BEQ     5$              ;;BR IF YES
021720  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
021722  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
021726  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
021732  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
021736  104401  021776                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
021742  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
021746  003347                          BGT     2$              ;;BR IF MORE TO DO
021750  002402                          BLT     6$              ;;BR IF DONE
```

```
021752  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
021754  000744                          BR      2$              ;;GO DO THE LAST DIGIT
021756  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
021760  012604                          MOV     (SP)+,R4        ;;RESTORE R4
021762  012603                          MOV     (SP)+,R3        ;;RESTORE R3
021764  016666  000002  000004         MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
021772  012616                          MOV     (SP)+,(SP)
021774  000002                          RTI                     ;;RETURN
021776  000                     8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
021777  000                             .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
022000  000                     $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
022001  000                     $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
022002  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

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

022004                  $TYPDS:
022004  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
022006  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
022010  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
022012  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
022014  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
022016  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
022022  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
022026  100004                         BPL     1$              ;;BR IF INPUT IS POS.
022030  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
022032  112766  000055  000001         MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
022040  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
022042  012703  022220                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
022046  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
022052  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
022054  016001  022210                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
022060  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
022062  002402                         BLT     4$              ;;BR IF DONE
022064  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
022066  000774                         BR      3$
022070  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
022072  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
022074  001002                         BNE     5$              ;;FALL THROUGH IF 0
022076  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
022100  100407                         BMI     7$              ;;BR IF YES
022102  106316                  5$:     ASLB    (SP)            ;;MSD?
022104  103003                         BCC     6$              ;;BR IF NO
022106  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
022114  052702  000060         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
022120  052702  000040         7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
022124  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
```

```
 5796  022126  005720                          TST     R0          ;;JUST INCREMENTING
 5797  022130  020027  000010                  CMP     R0,#10.     ;;CHECK THE TABLE INDEX
 5798  022134  002746                          BLT     2$          ;;GO DO THE NEXT DIGIT
 5799  022136  003002                          BGT     8$          ;;GO TO EXIT
 5800  022140  010502                          MOV     R5,R2       ;;GET THE LSD
 5801  022142  000764                          BR      6$          ;;GO CHANGE TO ASCII
 5802  022144  105726                  8$:     TSTB    (SP)+       ;;WAS THE LSD THE FIRST NON-ZERO?
 5803  022146  100003                          BPL     9$          ;;BR IF NO
 5804  022150  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 5805  022156  105013                  9$:     CLRB    (R3)        ;;SET THE TERMINATOR
 5806  022160  012605                          MOV     (SP)+,R5    ;;POP STACK INTO R5
 5807  022162  012603                          MOV     (SP)+,R3    ;;POP STACK INTO R3
 5808  022164  012602                          MOV     (SP)+,R2    ;;POP STACK INTO R2
 5809  022166  012601                          MOV     (SP)+,R1    ;;POP STACK INTO R1
 5810  022170  012600                          MOV     (SP)+,R0    ;;POP STACK INTO R0
 5811  022172  104401  022220                  TYPE    ,$DBLK      ;;NOW TYPE THE NUMBER
 5812  022176  016666  000002  000004          MOV     2(SP),4(SP) ;;ADJUST THE STACK
 5813  022204  012616                          MOV     (SP)+,(SP)
 5814  022206  000002                          RTI                 ;;RETURN TO USER
 5815  022210  023420                  $DTBL:  10000.
 5816  022212  001750                          1000.
 5817  022214  000144                          100.
 5818  022216  000012                          10.
 5819  022220  000004                  $DBLK:  .BLKW   4
 5820                                  .SBTTL  TYPE ROUTINE
 5821
 5822                                  ;;**********************************************************************
 5823                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 5824                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 5825                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 5826                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 5827                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 5828                                  ;*
 5829                                  ;*CALL:
 5830                                  ;*1) USING A TRAP INSTRUCTION
 5831                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 5832                                  ;*OR
 5833                                  ;*      TYPE
 5834                                  ;*      MESADR
 5835                                  ;*
 5836
 5837  022230  105767  156723          $TYPE:  TSTB    $TPFLG      ;;IS THERE A TERMINAL?
 5838  022234  100002                          BPL     1$          ;;BR IF YES
 5839  022236  000000                          HALT                ;;HALT HERE IF NO TERMINAL
 5840  022240  000430                          BR      3$          ;;LEAVE
 5841  022242  010046                  1$:     MOV     R0,-(SP)    ;;SAVE R0
 5842  022244  017600  000002                  MOV     @2(SP),R0   ;;GET ADDRESS OF ASCIZ STRING
 5843  022250  122767  000001  156774          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 5844  022256  001011                          BNE     62$         ;;NO,GO CHECK FOR APT CONSOLE
 5845  022260  132767  000100  156765          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 5846  022266  001405                          BEQ     62$         ;;NO,GO CHECK FOR CONSOLE
 5847  022270  010067  000004                  MOV     R0,61$      ;;SETUP MESSAGE ADDRESS FOR APT
 5848  022274  004767  000220                  JSR     PC,$ATY3    ;;SPOOL MESSAGE TO APT
 5849  022300  000000                  61$:    .WORD   0           ;;MESSAGE ADDRESS
 5850  022302  132767  000040  156743  62$:    BITB    #APTCSJP,$ENVM  ;;APT CONSOLE SUPPRESSED
 5851  022310  001003                          BNE     60$         ;;YES,SKIP TYPE OUT
```

```
 5842  022312  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 5843  022314  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 5844  022316  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 5845  022320  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
 5846  022322  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 5847  022326  000002                        RTI                     ;;RETURN
 5848  022330  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 5849  022334  001430                        BEQ     8$
 5850  022336  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 5851  022342  001006                        BNE     5$
 5852  022344  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
 5853  022346  104401                        TYPE                    ;;TYPE A CR AND LF
 5854  022350  001227                        $CRLF
 5855  022352  105067  000130                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 5856  022356  000755                        BR      2$              ;;GET NEXT CHARACTER
 5857  022360  004767  000056        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 5858  022364  126726  156566        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 5859  022370  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
 5860  022372  016746  156556                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 5861                                                                ;;AND THE NULL CHAR.
 5862  022376  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 5863  022402  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 5864  022404  004767  000032                JSR     PC,$TYPEC       ;;GO TYPE A NULL
 5865  022410  105367  000072                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 5866  022414  000770                        BR      7$              ;;LOOP
 5867
 5868                                  ;HORIZONTAL TAB PROCESSOR
 5869
 5870  022416  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 5871  022422  004767  000014        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 5872  022426  132767  000007  000052        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 5873  022434  001372                        BNE     9$              ;;TAB STOP
 5874  022436  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
 5875  022440  000724                        BR      2$              ;;GET NEXT CHARACTER
 5876  022442  105777  156502        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 5877  022446  100375                        BPL     $TYPEC
 5878  022450  116677  000002  156474        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 5879  022456  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 5880  022464  001003                        BNE     1$              ;;BRANCH IF NO
 5881  022466  105067  000014                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 5882  022472  000406                        BR      $TYPEX          ;;EXIT
 5883  022474  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 5884  022502  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
 5885  022504  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
 5886  022506  000000        $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
 5887  022510  000207        $TYPEX: RTS     PC
 5888
 5889                                  .SBTTL  APT COMMUNICATIONS ROUTINE
 5890
 5891                                  ;;**************************************************************
 5892  022512  112767  000001  000236 $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
 5893  022520  112767  000001  000226 $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
 5894  022526  000403                        BR      $ATYC
 5895  022530  112767  000001  000220 $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
 5896  022536                        $ATYC:
 5897  022536  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
```

```
 5898  022540  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 5899  022542  105767  000206                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
 5900  022546  001450                          BEQ     5$              ;;IF NOT:  BR
 5901  022550  122767  000001  156474          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
 5902  022556  001031                          BNE     3$              ;;IF NOT:  BR
 5903  022560  132767  000100  156465          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
 5904  022566  001425                          BEQ     3$              ;;IF NOT:  BR
 5905  022570  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
 5906  022574  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 5907  022602  005767  156424          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
 5908  022606  001375                          BNE     1$              ;;IF NOT:  WAIT
 5909  022610  010067  156432                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
 5910  022614  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
 5911  022616  001376                          BNE     2$
 5912  022620  166700  156422                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
 5913  022624  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
 5914  022626  010067  156416                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
 5915  022632  012767  000004  156372          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
 5916  022640  000413                          BR      5$
 5917  022642  017667  000004  000016  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
 5918  022650  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
 5919  022656  016746  155114                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
 5920  022662  004767  177342                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
 5921  022666  000000                  4$:     .WORD   0
 5922  022670                          5$:
 5923  022670  105767  000062          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
 5924  022674  001416                          BEQ     12$             ;;IF NOT:  BR
 5925  022676  005767  156350                  TST     $ENV            ;;RUNNING UNDER APT?
 5926  022702  001413                          BEQ     12$             ;;IF NOT:  BR
 5927  022704  005767  156322          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
 5928  022710  001375                          BNE     11$             ;;IF NOT:  WAIT
 5929  022712  017667  000004  156314          MOV     @4(SP),$FATAL   ;;GET ERROR #
 5930  022720  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 5931  022726  005267  156300                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
 5932  022732  105067  000020          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
 5933  022736  105067  000013                  CLRB    $LFLG           ;;CLEAR LOG FLAG
 5934  022742  105067  000006                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
 5935  022746  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 5936  022750  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 5937  022752  000207                          RTS     PC              ;;RETURN
 5938  022754     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
 5939  022755     000          $LFLG:  .BYTE   0               ;;LOG FLAG
 5940  022756     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
 5941        022760                          .EVEN
 5942        000200          APTSIZE=200
 5943        000001          APTENV=001
 5944        000100          APTSPOOL=100
 5945        000040          APTCSUP=040
 5946                        .SBTTL  TTY INPUT ROUTINE
 5947
 5948                        ;;**************************************************************************
 5949                        .ENABL .LSB
 5950
 5951                        ;;**************************************************************************
 5952                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
 5953                        ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
```

```
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;*WHEN OPERATING IN TTY FLAG MODE.
023760  022767  000176  156152  $CKSWR: CMP     #$WREG,SWR      ;;IS THE SOFT-SWR SELECTED?
023766  001074                          BNE     15$             ;;BRANCH IF NO
023770  105777  156150                  TSTB    @$TKS           ;;CHAR THERE?
023774  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
023776  117746  156144                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
023002  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
023006  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
023012  001062                          BNE     15$             ;;NO, RETURN TO USER
023014  126727  156114  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
023022  001456                          BEQ     15$             ;;BRANCH IF YES

023024  104401  023633                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
023030  104401  023640          $SWSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
023034  016746  155136                  MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
023040  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
023042  104401  023651                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
023046  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
023050  005046                          CLR     -(SP)           ;;THE NEW SWR
023052  105777  156066          7$:     TSTB    @$TKS           ;;CHAR THERE?
023056  100375                          BPL     7$              ;;IF NOT TRY AGAIN

023060  117746  156062                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
023064  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII



023070  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
023074  001005                          BNE     10$             ;;BRANCH IF NOT
023076  104401  023626                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
023102  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
023106  000757                          BR      19$             ;;LET'S TRY IT AGAIN


023110  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
023114  001022                          BNE     16$             ;;BRANCH IF NO
023116  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
023122  001403                          BEQ     11$             ;;BRANCH IF YES
023124  016677  000002  156006          MOV     2(SP),@SWR      ;;SAVE NEW SWR
023132  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
023136  104401  001227          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
023142  126727  155767  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
023150  001003                          BNE     15$             ;;BRANCH IF NOT
023152  012777  000100  155764          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
023160  000002                  15$:    RTI                     ;;RETURN
023162  004767  177254          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
023166  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
023172  002420                          BLT     18$             ;;BRANCH IF YES
023174  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
023200  003015                          BGT     18$             ;;BRANCH IF YES
023202  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
023206  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
023212  001403                          BEQ     17$             ;;BRANCH IF YES
023214  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
023216  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
```

```
 6010  023220  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
 6011  023222  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
 6012  023226  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 6013  023232  000707                          BR      7$              ;;GET THE NEXT ONE
 6014  023234  104401  001226          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
 6015  023240  000720                          BR      20$             ;;SIMULATE CONTROL-U
 6016                                  .DSABL  LSB
 6017
 6018
 6019                                  ;;*****************************************************************
 6020                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 6021                                  ;*CALL:
 6022                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
 6023                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
 6024                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
 6025                                  ;
 6026
 6027  023242  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
 6028  023244  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
 6029  023252  105777  155666          1$:     TSTB    @$TKS           ;;WAIT FOR
 6030  023256  100375                          BPL     1$              ;;A CHARACTER
 6031  023260  117766  155662  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
 6032  023266  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
 6033  023274  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
 6034  023302  001013                          BNE     3$              ;;BRANCH IF NO
 6035  023304  105777  155634          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
 6036  023310  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
 6037  023312  117746  155630                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
 6038  023316  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 6039  023322  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 6040  023326  001366                          BNE     2$              ;;IF NOT DISCARD IT
 6041  023330  000750                          BR      1$              ;;YES, RESUME
 6042  023332  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
 6043  023340  002407                          BLT     4$              ;;BRANCH IF YES
 6044  023342  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
 6045  023350  003003                          BGT     4$              ;;BRANCH IF YES
 6046  023352  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
 6047  023360  000002                  4$:     RTI                     ;;GO BACK TO USER
 6048                                  ;;*****************************************************************
 6049                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
 6050                                  ;*CALL:
 6051                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
 6052                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
 6053                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
 6054
 6055  023362  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
 6056  023364  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
 6057  023366  012703  023616          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
 6058  023372  022703  023626          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
 6059  023376  101456                          BLOS    4$              ;;BR IF YES
 6060  023400  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 6061  023402  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 6062  023404  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
 6063  023410  001022                          BNE     5$              ;;BR IF NO
 6064  023412  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
 6065  023414  001007                          BNE     6$              ;;BR IF NO
```

```
 6066  023416  112767  000134  000170          MOVB    #'\,9$          ;;TYPE A BACK SLASH
 6067  023424  104401  023614                  TYPE    ,9$
 6068  023430  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
 6069  023434  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
 6070  023436  020327  023616                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
 6071  023442  103434                          BLO     4$              ;;BR IF YES
 6072  023444  111367  000144                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
 6073  023450  104401  023614                  TYPE    ,9$             ;;GO TYPE
 6074  023454  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
 6075  023456  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
 6076  023460  001406                          BEQ     7$              ;;BR IF NO
 6077  023462  112767  000134  000124          MOVB    #'\,9$          ;;TYPE A BACK SLASH
 6078  023470  104401  023614                  TYPE    ,9$
 6079  023474  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
 6080  023476  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
 6081  023502  001003                          BNE     8$              ;;BR IF NO
 6082  023504  104401  023626                  TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
 6083  023510  000726                          BR      1$              ;;GO START OVER
 6084  023512  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
 6085  023516  001011                          BNE     3$              ;;BRANCH IF NO
 6086  023520  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
 6087  023522  104401  001227                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
 6088  023526  104401  023616                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
 6089  023532  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
 6090  023534  104401  001226          4$:     TYPE    ,$QUES          ;;TYPE A '?'
 6091  023540  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 6092  023542  111367  000046          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 6093  023546  104401  023614                  TYPE    ,9$
 6094  023552  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 6095  023556  001305                          BNE     2$              ;;LOOP IF NOT RETURN
 6096  023560  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 6097  023564  104401  001230                  TYPE    ,$LF            ;;TYPE A LINE FEED
 6098  023570  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
 6099  023572  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 6100  023574  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 6101  023576  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 6102  023604  012766  023616  000004          MOV     #$TTYIN,4(SP)
 6103  023612  000002                          RTI                     ;;RETURN
 6104  023614     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 6105  023615     000                          .BYTE   0               ;;TERMINATOR
 6106  023616  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
 6107  023626  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
 6108  023633     136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
 6109  023640  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
 6110  023646  020075     000
 6111  023651     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
 6112  023656  036440  000040
 6113
 6114                                   .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
 6115
 6116                                   ;;**************************************************************
 6117                                   ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
 6118                                   ;*CHANGE IT TO BINARY.
 6119                                   ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
 6120                                   ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
 6121                                   ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
                                       ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
```

```
                                        ;*CALL:
                                        ;*      RDOCT                   ;;READ AN OCTAL NUMBER
                                        ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                                        ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT

023662  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
023664  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
023672  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
023674  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
023676  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
023700  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
023702  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
023704  010067  000100                  MOV     R0,5$           ;;AND SAVE IT.
023710  005001                          CLR     R1              ;;CLEAR DATA WORD
023712  005002                          CLR     R2
023714  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
023716  001420                          BEQ     3$              ;;IF ZERO GET OUT
023720  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
023724  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
023726  122716  000067                  CMPB    #'7,(SP)
023732  002423                          BLT     4$
023734  006301                          ASL     R1              ;;*2
023736  006102                          ROL     R2
023740  006301                          ASL     R1              ;;*4
023742  006102                          ROL     R2
023744  006301                          ASL     R1              ;;*8
023746  006102                          ROL     R2
023750  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
023754  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
023756  000756                          BR      2$              ;;LOOP
023760  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
023762  010166  000012                  MOV     R1,12.(SP)      ;;SAVE THE RESULT
023766  010267  000026                  MOV     R2,$HIOCT
023772  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
023774  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
023776  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
024000  000002                          RTI                     ;;RETURN
024002  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
024004  105010                          CLRB    (R0)            ;;SET A TERMINATOR
024006  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
024010  000000                  5$:     .WORD   0
024012  104401  001226                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
024016  000730                          BR      1$              ;;TRY AGAIN
024020  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
                                .SBTTL  TRAP DECODER

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

024022  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
024024  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
024030  005740                          TST     -(R0)           ;;BACKUP BY 2
024032  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
```

```
024034  006300                  ASL     R0              ;;POSITION FOR INDEXING
024036  016000  024056          MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
024042  000200                  RTS     R0              ;;GO TO ROUTINE


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

024044  011646          $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
024046  016666  000004  000002  MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
024054  000002                  RTI                     ;;RESTORE THE PSW

                        .SBTTL  TRAP TABLE

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

                        ;       ROUTINE
                        ;       -------
024056  024044  $TRPAD: .WORD   $TRAP2
024060  022230          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
024062  021602          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
024064  021556          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
024066  021616          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
024070  022024          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

024072  023030          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

024074  022760          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
024076  023242          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
024100  023362          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
024102  023662          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
                        .SBTTL  POWER DOWN AND UP ROUTINES

                        ;;**************************************************************
                        ;POWER DOWN ROUTINE
024104  012737  024250  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
024112  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
024120  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
024122  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
024124  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
024126  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
024130  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
024132  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
024134  017746  155000          MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
024140  010667  000110          MOV     SP,$SAVR6       ;;SAVE SP
024144  012737  024156  000024  MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
024152  000000                  HALT
024154  000776                  BR      .-2             ;;HANG UP

                        ;;**************************************************************
                        ;POWER UP ROUTINE
024156  012737  024250  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
024164  016706  000064          MOV     $SAVR6,SP       ;;GET SP
024170  005067  000060          CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
024174  005267  000054  1$:     INC     $SAVR6          ;;WAIT FOR THE INC
024200  001375                  BNE     1$              ;;OF  WORD
```

```
024202  012677  154732          MOV     (SP)+,@#SWR     ;;POP STACK INTO @#SWR
024206  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
024210  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
024212  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
024214  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
024216  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
024220  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
024222  012737  024104  000024  MOV     #$PWRDN,@#PWR.EC ;;SET UP THE POWER DOWN VECTOR
024230  012737  000340  000026  MOV     #340,@#PWR.EC+2 ;;PRIO:7
024236  104401                  TYPE                    ;;REPORT THE POWER FAILURE
024240  024256          $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
024242  012716                  MOV     (PC)+,(SP)      ;;RESTART AT RSTRTA
024244  002730          $PWRAD: .WORD   RSTRTA          ;;RESTART ADDRESS
024246  000002                  RTI
024250  000000          $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
024252  000776                  BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
024254  000000          $SAVR6: 0                       ;;PUT THE SP HERE
024256  050516  047523  042527 $POWER:  .ASCIZ  <15><12>"POWER"
024264  000122
                                .EVEN
```

```
                                        ;;*****************************************************************
                                        ;COMMON DH11 SERVICE ROUTINES
                                        ;;*****************************************************************

                                        ;THIS ROUTINE IS CALLED DURING START UP TO LOAD THE XMITTER
                                        ;OUTPUT BUFFER WITH A BINARY COUNT TEST PATTERN

024266  012701  037112          LDTBF1: MOV     #TBUF,R1        ;POINT TO START OF BUFFER
024272  005002                  CLR     R2              ;INIT DATA BYTE GENERATOR
024274  110221                  1$:     MOVB    R2,(R1)+        ;LOAD ONE CHAR
024276  005202                          INC     R2              ;GENERATE NEXT CHAR
024300  022702  000400                  CMP     #400,R2         ;LOADED 256(10) BYTES
024304  001373                          BNE     1$              ;BR IF NOT
024306  000207                          RTS     PC              ;RETURN TO START TESTING

                                        ;THIS ROUTINE SETS UP THE ERROR INFORMATION REQUIRED BY ANY TEST
                                        ;USING A "DH1" HEADER

024310  004767  002612          SUER1:  JSR     PC,SAPS         ;SAVE THE ERROR PSW
024314  116700  154562                  MOVB    $TSTNM,R0       ;SAVE THE TEST NO.
024320  010067  154636                  MOV     R0,$REG0        ;SAVE THE TEST NO. FOR ERROR PRINT
024324  010167  154634                  MOV     R1,$REG1        ;SAVE THE DH11 ADDR
024330  010267  154632                  MOV     R2,$REG2        ;SAVE THE REG ADDRESS
024334  010667  154636                  MOV     R6,$REG6        ;SAVE THE SP
024340  062767  000002  154630          ADD     #2,$REG6        ;CORRECT FOR CALLING JSR
024346  000207                          RTS     PC              ;RETURN TO CALLING ROUTINE

                                        ;THIS ROUTINE IS CALLED BY THOSE TESTS USING A "DH2" HEADER TO
                                        ;SAVE THE ERROR INFORMATION IN "DT2"

024350  004767  002552          SUER2:  JSR     PC,SAPS         ;SAVE THE ERROR PSW
024354  116700  154522          SUER2A: MOVB    $TSTNM,R0       ;GET THE TEST NO.
024360  010067  154576                  MOV     R0,$REG0        ;SAVE THE REGISTERS-TEST#
024364  010167  154574                  MOV     R1,$REG1        ;SAVE THE DH ADDRESS
024370  010267  154572                  MOV     R2,$REG2        ;SAVE THE REGISTER ADDRESS
024374  010367  154570                  MOV     R3,$REG3        ;SAVE THE WAS DATA
024400  010467  154566                  MOV     R4,$REG4        ;SAVE THE S B DATA
024404  010667  154566                  MOV     R6,$REG6        ;SAVE THE STACK POINTER
024410  062767  000002  154560          ADD     #2,$REG6        ;CORRECT FOR CALLING JSR
024416  000207                          RTS     PC              ;RETURN TO REPORT ERROR

                                        ;THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
                                        ;BUS ERROR AND RSVD INSTR ERROR ROUTINES

024420  010067  154536          SUER3:  MOV     R0,$REG0        ;SAVE THE REGS
024424  010167  154534                  MOV     R1,$REG1
024430  010267  154532                  MOV     R2,$REG2
024434  000207                          RTS     PC              ;RETURN TO REPORT ERROR

                                        ;THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
                                        ;CAR/BCR MEMORY PATTERNS TESTS

024436  005067  154540          SUER4:  CLR     $TMP0           ;SAVE THE LINE NO. WRITTEN
024442  116767  003546  154532          MOVB    LINEA,$TMP0
024450  116700  154426                  MOVB    $TSTNM,R0       ;SAVE THE TEST NUMBER
024454  010067  154502                  MOV     R0,$REG0        ;SAVE THE REGISTER INFORMATION
```

```
  6310  024460  010167  154500            MOV     R1,$REG1
  6311  024464  010267  154476            MOV     R2,$REG2
  6312  024470  010367  154474            MOV     R3,$REG3
  6313  024474  010467  154472            MOV     R4,$REG4
  6314  024500  000207                     RTS     PC                  ;RETURN TO PATTERNS TEST
  6315
  6316                                ;THIS ROUTINE IS CALLED TO SELECT A NEW LINE NO. BASED ON THE
  6317                                ;VALUE OF THE LINE SELECTION PARAMETER
  6318
  6319                                        ;CALLING SEQUENCE:
  6320
  6321                                        ;JSR    PC,SELINE           ;CALL THE ROUTINE
  6322                                        ;BR     1$                  ;EXIT BRANCH-ROUTINE MOVES THE RETURN
  6323                                                                    ;PC AROUND THIS BR IF MORE LINES ARE
  6324                                                                    ;YET TO BE TESTED
  6325
  6326  024502  105767  003505           SELINE: TSTB    LINE+1              ;FIRST TIME THROUGH FOR ANY TEST ?
  6327  024506  001010                     BNE     1$                  ;BR IF NOT
  6328  024510  105167  003477            COMB    LINE+1              ;SET ENTRY FLAG
  6329  024514  012767  000001  002462     MOV     #1,LINMSK           ;INIT SELECT TEST MASK TO TEST LINE 00
  6330  024522  105067  003464            CLRB    LINE                ;START WITH LINE #00
  6331  024526  000405                     BR      2$                  ;GO TEST FOR LINE #00
  6332  024530  105267  003456           1$:      INCB    LINE                ;GENERATE NEW LINE NO.
  6333  024534  006367  002444            ASL     LINMSK              ;SHIFT SELECT MASK TO TEST NXT LINE
  6334  024540  001407                     BEQ     3$                  ;RETURN TO EXIT BRANCH - ALL LINES DONE
  6335  024542  036767  002436  002432   2$:      BIT     LINMSK,LINSEL       ;IS THE LINE SELECTED FOR TEST ??
  6336  024550  001767                     BEQ     1$                  ;BR IF NOT
  6337  024552  062716  000002            ADD     #2,(SP)             ;MOVE RETURN PC AROUND EXIT BRANCH
  6338  024556  000402                     BR      4$                  ;RETURN TO TEST SELECTED LINE
  6339  024560  005067  003426           3$:      CLR     LINE                ;INIT ENTRY FLAG AND LINE NO. TO 000
  6340  024564  142777  000017  002400   4$:      BICB    #17,@DHADR          ;INIT LINE SELECT BITS IN "SCR"
  6341  024572  000207                     RTS     PC                  ;RETURN TO CALLING TEST
  6342
  6343                                ;THIS ROUTINE IS CALLED TO CONVERT EITHER THE "DH" NUMBER OR THE
  6344                                ;"LINE" NUMBER TO TWO ASCII CHARACTERS AND MOVE THEM INTO A
  6345                                ;PARTICULAR MESSAGE BUFFER FOR ERROR REPORTING
  6346
  6347                                        ;CALLING SEQUENCE
  6348
  6349                                        ;JSR    R5,SUNUM            ;CALL TO THIS ROUTINE
  6350                                        ;ADDR1                      ;ADDRESS OF THE NUMBER TO BE CONVERTED
  6351                                        ;ADDR2                      ;ADDRESS OF THE MSG BUFFER SLOT
  6352
  6353  024574                          SUNUM:
  6354  024574  010046                     MOV     R0,-(SP)            ;;PUSH R0 ON STACK
  6355  024576  010146                     MOV     R1,-(SP)            ;;PUSH R1 ON STACK
  6356  024600  010246                     MOV     R2,-(SP)            ;;PUSH R2 ON STACK
  6357  024602  012500                     MOV     (R5)+,R0            ;GET ADDRESS OF NUMBER
  6358  024604  012501                     MOV     (R5)+,R1            ;GET MSG BUFFER ADDR
  6359  024606  111000                     MOVB    (R0),R0             ;GET NO. TO BE CONVERTED
  6360  024610  010002                     MOV     R0,R2               ;SAVE IT IN R2
  6361  024612  006202                     ASR     R2                  ;SHIFT MSD TO LSD POSITION
  6362  024614  006202                     ASR     R2
  6363  024616  006202                     ASR     R2
  6364  024620  042702  177770            BIC     #177770,R2          ;CLR JUNK BITS
  6365  024624  062702  000060            ADD     #60,R2              ;MAKE IT ASCII
```

```
 6366  024630  110321                        MOVB    R2,(R1)+        ;PUT IT IN MSG BUFFER
 6367  024632  042700  177770                BIC     #177770,R0      ;CLR JUNK FROM LSD
 6368  024636  062700  000060                ADD     #60,R0          ;MAKE IT ASCII
 6369  024642  110011                        MOVB    R0,(R1)         ;PUT LSD IN THE BUFFER
 6370  024644  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 6371  024646  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 6372  024650  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 6373  024652  000205                        RTS     R5              ;RETURN TO CALLER
 6374
 6375                          ;THIS ROUTINE IS CALLED TO CLEAR THE "CAR" AND "BCR" MEMORIES
 6376                          ;IT ASSUMES THAT THE ADDRESS OF THE "SCR" IS IN R1
 6377
 6378  024654  005067  154340  CLCABC: CLR     $TMP7           ;INIT A COUNTER
 6379  024660  116711  154334  1$:     MOVB    $TMP7,(R1)      ;SELECT A LINE
 6380  024664  005061  000006          CLR     CAR(R1)         ;CLEAR A CAR LOCATION
 6381  024670  005061  000010          CLR     BCR(R1)         ;CLEAR A BCR LOCATION
 6382  024674  005267  154320          INC     $TMP7           ;GENERATE NEW LINE NO.
 6383  024700  022767  000020  154312  CMP     #20,$TMP7       ;DONE ALL LINES ?
 6384  024706  001364                  BNE     1$              ;BR IF NOT
 6385  024710  142711  000017          BICB    #17,(R1)        ;SET "SCR" TO SELECT LINE 00
 6386  024714  000207                  RTS     PC              ;RETURN TO CALLER
 6387
 6388                          ;THIS ROUTINE IS CALLED TO LOAD THE "BCR" MEMORY WITH ALL ONES
 6389                          ;IT ASSUMES THAT THE ADDRESS OF THE SCR IS IN R1
 6390
 6391  024716  005067  154276  LDBCR:  CLR     $TMP7           ;INIT A COUNTER
 6392  024722  116711  154272  1$:     MOVB    $TMP7,(R1)      ;SELECT A LINE
 6393  024726  012761  177777  000010          MOV     #-1,BCR(R1)     ;LOAD BCR LOC. WITH 177777
 6394  024734  005267  154260          INC     $TMP7           ;GENERATE NEXT LINE NO.
 6395  024740  022767  000020  154252  CMP     #20,$TMP7       ;DONE ALL LINES ?
 6396  024746  001365                  BNE     1$              ;BR IF NOT
 6397  024750  142711  000017          BICB    #17,(R1)        ;SET "SCR" TO SELECT LINE 00
 6398  024754  000207                  RTS     PC              ;RETURN TO CALLER
 6399
 6400                          ;THIS ROUTINE CALLED TO SET UP FOR PARITY TESTS
 6401
 6402  024756  012767  000020  154226  SUPPAR: MOV     #20,$TMP4       ;SET UP FOR 16. LINES
 6403  024764  105011                  CLRB    (R1)            ;INIT SCR TO START AT LINE 00
 6404  024766  005002                  CLR     R2              ;INIT INDEX REGISTER FOR RBUF (EVEN)
 6405  024770  012703  000200          MOV     #200,R3         ;SET UP CONSTANT
 6406  024774  012704  000001          MOV     #1,R4           ;INIT INDEX REG FOR RBUF (ODD)
 6407  025000  012761  037112  000006  1$:     MOV     #TBUF,CAR(R1)   ;LOAD BUS ADDRESS REWG
 6408  025006  016761  154204  000010          MOV     $TMP6,BCR(R1)   ;LOAD BYTE COUNT REG
 6409  025014  016761  154174  000004          MOV     $TMP5,LPR(R1)   ;LOAD LINE PARAMETERS
 6410  025022  105062  036112                  CLRB    RBUF(R2)        ;INIT DATA BYTE IN RBUF TO START AT 000
 6411  025026  110364  036112                  MOVB    R3,RBUF(R4)     ;SET CONSTANT IN HIGH BYTE
 6412  025032  005211                  INC     (R1)            ;SELECT NEXT LINE
 6413  025034  005203                  INC     R3              ;GENERATE NEW CONSTANT
 6414  025036  062702  000002          ADD     #2,R2           ;UPDATE POINTERS TO RBUF (EVEN,ODD)
 6415  025042  062704  000002          ADD     #2,R4
 6416  025046  005367  154140          DEC     $TMP4           ;COUNT ONE LINE SETUP
 6417  025052  001352                  BNE     1$              ;BR TILL ALL 16. SET UP
 6418  025054  012704  027452          MOV     #MULPTB,R4      ;SET UP TABLE POINTER
 6419  025060  016724  154132  2$:     MOV     $TMP6,(R4)+     ;SET UP BYTE COUNT ENTRY
 6420  025064  022704  027512          CMP     #MULPTB+40,R4   ;SET UP ALL COUNTS ?
 6421  025070  001373                  BNE     2$              ;BR IF NOT
```

```
025072  105011                          CLRB    (R1)            ;INIT SCR TO SELECT LINE 00
025074  000207                          RTS     PC              ;RETURN TO PARITY TEST

                        ;THIS ROUTINE AUTOSIZES THE SYSTEM TO DETERMINE THE ADDRESSES AND
                        ;VECTORS OF THE DH11'S AND DM11-BB'S.

025076  010046          AUTOSZ: MOV     R0,-(SP)
025100  005003                  CLR     R3
025102  012702  027732          MOV     #DHADRS,R2      ;POINT TO BEGINNING OF TABLE
025106  005022          25$:    CLR     (R2)+           ;CLEAR AUTOSIZER TABLES.
025110  005203                  INC     R3
025112  020327  000102          CMP     R3,#102         ;HAVE WE CLEARED ALL ENTRIES?
025116  001373                  BNE     25$             ;BRANCH IF NOT.
025120  013746  000004          MOV     @#4,-(SP)       ;SAVE TRAP VECTOR.
025124  012737  025232  000004  MOV     #4$,@#4         ;SETUP FOR NON-EXISTENT MEMORY TRAP.
025132  012703  030034          MOV     #DMADRS,R3      ;SETUP DM ADDRESS TABLE POINTER.
025136  012702  027732          MOV     #DHADRS,R2      ;SET UP DH ADDRESS TABLE POINTER.

025142  012701  160020          MOV     #160020,R1      ;R1=FIRST ADDRESS TO BE TESTED.

025146  005711          1$:     TST     (R1)            ;SEE IF ADDRESS IN R1 RESPONDS.
025150  005761  000016          TST     16(R1)          ;CHECK TO SEE IF DEVICE IS MODULO 20.
025154  052711  004000          BIS     #4000,(R1)      ;IF IT IS, CONTINUE
                                                        ;AND CHECK TO SEE
025160  052711  001000          BIS     #1000,(R1)      ;IF THIS ADDRESS CONTAINS
025164  052711  002000          BIS     #2000,(R1)      ;A DH-11.
025170  032711  003000          BIT     #3000,(R1)      ;CHECK TO INSURE THESE BITS SET.
025174  001410                  BEQ     3$              ;IF NOT, BRANCH.
                                                        ;SET THE MAINTENANCE BIT, THE NON-
025176  052711  000400          BIS     #400,(R1)       ;EXISTENT MEMORY BIT AND THE CLEAR
                                                        ;NON-EXISTENT MEMORY INTERRUPT BIT.
025202  032711  002400          BIT     #2400,(R1)      ;IS THIS A DH-11? (BITS 8 AND 10 SHOULD
                                                        ;CLEAR IF THIS IS A DH11.)

025206  001003                  BNE     3$              ;IF NOT, CHECK TO SEE IF THIS IS A DM11-BB.
025210  042711  001000          BIC     #1000,(R1)      ;CLEAR MAINTENANCE BIT.
025214  010122                  MOV     R1,(R2)+        ;SAVE THE ADDRESS IN THE DH ADR TABLE.

025216  020127  163760  ←      3$:     CMP     R1,#163760      ;HAVE WE REACHED THE TOP OF THE FLOATING ADDRESSES.
025222  001406                  BEQ     5$              ;IF YES, GET OUT.
025224  062701  000020          ADD     #20,R1          ;IF NOT, UPDATE ADDRESS AND
025230  000746                  BR      1$              ;GO CHECK IT.

025232  012716  025216  ←      4$:     MOV     #3$,(SP)        ;IF DH ADDRESS DOES NOT RESPOND, GO TO 3$.
025236  000002                  RTI

                        ;TEST FOR DM11 BB ADDRESS

025240  012737  025272  000004  5$:     MOV     #6$,@#4         ;SETUP FOR NON-EXISTENT MEMORY TRAP.
025246  012701  170500          MOV     #170500,R1      ;R1=FIRST ADDRESS TO BE TESTED.
025252  005711          21$:    TST     (R1)            ;SEE IF ADDRESS RESPONDS.
025254  010123                  MOV     R1,(R3)+        ;IF IT DOES, THIS IS A DM11-BB.
                                                        ;SO SAVE THE ADDRESS.
025256  020127  170670  ←      23$:    CMP     R1,#170670      ;HAVE WE REACHED THE TOP OF THE DM11 ADDRESSES?
025262  001406                  BEQ     22$             ;IF YES, GET OUT.
```

```
 6478  025264  062701  000010                    ADD     #10,R1          ;IF NOT, UPDATE ADDRESS AND
 6479  025270  000770                            BR      21$             ;GO CHECK IT.
 6480
 6481  025272  012716  025256 *          6$:     MOV     #23$,(SP)       ;IF DM ADDRESS DOES NOT RESPOND, GO TO 23$.
 6482  025276  000002                            RTI
 6483
 6484  025300  012637  000004          22$:    MOV     (SP)+,@#4       ;RESTORE TRAP VECTOR.
 6485  025304  162702  027732                    SUB     #DHADRS,R2      ;HAVE WE FOUND ANY DH11'S AT ALL?
 6486  025310  001003                            BNE     7$              ;IF YES, BRANCH
 6487  025312  104401  035451                    TYPE    .MSG1           ;NO DH11'S WERE FOUND.
 6488  025316  000000                            HALT
 6489
 6490  025320  006202                  7$:     ASR     R2              ;R2 NOW CONTAINS THE NUMBER
 6491  025322  005000                            CLR     R0              ;OF DH'S FOUND.
 6492  025324  006100                  8$:     ROL     R0              ;FILL R0 WITH 1'S
 6493  025326  005200                            INC     R0              ;CORRESPONDING TO
 6494  025330  005302                            DEC     R2              ;THE NUMBER OF DH'S
 6495  025332  005702                            TST     R2              ;FOUND.
 6496  025334  001373                            BNE     8$
 6497  025336  010067  002642                    MOV     R0,$DHSEL       ;$DHSEL CONTAINS THE DH SELECTION PARAMETER.
 6498                                                                    ;IE. ALL DH'S FOUND WILL BE TESTED.
 6499
 6500                                  ;FIND DH VECTOR:
 6501  025342  012702  027732                    MOV     #DHADRS,R2      ;SETUP POINTER TO BEGINNING OF DH
 6502  025346  012705  027774                    MOV     #DHVEC,R5       ;ADDRESS TABLE AND VECTOR TABLE.
 6503  025352  012737  000340  000022            MOV     #340,@#IOTVEC+2 ;SET IOT TRAP PRIORITY TO 7.
 6504  025360  012737  025470  000020            MOV     #12$,@#IOTVEC   ;SETUP IOT TRAP VECTOR.
 6505  025366  012703  000300                    MOV     #300,R3         ;START OF FLOATING VECTORS
 6506  025372  012704  000302                    MOV     #302,R4         ;PC OF IOT INSTR.
 6507
 6508  025376  010423                  9$:     MOV     R4,(R3)+        ;FILL VECTOR AREA WITH ADDRESS
 6509                                                                    ;OF NEXT INSTR (.+2)
 6510  025400  012724  000004                    MOV     #4,(R4)+        ;NEXT INSTRUCTION IS AN IOT TRAP.
 6511  025404  022324                            CMP     (R3)+,(R4)+     ;UPDATE R3+R4.
 6512  025406  020427  001000                    CMP     R4,#1000        ;HAVE WE REACHED TO TOP OF THE
 6513                                                                    ;VECTOR SPACE?
 6514  025412  101771                            BLOS    9$              ;IF NOT, REPEAT PROCESS.
 6515
 6516  025414  005712                  10$:    TST     (R2)            ;HAVE WE CHECK ALL DH'S?
 6517  025416  001441                            BEQ     13$             ;IF YES, GET OUT + CHECK FOR DM11 BB'S VECTORS.
 6518
 6519  025420  005067  152352                    CLR     PS              ;ZERO CPU PRIORITY.
 6520  025424  052772  001000  000000            BIS     #1000,@(R2)     ;SET MAINTENANCE BIT
 6521  025432  052772  000300  000000            BIS     #300,@(R2)      ;ATTEMPT TO CAUSE RECEIVER
 6522                                                                    ;INTERRUPT.
 6523  025440  005000                            CLR     R0
 6524
 6525  025442  005200                  11$:    INC     R0              ;WAIT...
 6526  025444  001376                            BNE     11$
 6527  025446  104401  035501                    TYPE    .MSG2           ;ERROR MSG-NO DH RECEIVER INTERRUPT OCCURRED.
 6528  025452  052772  004000  000000            BIS     #4000,@(R2)     ;DO A MASTER CLEAR
 6529  025460  042772  001000  000000            BIC     #1000,@(R2)     ;CLEAR MAINTENANCE BIT
 6530  025466  000752                            BR      10$
 6531
 6532  025470  011601                  12$:    MOV     (SP),R1
 6533  025472  042701  000007                    BIC     #7,R1           ;CLEAR GARBAGE.
```

```
 6534  025476  010125                          MOV     R1,(R5)+        ;SAVE VECTOR ADDRESS.
 6535  025500  022626                          CMP     (SP)+,(SP)+     ;POP STACK
 6536  025502  012716  025414                  MOV     #10$,(SP)       ;SETUP FOR RETURN.
 6537  025506  052772  004000  000000          BIS     #4000,@(R2)     ;DO A MASTER CLEAR
 6538  025514  042732  001000                  BIC     #1000,@(R2)+    ;CLEAR MAINTENANCE BIT.
 6539  025520  000002                          RTI
 6540
 6541                                  ;FIND DM11 BB VECTORS:
 6542
 6543  025522  012702  030034          13$:    MOV     #DMADRS,R2      ;SET POINTERS TO BEGINNING OF
 6544  025526  012705  030076                  MOV     #DMVEC,R5       ;ADR TABLE & VECTOR TABLE.
 6545  025532  012737  025614  000020          MOV     #16$,@#IOTVEC   ;SET IOT TRAP VECTOR.
 6546
 6547  025540  005712                  14$:    TST     (R2)            ;HAVE WE CHECKED ALL DM'S?
 6548  025542  001441                          BEQ     17$             ;IF YES, GET OUT.
 6549  025544  005067  152226                  CLR     PS              ;ZERO CPU PRIORITY
 6550  025550  052772  001000  000000          BIS     #1000,@(R2)     ;SET MAINTENANCE BIT.
 6551  025556  052772  000300  000000          BIS     #300,@(R2)      ;ATTEMPT TO CAUSE INTERRUPT.
 6552  025564  005000                          CLR     R0
 6553
 6554  025566  005200                  15$:    INC     R0              ;WAIT....
 6555  025570  001376                          BNE     15$
 6556  025572  104401  035546                  TYPE    ,MSG3           ;ERROR MSG - NO DM11-BB INTERRUPT OCCURRED.
 6557  025576  052772  004000  000000          BIS     #4000,@(R2)     ;CLEAR BITS PREVIOUSLY SET.
 6558  025604  042772  001000  000000          BIC     #1000,@(R2)     ;CLEAR MAINTENANCE BIT.
 6559  025612  000752                          BR      14$
 6560
 6561  025614  011601                  16$:    MOV     (SP),R1
 6562  025516  162701  000004                  SUB     #4,R1           ;CALCULATE VECTOR ADDRESS.
 6563  025622  010125                          MOV     R1,(R5)+        ;SAVE VECTOR ADDRESS.
 6564  025624  022626                          CMP     (SP)+,(SP)+     ;POP STACK.
 6565  025626  012716  025540                  MOV     #14$,(SP)       ;SETUP FOR RETURN.
 6566  025632  052772  004000  000000          BIS     #4000,@(R2)     ;CLEAR BITS PREVIOUSLY SET.
 6567  025640  042732  001000                  BIC     #1000,@(R2)+    ;CLEAR MAINTENANCE BIT AND
 6568                                                                  ;POINT TO NEXT DM11-BB ADDRESS.
 6569  025644  000002                          RTI
 6570
 6571  025646  012737  020746  000020  17$:    MOV     #$SCOPE,@#IOTVEC  ;RESTORE IOT VECTOR FOR SCOPE ROUTINE.
 6572  025654  012600                          MOV     (SP)+,R0        ;RESTORE R0.
 6573  025656  012703  000300                  MOV     #300,R3         ;START OF FLOATING VECTORS.
 6574  025662  012704  000302                  MOV     #302,R4
 6575
 6576  025666  010423                  18$:    MOV     R4,(R3)+        ;FILL VECTOR AREA WITH ADDRESS OF NEXT
 6577                                                                  ;INSTRUCTION (.+2).
 6578  025670  012724  000000                  MOV     #0,(R4)+        ;NEXT INSTRUCTION IS A HALT.
 6579  025674  022324                          CMP     (R3)+,(R4)+     ;UPDATE R3 & R4.
 6580  025676  020427  001000                  CMP     R4,#1000        ;ARE WE DONE?
 6581  025702  101771                          BLOS    18$             ;IF NOT, REPEAT UNTIL ADDRESSES
 6582                                                                  ;377 TO 777 ARE DONE.
 6583  025704  013701  027774                  MOV     @#DHVEC,R1      ;LET R1 POINT TO 1ST DH VECTOR ADDRESS.
 6584  025710  005737  027776                  TST     @#DHVEC+2       ;IS THERE MORE THAN ONE VECTOR?
 6585  025714  001403                          BEQ     26$             ;BRANCH IF NOT.
 6586  025716  163701  027776                  SUB     @#DHVEC+2,R1    ;DETERMINE NUMBER OF ADDRESSES
 6587                                                                  ;BETWEEN DH VECTORS (10(8) OR 20(8)).
 6588  025722  005401                          NEG     R1              ;MAKE R1 POSITIVE.
 6589  025724  010167  002246          26$:    MOV     R1,ADRVEC       ;SAVE THAT NUMBER.
```

```
 6590  025730  032777  000002  153202          BIT     #BIT1,@SWR      ;SHOULD DEVICE MAP BE TYPED OUT?
 6591  025736  001442                          BEQ     20$             ;IF NOT, RETURN.
 6592  025740  104401                          TYPE                    ;TYPEOUT MAP OF DH & DM11-BB'S
 6593  025742  035613                          DEVMAP                  ;FOUND.
 6594  025744  012701  027732                  MOV     #DHADRS,R1      ;R1=BEGINNING OF DH ADDRESS TABLE.
 6595  025750  012702  027774                  MOV     #DHVEC,R2       ;R2=BEGINNING OF DH VECTOR TABLE.
 6596  025754  012703  030034                  MOV     #DMADRS,R3      ;R3=BEGINNING OF DM11-BB ADDRESS TABLE.
 6597  025760  012704  030076                  MOV     #DMVEC,R4       ;R4=BEGINNING OF DM11-BB VECTOR TABLE.
 6598
 6599  025764  012146                  19$:    MOV     (R1)+,-(SP)     ;MOVE DATA TO BE TYPED
 6600  025766  104403                          TYPOS                   ;TYPE DATA
 6601  025770     006                          .BYTE   6
 6602  025771     001                          .BYTE   1
 6603  025772  012246                          MOV     (R2)+,-(SP)     ;MOVE DATA TO BE TYPED
 6604  025774  104403                          TYPOS                   ;TYPE DATA
 6605  025776     005                          .BYTE   5
 6606  025777     000                          .BYTE   0
 6607  026000  104401  035607                  TYPE    ,SPACE
 6608  026004  012346                          MOV     (R3)+,-(SP)     ;MOVE DATA TO BE TYPED.
 6609  026006  104403                          TYPOS                   ;TYPE DATA.
 6610  026010     006                          .BYTE   6
 6611  026011     001                          .BYTE   1
 6612  026012  104401  035607                  TYPE    ,SPACE
 6613  026016  012446                          MOV     (R4)+,-(SP)     ;MOVE DATA TO BE TYPED.
 6614  026020  104403                          TYPOS                   ;TYPE DATA.
 6615  026022     005                          .BYTE   5
 6616  026023     000                          .BYTE   0
 6617  026024  104401                          TYPE                    ;TYPE A CARRIAGE RETURN & LINE FEED.
 6618  026026  001227                          $CRLF
 6619  026030  005711                          TST     (R1)            ;HAVE WE TYPED ALL DH ENTRIES?
 6620  026032  001354                          BNE     19$             ;IF NOT, DO IT AGAIN.
 6621  026034  005713                          TST     (R3)            ;HAVE WE TYPED ALL DM ENTRIES?
 6622  026036  001352                          BNE     19$             ;IF NOT - ONE MORE TIME.
 6623  026040  104401  001227                  TYPE    ,$CRLF
 6624  026044  000207                  20$:    RTS     PC              ;IF YES, GO BACK TO MAIN PROGRAM.
 6625
 6626                                          ;THIS ROUTINE IS USED TO ACCEPT INPUT PARAMETERS FROM THE CONSOLE
 6627                                          ;TELETYPE
 6628
 6629  026046  104401                  INPARA: TYPE
 6630  026050  035354                          VCWC                    ;"ASK FOR NO. ADDRESSES BETWEEN VECTORS"
 6631  026052  104412                          RDOCT                   ;READ OCTAL NO. FM TTY
 6632  026054  012600                          MOV     (SP)+,R0        ;GET THE NO. HE TYPED
 6633  026056  001407                          BEQ     3$              ;BR IF HE TYPED <CR>
 6634  026060  022700  000010                  CMP     #10,R0          ;10(8) ADDRESSES BETWEEN VECTORS ?
 6635  026064  001406                          BEQ     4$              ;BR IF YES
 6636  026066  022700  000020                  CMP     #20,R0          ;20(8) ADDRESSES BETWEEN VECTORS ??
 6637  026072  001403                          BEQ     4$              ;BR IF YES
 6638  026074  000764                          BR      INPARA          ;ASK ALL OVER AGAIN
 6639  026076  012700  000020          3$:     MOV     #20,R0          ;SET UP CONSTANT FOR 20(8) ADDRESSES
 6640  026102  000207                  4$:     RTS     PC              ;RETURN TO CALLER
 6641
 6642
 6643
 6644  026104  012700  177777          INPARC: MOV     #-1,R0          ;SET FLAG IN R0
 6645  026110  000167  154034                  JMP     BEGINA          ;GO ASK FOR SELECT PARAMETER
```

```
                        6646
026114  012767  177777  001546  INPARX: MOV     #-1,VCFLG       ;SET SETUP FLAG
026122  000167  154022                  JMP     BEGINA          ;GO START UP
                        6649
026126  013701  027732          INPAR:  MOV     @#DHADRS,R1     ;MOVE ADDRESS OF FIRST DH INTO R1.
026132  032777  000001  153000          BIT     #BIT0,@SWR      ;ARE PARAMETERS TO BE INPUT MANUALLY?
026140  001405                          BEQ     2$              ;BRANCH IF NOT.
026142  104401                  1$:     TYPE                    ;ASK FOR DEVICE ADDRESS
026144  034731                          INMSG1
026146  104412                          RDOCT                   ;READ IN WHAT IS TYPED
026150  012601                          MOV     (SP)+,R1        ;GET THE NO. HE TYPED
026152  001403                          BEQ     INPAR1          ;BR IF DEFAULT
026154  004767  000160          2$:     JSR     PC,CHKADR       ;GO CHECK VALIDITY OF THE ADDR
026160  000770                          BR      1$              ;ERROR BRANCH
                        6660
026162  013701  027774          INPAR1: MOV     @#DHVEC,R1      ;MOVE FIRST DH VECTOR INTO R1.
026166  032777  000001  152744          BIT     #BIT0,@SWR      ;ARE PARAMETERS TO BE INPUT MANUALLY?
026174  001405                          BEQ     2$              ;BRANCH IF NOT.
026176  104401                  1$:     TYPE                    ;ASK FOR VECTOR ADDRESS
026200  034775                          INMSG2
026202  104412                          RDOCT                   ;READ IN WHAT HE TYPES
026204  012601                          MOV     (SP)+,R1        ;GET THE ADDRESS
026206  001403                          BEQ     INPAR3          ;BR IF DEFAULT
026210  004767  000240          2$:     JSR     PC,CHKVCT       ;GO CHECK VALIDITY OF VECTOR
026214  000770                          BR      1$              ;ERROR BRANCH
                        6671
026216  013701  030204          INPAR3: MOV     @#$DHSEL,R1     ;MOVE DEVICE SELECTION PARAMETER INTO R1.
026222  005700                          TST     R0              ;DID WE START AT 210?
026224  100404                          BMI     2$              ;BRANCH IF YES.
026226  032777  000001  152704          BIT     #BIT0,@SWR      ;IS PARAMETER TO BE INPUT MANUALLY?
026234  001405                          BEQ     1$              ;BRANCH IF NOT.
026236  104401                  2$:     TYPE                    ;ASK FOR DEVICE SELECTION PARAMETER
026240  035044                          INMSG3
026242  104412                          RDOCT                   ;READ IN WHAT HE TYPES
026244  012601                          MOV     (SP)+,R1        ;GET THE SELECT PARAMETER
026246  001402                          BEQ     INPAR4          ;BR IF  DEFAULT
026250  010167  000724          1$:     MOV     R1,DHSEL        ;SET UP DH11 SELECTION PARAMETER
                        6683
026254  005700                  INPAR4: TST     R0              ;DID WE START AT 210?
026256  100404                          BMI     3$              ;BRANCH IF YES.
026260  032777  000001  152652          BIT     #BIT0,@SWR      ;IS LINE SELECT PARAMETER TO BE INPUT MANUALLY?
026266  001410                          BEQ     1$              ;BRANCH IF NO.
026270  104401                  3$:     TYPE                    ;ASK FOR LINE SELECT PARAMETER
026272  035242                          INMSG6
026274  104412                          RDOCT                   ;GET WHAT HE TYPES
026276  012601                          MOV     (SP)+,R1        ;GET PARAMETER
026300  001403                          BEQ     1$              ;BR IF DEFAULT
026302  010167  000674                  MOV     R1,LINSEL       ;SET UP LINE SELECT PARAMETER
026306  000403                          BR      2$              ;CONTINUE
026310  012767  177777  000664  1$:     MOV     #-1,LINSEL      ;SET UP DEFAULT (ALL LINES)
026316  032777  000400  152614  2$:     BIT     #BIT8,@SWR      ;HALT AFTER SET UP ??
026324  001403                          BEQ     EXPAR           ;BR IF  NOT
026326  104401                          TYPE                    ;TYPE CONTINUE MESSAGE PRIOR TO HALTING
026330  035304                          INMSG7
026332  000000                          HALT                    ;DEPRESS CONTINUE TO RESUME TESTING
026334  000167  154264          EXPAR:  JMP     START2          ;GO START UP THE PROGRAM
```

```
                                                
                                                
026340  020127  160020          CHKADR: CMP     R1,#160020      ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
026344  002001                          BGE     1$              ;BR IF YES
026346  000437                          BR      4$              ;BR IF NOT
026350  020127  160420          1$:     CMP     R1,#160420      ;IS IT BELOW THE HIGH LIMIT?
026354  002401                          BLT     2$              ;BR IF YES
026356  000433                          BR      4$              ;BR IF NOT
026360  032701  000017          2$:     BIT     #17,R1          ;CORRECT BOUNDARY ?
026364  001030                          BNE     4$              ;BR IF NOT
026366  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC AROUND ERROR BRANCH
026372  012702  027570                  MOV     #DHADTB,R2      ;POINT TO BEGIN OF ADDR TABLE
026376  032777  000001  152534          BIT     #BIT0,@$SWR     ;ARE WE AUTOSIZING?
026404  001011                          BNE     3$              ;BRANCH IF NOT.
026406  012703  027732                  MOV     #DHADRS,R3      ;POINT TO BEGINNING OF AUTOSIZER
                                                                ;DH ADDRESS TABLE.
026412  016704  001566                  MOV     $DHSEL,R4
026416  012322                  6$:     MOV     (R3)+,(R2)+     ;MOVE CONTENTS OF AUTOSIZER DH TABLE
                                                                ;TO THE TABLE USED BY PROGRAM.
026420  006204                          ASR     R4
026422  005704                          TST     R4              ;HAVE WE MOVED ALL TABLE ENTRIES?
026424  001374                          BNE     6$              ;BRANCH IF NOT--ONE MORE TIME.
026426  000411                          BR      5$              ;RETURN TO INPUT ROUTINES.
026430  010122                  3$:     MOV     R1,(R2)+        ;SETR UP A TABLE ENTRY
026432  062701  000020                  ADD     #20,R1          ;GENERATE NEXT DH11 ADDR
026436  022702  027630                  CMP     #DHADTB+40,R2   ;END OF TABLE ?
026442  001372                          BNE     3$              ;BR IF NOT
026444  000402                          BR      5$              ;RETURN TO INPUT ROUTINES
026446  104401                  4$:     TYPE                    ;TELL HIM HE GOOFED
026450  035215                          INMSG4
026452  000207                  5$:     RTS     PC              ;RETURN TO INPUT ROUTINES
                                                
026454  020127  000300          CHKVCT: CMP     R1,#300         ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
026460  002001                          BGE     1$              ;BR IF YES
026462  000436                          BR      4$              ;BR IF NOT
026464  020127  001000          1$:     CMP     R1,#1000        ;IS IT BELOW THE HIGH LIMIT?
026470  002401                          BLT     2$              ;BR IF YES
026472  000432                          BR      4$              ;BR IF NOT
026474  032701  000007          2$:     BIT     #7,R1           ;CORRECT BOUNDARY ?
026500  001027                          BNE     4$              ;BR IF NOT
026502  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC AROUND ERROR BRANCH
026506  012702  027630                  MOV     #DHVCTB,R2      ;POINT TO BEGIN OF VECTOR TABLE
026512  032777  000001  152420          BIT     #BIT0,@$SWR     ;ARE WE AUTOSIZING?
026520  001011                          BNE     3$              ;BRANCH IF NOT.
026522  012703  027774                  MOV     #DHVEC,R3       ;POINT TO BEGINING OF AUTOSIZER
                                                                ;DH VECTOR TABLE.
026526  016704  001452                  MOV     $DHSEL,R4
026532  012322                  6$:     MOV     (R3)+,(R2)+     ;MOVE CONTENTS OF AUTOSIZER VECTOR
                                                                ;TABLE TO TABLE USED BY PROGRAM.
026534  006204                          ASR     R4
026536  005704                          TST     R4              ;HAVE WE MOVED ALL TABLE ENTRIES?
026540  001374                          BNE     6$              ;BRANCH IF NOT--ONE MORE TIME.
026542  000410                          BR      5$              ;RETURN TO INPUT ROUTINES.
026544  010122                  3$:     MOV     R1,(R2)+        ;SETR UP A TABLE ENTRY
026546  060001                          ADD     R0,R1           ;GENERATE NEXT DH11 ADDR
026550  022702  027670                  CMP     #DHVCTB+40,R2   ;END OF TABLE ?
```

```
 6758  026554  001373                          BNE     3$              ;BR IF NOT
 6759  026556  000402                          BR      5$              ;RETURN TO INPUT ROUTINES
 6760  026560  104401                  4$:     TYPE                    ;TELL HIM HE GOOFED
 6761  026562  035166                          INMSGS
 6762  026564  000207                  5$:     RTS     PC              ;RETURN TO INPUT ROUTINES
 6763
 6764                                  ;THESE TWO ROUTINES SERVICE UNEXPECTED BUS ERROR AND RSVD INSTR TRAPS
 6765
 6766  026566  012767  000340  152406  BUSER:  MOV     #340,$TMP0      ;SAVE THE PSW
 6767  026574  010667  152376          BUSER:  MOV     SP,$REG6        ;SAVE THE SP
 6768  026600  012601                          MOV     (SP)+,R1        ;GET THE TRAP PC
 6769  026602  012602                          MOV     (SP)+,R2        ;GET THE TRAP PSW
 6770  026604  116700  152272                  MOVB    $TSTNM,R0       ;GET TEST NO.
 6771  026610  012706  001100                  MOV     #STACK,SP       ;RESET THE STACK POINTER
 6772  026614  004767  175600                  JSR     PC,SUER3        ;GO SET UP ERROR INFO
 6773  026620  012767  026630  152262          MOV     #1$,$LPERR      ;ALWAYS COME BACK TO 1$
 6774  026626  104027                          ERROR   27              ;UNEXPECTED BUS ERROR TRAP
 6775  026630  000005                  1$:     RESET                   ;PREPARE TO RESTART.
 6776  026632  004767  000240                  JSR     PC,CHPS1        ;GO CLEAR PSW
 6777  026636  000167  154056                  JMP     REST1           ;GO RESTART THE PROGRAM
 6778
 6779  026642  012767  000340  152332  RESERR: MOV     #340,$TMP0      ;SAVE THE PSW
 6780  026650  010667  152322          RESERR: MOV     SP,$REG6        ;SAVE THE SP
 6781  026654  012601                          MOV     (SP)+,R1        ;GET THE TRAP PC
 6782  026656  012602                          MOV     (SP)+,R2        ;GET THE TRAP PSW
 6783  026660  116700  152216                  MOVB    $TSTNM,R0       ;GET TEST NO.
 6784  026664  012706  001100                  MOV     #STACK,SP       ;RESET THE STACK POINTER
 6785  026670  004767  175524                  JSR     PC,SUER3        ;GO SET UP ERROR INFO
 6786  026674  012767  026704  152206          MOV     #1$,$LPERR      ;ALWAYS COME BACK TO 1$
 6787  026702  104030                          ERROR   30              ;UNEXPECTED RSVD INSTR ERROR TRAP
 6788  026704  000005                  1$:     RESET                   ;PREPARE TO RESTART
 6789  026706  004767  000164                  JSR     PC,CHPS1        ;GO CLEAR PSW
 6790  026712  000167  154002                  JMP     REST1           ;GO RESTART THE PROGRAM
 6791
 6792                                  ;THIS ROUTINE IS CALLED WHEN A TEST NEEDS TO RESTORE THE TRAP
 6793                                  ;CATCHER IN THE DH11 VECTOR
 6794
 6795  026716  016703  000252          RESTRP: MOV     DHVCT,R3        ;GET VECTOR ADDRESS
 6796  026722  010313                          MOV     R3,(R3)         ;RESTORE THE TRAP CATCHER
 6797  026724  062723  000002                  ADD     #2,(R3)+
 6798  026730  005023                          CLR     (R3)+
 6799  026732  010313                          MOV     R3,(R3)
 6800  026734  062723  000002                  ADD     #2,(R3)+
 6801  026740  005023                          CLR     (R3)+
 6802  026742  000207                          RTS     PC              ;RETURN TO CALLING TEST
 6803
 6804                                  ;THIS ROUTINE CALLED BY ANY TEST THAT NEEDS A TIMING WAIT LOOP
 6805                                  ;"TIMEA" IS INITIALIZED BY THE CALLING ROUTINE TO THE MINIMUM REQUIRED
 6806                                  ;VALUE AND "TIMEB" IS CLEARED TO 000000. IF A TIME OUT OCCURS THIS
 6807                                  ;ROUTINE WILL MOVE THE RETURN PC AROUND THE "LOOP" BRANCH BACK IN
 6808                                  ;THE ROUTINE THAT CALLED IT TO ALLOW REPORTING AN ERROR MESSAGE
 6809
 6810  026744  005267  001272          TIMEIT: INC     TIMEB           ;COUNT B
 6811  026750  001005                          BNE     1$              ;BR IF NOT ZERO
 6812  026752  005367  001262                  DEC     TIMEA           ;COUNT TIME A
 6813  026756  001002                          BNE     1$              ;BR IF NO TIMEOUT
```

```
 6814  026760  062716  000002                    ADD     #2,(SP)         ;MOVE RETURN PC TO ALLOW ERROR REPORT
 6815  026764  000207                    1$:     RTS     PC              ;RETURN TO THE CALLING TEST
 6816
 6817                                    ;THIS ROUTINE CALLED BY THE AUTO ECHO TEST TO SET UP FOR TRANSFERRING
 6818                                    ;A BINARY COUNT TEST PATTERN ON ALL LINES
 6819
 6820  026766  012767  000020  152222    SETALL: MOV     #20,$TMP6       ;SET UP SIXTEEN LINES
 6821  026774  005002                            CLR     R2              ;INIT A TABLE INDEX REG
 6822  026776  012703  000200                    MOV     #200,R3         ;SET UP TO GENERATE HI BYTE OF EXPECTED DATA
 6823  027002  012704  000001                    MOV     #1,R4           ;SET UP INDEX REG TO ODD BYTES
 6824  027006  005011                            CLR     (R1)            ;START WITH LINE 00
 6825  027010  012761  037112  000006    1$:     MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDR REG
 6826  027016  012761  177400  000010            MOV     #-400,BCR(R1)   ;SET UP BYTE COUNT REG
 6827  027024  012761  031403  000004            MOV     #31403,LPR(R1)  ;SET UP FOR 4800 BAUD/8 BIT CHARS
 6828  027032  105062  036112                    CLRB    RBUF(R2)        ;START WITH DATA CHAR OF 000
 6829  027036  110364  036112                    MOVB    R3,RBUF(R4)     ;SET UP HIGH BYTE OF EXPECTED DATA
 6830  027042  005211                            INC     (R1)            ;GEN NEW LINE NO. IN SCR
 6831  027044  005203                            INC     R3              ;UPDATE THE POINTERS AND DATA
 6832  027046  062702  000002                    ADD     #2,R2
 6833  027052  062704  000002                    ADD     #2,R4
 6834  027056  005367  152134                    DEC     $TMP6           ;COUNT ONE LINE DONE
 6835  027062  001352                            BNE     1$              ;BR TIL ALL 16 SET UP
 6836  027064  016767  000112  000356            MOV     LINSEL,LINACT   ;SET SOFTWARE FLAG FOR ALL LINES ACTIVE
 6837  027072  005011                            CLR     (R1)            ;PUT SCR REG BACK TO LINE 00
 6838  027074  000207                            RTS     PC              ;RETURN TO AUTO ECHO TEST
 6839
 6840
 6841                                    ;THIS ROUTINE IS CALLED TO SET PSW PRIORITY TO 000 IN ORDER
 6842                                    ;TO BE LSI11 COMPATIBLE
 6843
 6844  027076  012746  000000            CHPS1:  MOV     #0,-(SP)        ;NEW PSW
 6845  027102  012746  027110                    MOV     #1$,-(SP)       ;NEW PC
 6846  027106  000002                            RTI                     ;CHANGE PSW
 6847  027110  000207                    1$:     RTS     PC              ;RETURN TO CALLING TEST
 6848
 6849                                    ;THIS ROUTINE DOES THE SAME THING EXCEPT IT SET THE PSW
 6850                                    ;PRIORITY TO 340 (LEVEL 7 ) TO LOCK OUT INTRS
 6851
 6852  027112  012746  000340            CHPS2:  MOV     #340,-(SP)      ;NEW PSW
 6853  027116  012746  027124                    MOV     #1$,-(SP)       ;NEW PC
 6854  027122  000002                            RTI                     ;CHANGE THE PSW
 6855  027124  000207                    1$:     RTS     PC              ;RETURN TO CALLING TEST
 6856
 6857                                    ;THIS ROUTINE IS ALSO FOR LSI11 COMPATIBILITY AND IT IS CALLED
 6858                                    ;TO SAVE THE PSW IN "$TMP0"
 6859
 6860  027126  005046                    SAPS:   CLR     -(SP)           ;TEMP STORAGE TO SAVE PSW
 6861  027130  016746  150700                    MOV     34,-(SP)        ;SAVE TRAP VECTOR POINTER
 6862  027134  012767  027144  150672            MOV     #1$,34          ;GO TO 1$ ON TRAP
 6863  027142  104400                            TRAP                    ;GO TO IT
 6864  027144  016666  000002  000006    1$:     MOV     2(SP),6(SP)     ;GET PSW SAVED
 6865  027152  012716  027160                    MOV     #2$,(SP)        ;GO TO 2$ ON RTI
 6866  027156  000002                            RTI
 6867  027160  012667  150650            2$:     MOV     (SP)+,34        ;RESTORE VECTOR
 6868  027164  012667  152012                    MOV     (SP)+,$TMP0     ;FINALLY SAVE PSW IN $TMP0
 6869  027170  000207                            RTS     PC
```

```
;
```

```
                        ;;*****************************************************************
                        ;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 ADDRESS REG.
                 000014         BKR=14          ;INDEX CONST. TO ACCESS BREAK CONTROL REG.
                 000016         SSR=16          ;INDEX CONST. TO ACCESS SILO STATUS REG.

027172  000000  DHADR:  0               ;HOLDS THE "SCR" ADDRESS OF THE DH11 UNDER TEST
027174  000000  DHVCT:  0               ;HOLDS THE 1ST VECTOR ADDRESS OF THE DH11 UNDER TEST
027176  000000  SEL_MSK: 0              ;BIT TST MARKER FOR SELECTING DH11'S
027200  000003  DHSEL:  3               ;SPECIFIES DH11'S SELECTED FOR TEST
027202  177777  LINSEL: 177777          ;SPECIFIES LINES TO TEST
027204  000000  LINMSK: 0               ;MARKER USED TO TEST FOR LINES TO TEST
027206  000000  LMSK1:  0               ;ALTERNATE MARKER TO SUPPORT THE
                                        ;SELECT LINES FEATURE
027210  000004  MSTCLR: .BLKW   4       ;FOUR WORD ADDRESS TABLE USED BY THE TEST THAT
                                        ;CHECKS OPERATION OF "MASTER CLR"

027220  177777  PATRNA: 177777          ;BIT PATTERNS USED WITH "CAR" AND "BCR" TESTS
027222  125252          125252
027224  052525          052525
027226  000000          000000          ;TABLE TERMINATOR

027230  000060  PATRNB: 60              ;BIT PATTERNS USED IN "CAR" MEM EXT BIT TEST
027232  000300          300
027234  000020          20
027236  000100          100
027240  000040          40
027242  000200          200
027244  000000          0               ;TABLE TERMINATOR
027246  000000          0               ;TABLE TERMINATOR

                        ;THIS TABLE STORES THE BYTE COUNT AND LINE PARAMETERS FOR THE
                        ;8 SUBTESTS IN THE MULTILINE PARITY/DATA TEST

027250  177400  PRTYTB: -400                    ;256 CHARS
027252  027363          27363                 ;2400 BAUD - ODD PARITY - 8 BITS
027254  177400          -400                  ;256 CHARS
027256  027323          27323                 ;2400 BAUD - EVEN PARITY - 8 BITS
027260  177600          -200                  ;128 CHARS
027262  027362          27362                 ;2400 BAUD - ODD PARITY - 7 BITS
027264  177600          -200                  ;128 CHARS
027266  027322          27322                 ;2400 BAUD - EVEN PARITY - 7 BITS
027270  177700          -100                  ;64 CHARS
027272  027361          27361                 ;2400 BAUD - ODD PARITY - 6 BITS
027274  177700          -100                  ;64 CHARS
027276  027321          27321                 ;2400 BAUD - EVEN PARITY - 6 BITS
027300  177740          -40                   ;32 CHARS
027302  027360          27360                 ;2400 BAUD - ODD PARITY - 5 BITS
027304  177740          -40                   ;32 CHARS
027306  027320          27320                 ;2400 BAUD - EVEN PARITY - 5 BITS
```

```
                                        ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
                                        ;TEST (ALL 1'S DATA TABLE)

027310  100377          AETAB:  100377          ;TEST DATA FOR LINE 00
027312  100777                  100777          ;TEST DATA FOR LINE 01
027314  101377                  101377
027316  101777                  101777
027320  102377                  102377
027322  102777                  102777
027324  103377                  103377
027326  103777                  103777
027330  104377                  104377
027332  104777                  104777
027334  105377                  105377
027336  105777                  105777
027340  106377                  106377
027342  106777                  106777
027344  107377                  107377
027346  107777                  107777          ;TEST DATA FOR LINE 17

                                        ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
                                        ;TEST (ALL 0'S DATA TABLE)

027350  100000          AETAB0: 100000          ;TEST DATA FOR LINE 00
027352  100400                  100400          ;TEST DATA FOR LINE 01
027354  101000                  101000
027356  101400                  101400
027360  102000                  102000
027362  102400                  102400
027364  103000                  103000
027366  103400                  103400
027370  104000                  104000
027372  104400                  104400
027374  105000                  105000
027376  105400                  105400
027400  106000                  106000
027402  106400                  106400
027404  107000                  107000
027406  107400                  107400          ;TEST DATA FOR LINE 17

                                ;THIS TABLE USED BY THE AUTO ECHO TEST 2 TO RESET ACTIVE BIT WHEN A
                                ;LINE IS DONE

027410  000001          LINBIT: BIT00                   ;DEACTIVATE LINE 00
027412  000002                  BIT01                   ;DEACTIVATE LINE 01
027414  000004                  BIT02
027416  000010                  BIT03
027420  000020                  BIT04
027422  000040                  BIT05
027424  000100                  BIT06
027426  000200                  BIT07
027430  000400                  BIT08
027432  001000                  BIT09
027434  002000                  BIT10
027436  004000                  BIT11
```

```
                                                BIT12
                                                BIT13
                                                BIT14
                                                BIT15                   ;DEACTIVATE LINE 17

027450  000000                          LINACT: 0                       ;MAINTAINS STATUS OF ACTIVE LINES
                                                                        ;DURING AUTO ECHO TEST 2


                                        ;THIS TABLE CONTAINS 16. COUNTERS USED B N THE MULTI-LINE
                                        ;PARITY TEST TO KEEP TRACK OF TOTAL CHARS RECEIVED

027452  000020                          MULPTB: .BLKW   16.             ;SIXTEEN WORD COUNTERS TABLE

                                        ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE BREAK BIT
                                        ;TEST

027512  120000                          BRKTAB: 120000          ;TEST DATA FOR LINE 00
027514  120400                                  120400          ;TEST DATA FOR LINE 01
027516  121000                                  121000
027520  121400                                  121400
027522  122000                                  122000
027524  122400                                  122400
027526  123000                                  123000
027530  123400                                  123400
027532  124000                                  124000
027534  124400                                  124400
027536  125000                                  125000
027540  125400                                  125400
027542  126000                                  126000
027544  126400                                  126400
027546  127000                                  127000
027550  127400                                  127400          ;TEST DATA FOR LINE 17

027552  131177                          RGMSK1: 131177          ;MASK TO SPECIFY R/W BITS FOR NORMAL "SCR" REG TEST
027554  046600                          RGMSK2: 46600           ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR NORMAL MODE TEST
027556  177767                          RGMSK3: 177767          ;MASK TO SPECIFY R/W BITS IN "LPR"
027560  177777                          RGMSK4: 177777          ;MASK TO SPECIFY R/W BITS IN "BKR"
027562  100077                          RGMSK5: 100077          ;MASK TO SPECIFY R/W BITS IN "SSR"
027564  042200                          RGMSK6: 42200           ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR MAINT. MODE TEST
027566  030100                          INTMSK: 30100           ;MASK USED TO SELECT INTR BITS TO TEST


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

027570  160020                          DHADTB: 160020          ;ADDRESS OF FIRST DH11
027572  160040                                  160040          ;ADDRESS OF SECOND DH11
027574  160060                                  160060
027576  160100                                  160100
027600  160120                                  160120
027602  160140                                  160140
027604  160160                                  160160
027606  160200                                  160200
027610  160220                                  160220
027612  160240                                  160240
027614  160260                                  160260
```

```
                                                160300
                                                160320
                                                160340
                                                160360
027626  160400                                  160400          ;ADDRESS OF THE LAST DH11

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

027630  000330                          DHVCTB: 330             ;ADDRESS OF VECTOR FOR FIRST DH11
027632  000350                                  350             ;ADDRESS OF VECTOR FOR SECOND DH11
027634  000370                                  370
027636  000410                                  410
027640  000430                                  430
027642  000450                                  450
027644  000470                                  470
027646  000510                                  510
027650  000530                                  530
027652  000550                                  550
027654  000570                                  570
027656  000610                                  610
027660  000630                                  630
027662  000650                                  650
027664  000670                                  670
027666  000710                                  710             ;ADDRESS OF VECTOR FOR LAST DH11

027670  000000                          VCFLG:  0               ;VECTOR SET UP FLAGG


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

027672  120240                          BRLVL:  120240          ;BRLEVELS FOR FIRST DH11
027674  120240                                  120240          ;BR LEVELS FOR SECOND DH11
027676  120240                                  120240
027700  120240                                  120240
027702  120240                                  120240
027704  120240                                  120240
027706  120240                                  120240
027710  120240                                  120240
027712  120240                                  120240
027714  120240                                  120240
027716  120240                                  120240
027720  120240                                  120240
027722  120240                                  120240
027724  120240                                  120240
027726  120240                                  120240
027730  120240                                  120240          ;BR LEVELS FOR LAST DH11

                                        ;THIS DH ADDRESS TABLE IS FILLED BY THE AUTOSIZER.

027732                                  DHADRS:
027732  000000                          .WORD   0
027734  000000                          .WORD   0
027736  000000                          .WORD   0
```

```
7095  027740  000000                  .WORD   0
7096  027742  000000                  .WORD   0
7097  027744  000000                  .WORD   0
7098  027746  000000                  .WORD   0
7099  027750  000000                  .WORD   0
7100  027752  000000                  .WORD   0
7101  027754  000000                  .WORD   0
7102  027756  000000                  .WORD   0
7103  027760  000000                  .WORD   0
7104  027762  000000                  .WORD   0
7105  027764  000000                  .WORD   0
7106  027766  000000                  .WORD   0
7107  027770  000000                  .WORD   0
7108  027772  000000                  .WORD   0
7109
7110                                  ;THIS DH VECTOR TABLE IS FILLED BY THE AUTOSIZER.
7111
7112  027774                          DHVEC:
7113  027774  000000                  .WORD   0
7114  027776  000000                  .WORD   0
7115  030000  000000                  .WORD   0
7116  030002  000000                  .WORD   0
7117  030004  000000                  .WORD   0
7118  030006  000000                  .WORD   0
7119  030010  000000                  .WORD   0
7120  030012  000000                  .WORD   0
7121  030014  000000                  .WORD   0
7122  030016  000000                  .WORD   0
7123  030020  000000                  .WORD   0
7124  030022  000000                  .WORD   0
7125  030024  000000                  .WORD   0
7126  030026  000000                  .WORD   0
7127  030030  000000                  .WORD   0
7128  030032  000000                  .WORD   0
7129
7130                                  ;THIS DM ADDRESS TABLE IS FILLED BY THE AUTOSIZER.
7131
7132  030034                          DMADRS:
7133  030034  000000                  .WORD   0
7134  030036  000000                  .WORD   0
7135  030040  000000                  .WORD   0
7136  030042  000000                  .WORD   0
7137  030044  000000                  .WORD   0
7138  030046  000000                  .WORD   0
7139  030050  000000                  .WORD   0
7140  030052  000000                  .WORD   0
7141  030054  000000                  .WORD   0
7142  030056  000000                  .WORD   0
7143  030060  000000                  .WORD   0
7144  030062  000000                  .WORD   0
7145  030064  000000                  .WORD   0
7146  030066  000000                  .WORD   0
7147  030070  000000                  .WORD   0
7148  030072  000000                  .WORD   0
7149  030074  000000                  .WORD   0
7150
```

```
                                        ;THIS DM VECTOR TABLE IS FILLED BY THE AUTOSIZER.

030076                                  DMVEC:
030076  000000                          .WORD   0
030100  000000                          .WORD   0
030102  000000                          .WORD   0
030104  000000                          .WORD   0
030106  000000                          .WORD   0
030110  000000                          .WORD   0
030112  000000                          .WORD   0
030114  000000                          .WORD   0
030116  000000                          .WORD   0
030120  000000                          .WORD   0
030122  000000                          .WORD   0
030124  000000                          .WORD   0
030126  000000                          .WORD   0
030130  000000                          .WORD   0
030132  000000                          .WORD   0
030134  000000                          .WORD   0

030136                                  $LNSEL:
030136  000001                          .WORD   1
030140  000002                          .WORD   2
030142  000004                          .WORD   4
030144  000010                          .WORD   10
030146  000020                          .WORD   20
030150  000040                          .WORD   40
030152  000100                          .WORD   100
030154  000200                          .WORD   200
030156  000400                          .WORD   400
030160  001000                          .WORD   1000
030162  002000                          .WORD   2000
030164  004000                          .WORD   4000
030166  010000                          .WORD   10000
030170  020000                          .WORD   20000
030172  040000                          .WORD   40000
030174  100000                          .WORD   100000

030176  000000                  ADRVEC: 0               ;ADDRESSES BETWEEN VECTORS - FILLED BY THE AUTOSIZER
030200  000000                  DHMCSR: 0               ;DM11-BB CONTROL AND STATUS REGISTER.
030202  000000                  DHMLSR: 0               ;DM11-BB LINE STATUS REGISTER.
030204  000000                  $DHSEL: 0               ;DEVICE SELECT PARAMETER - FILLLED BY THE AUTOSIZER.
030206     000                  DHRLVL: .BYTE   0               ;BR LEVEL FOR RCVR
030207     000                  DHTLVL: .BYTE   0               ;BR LEVEL FOR XMITTER
030210  000000                  DHNUM:  0               ;CONTAINS NUMBER OF THE DM11 UNDER TEST
030212  000000                  LINE:   0               ;CONTINES NUMBER OF THE LINE UNDER TEST
030214  000000                  LINEA:  0               ;LOCATION TO SAVE LINE NUMBER
                                ;ADDRESS POINTERS TO SET UP TABLES WHEN INPUTTING PARAMETERS

030216  000000                  ADPTR:  0                       ;POINTS TO ADDRESS TABLE
030220  000000                  VCPTR:  0                       ;POINTS TO VECTOR TABLE
030222  000000                  BRPTR:  0                       ;POINTS TO BR LEVEL TABLE
                                ;THE FOLLOWING TEN CONSTANTS ARE USED BY THE DM11-BB PORTION OF THIS PROGRAM.
```

```
                        000400          STEP=400
                        000001          LINENA=1
                        000002          TRMRDY=2
                        000004          RS=4
                        000010          SECTX=10
                        000020          SECRX=20
                        000040          CS=40
                        000100          CO=100
                        000200          RING=200
                        002000          CLRMUX=2000

030224  000000          TDATA1: 0               ;DATA BUFFER FOR BASIC DATA TEST
030226  000037          TDATA2: 37              ;TEST DATA FOR FIVE BIT CHAR
030230  000077                  77              ;TEST DATA FOR SIX BIT CHAR
030232  000177                  177             ;TEST DATA FOR SEVEN BIT CHAR
030234  000377                  377             ;TEST DATA FOR EIGHT BIT CHAR
030236  000000          TITFLG: 0               ;FLAG TO ALLOW PRINTING TITLE ONLY ONCE
030240  000000          TIMEA:  0               ;GENERAL PURPOSE TIMERS
030242  000000          TIMEB:  0
030244  000000          TIMEC:  0               ;TIMER FOR TIMING TESTS
030246  000000          TNULL:  0               ;CONTAINS TWO NULL CHARS USED BY BREAK TEST
```

```
                                        ;;*****************************************************************
                                        ;ERROR MESSAGE INFORMATION - MESSAGE BUFFERS AND POINTERS
                                        ;;*****************************************************************

                                        ;INFORMATION FOR MESSAGE 1

030250  044104  030461  051040  EM1:    .ASCIZ  'DH11 REGISTER REFERENCE CAUSED TIMECUT'
030256  043505  051511  042524
030264  020122  042522  042506
030272  042522  041516  020105
030300  040503  051525  042105
030306  052040  046511  047505
030314  052125     000
030317     040  050050  024503  DH1:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  REGADR'
030324  020040  020040  050050
030332  024523  020040  020040
030340  051450  024520  020040
030346  020040  042524  052123
030354  020040  042040  053105
030362  042101  020122  051040
030370  043505  042101  000122
                        .EVEN
030376  001116  001202  001176  DT1:    .WORD   $ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,0
030404  001162  001164  001166
030412  000000

                                        ;INFORMATION FOR MESSAGE 2

030414  054523  052123  046505  EM2:    .ASCIZ  'SYSTEM CONTROL REGISTER ERROR'
030422  041440  047117  051124
030430  046117  051040  043505
030436  051511  042524  020122
030444  051105  047522  000122
030452  024040  041520  020051  DH2:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S B'
030460  020040  024040  051520
030466  020051  020040  024040
030474  050123  020051  020040
030502  052040  051505  020124
030510  020040  042504  040526
030516  051104  020040  042522
030524  040507  051104  020040
030532  053440  051501  020040
030540  020040  051440  041057
030546     000
        030550                  .EVEN
030550  001116  001202  001176  DT2:    .WORD   $ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4,0
030556  001162  001164  001166
030564  001170  001172  000000

                                        ;INFORMATION FOR MESSAGE 3

030572  044104  030461  046440  EM3:    .ASCIZ  'DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG'
030600  051501  042524  020122
030606  046103  040505  020122
030614  040506  046111  042105
030622  052040  020117  046103
```

```
 7287  030630  020122  050123  041505
 7288  030636  043111  042511  020104
 7289  030644  042522  000107
 7290
 7291                                  ;INFORMATION FOR MESSAGE 4
 7292
 7293  030650  044514  042516  050040  EM4:    .ASCIZ  'LINE PARAMETER REGISTER ERROR'
 7294  030656  051101  046501  052105
 7295  030664  051105  051040  043505
 7296  030672  051511  042524  020122
 7297  030700  051105  047522  000122
 7298
 7299                                  ;INFORMATION FOR MESSAGE 5
 7300
 7301  030706  051102  040505  020113  EM5:    .ASCIZ  'BREAK CONTROL REGISTER ERROR'
 7302  030714  047503  052116  047522
 7303  030722  020114  042522  044507
 7304  030730  052123  051105  042440
 7305  030736  051122  051117     000
 7306
 7307                                  ;INFORMATION FOR MESSAGE 6
 7308
 7309  030743     123  046111  020117  EM6:    .ASCIZ  'SILO STATUS REGISTER ERROR'
 7310  030750  052123  052101  051525
 7311  030756  051040  043505  051511
 7312  030764  042524  020122  051105
 7313  030772  047522  000122
 7314
 7315                                  ;INFORMATION FOR MESSAGE 7
 7316
 7317  030776  052503  051122  047105  EM7:    .ASCIZ  'CURRENT ADDRESS REGISTER ERROR - LINE #XX'
 7318  031004  020124  042101  051104
 7319  031012  051505  020123  042522
 7320  031020  044507  052123  051105
 7321  031026  042440  051122  051117
 7322  031034  026440  046040  047111
 7323  031042  020105  054043  000130
 7324
 7325                                  ;INFORMATION FOR MESSAGE 10
 7326
 7327  031050  054502  042524  041440  EM10:   .ASCIZ  'BYTE COUNTER REGISTER ERROR - LINE #XX'
 7328  031056  052517  052116  051105
 7329  031064  051040  043505  051511
 7330  031072  042524  020122  051105
 7331  031100  047522  020122  020055
 7332  031106  044514  042516  021440
 7333  031114  054130     000
 7334
 7335                                  ;INFORMATION FOR MESSAGE 11
 7336
 7337  031117     125  042516  050130  EM11:   .ASCIZ  'UNEXPECTED DH11 RCVR INTERRUPT'
 7338  031124  041505  042524  020104
 7339  031132  044104  030461  051040
 7340  031140  053103  020122  047111
 7341  031146  042524  051122  050125
 7342  031154  000124
```

```
                                        ;INFORMATION FOR MESSAGE 12

031156  047125  054105  042520  EM12:   .ASCIZ  'UNEXPECTED DH11 XMITTR INTERRUPT'
031164  052103  042105  042040
031172  030510  020061  046530
031200  052111  051124  044440
031206  052116  051105  052522
031214  052120     000

                                        ;INFORMATION FOR MESSAGE 13

031217     103  040510  020122  EM13:   .ASCIZ  'CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT'
031224  053101  044501  040514
031232  046102  020105  040506
031240  046111  042105  052040
031246  020117  042507  042516
031254  040522  042524  051040
031262  053103  020122  047111
031270  042524  051122  050125
031276  000124

                                        ;INFORMATION FOR MESSAGE 14

031300  051124  047101  046523  EM14:   .ASCIZ  'TRANSMITTER NPR LOGIC ERROR - LINE #  '
031306  052111  042524  020122
031314  050116  020122  047514
031322  044507  020103  051105
031330  047522  020122  020055
031336  044514  042516  021440
031344  020040     000

                                        ;INFORMATION FOR MESSAGE 15

031347     130  044515  052124  EM15:   .ASCIZ  'XMITTR FAILED TO INTERRUPT - LINE #  '
031354  020122  040506  046111
031362  042105  052040  020117
031370  047111  042524  051122
031376  050125  020124  020055
031404  044514  042516  021440
031412  020040     000

                                        ;INFORMATION FOR MESSAGE 16

031415     122  053103  020122  EM16:   .ASCIZ  'RCVR FAILED TO INTERRUPT'
031422  040506  046111  042105
031430  052040  020117  047111
031436  042524  051122  050125
031444  000124

                                        ;INFORMATION FOR MESSAGE 17

031446  051124  047101  046523  EM17:   .ASCIZ  'TRANSMITTER TIMING ERROR - LINE #  '
031454  052111  042524  020122
031462  044524  044515  043516
031470  042440  051122  051117
```

```
7399  031476  026440  046040  047111
7400  031504  020105  020043  000040
7401  031512  024040  041520  020051  DH6:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  SPEED   TIMEB   TIMEC'
7402  031520  020040  024040  051520
7403  031526  020051  020040  024040
7404  031534  050123  020051  020040
7405  031542  052040  051505  020124
7406  031550  020040  042504  040526
7407  031556  051104  020040  050123
7408  031564  042505  020104  020040
7409  031572  044524  042515  020102
7410  031600  020040  044524  042515
7411  031606  000103
7412
7413                                  ;INFORMATION FOR MESSAGE 20
7414
7415  031610  042522  042503  053111  EM20:   .ASCIZ  'RECEIVER TIMING ERROR - LINE #  '
7416  031616  051105  052040  046511
7417  031624  047111  020107  051105
7418  031632  047522  020122  020055
7419  031640  044514  042516  021440
7420  031646  020040     000
7421
7422                                  ;INFORMATION FOR MESSAGE 21
7423
7424  031651     122  053103  020122  EM21:   .ASCIZ  'RCVR FAILED TO INTERRUPT - LINE #  '
7425  031656  040506  046111  042105
7426  031664  052040  020117  047111
7427  031672  042524  051122  050125
7428  031700  020124  020055  044514
7429  031706  042516  021440  020040
7430  031714     000
7431
7432                                  ;INFORMATION FOR MESSAGE 22
7433
7434  031715     103  040510  020122  EM22:   .ASCIZ  'CHAR AVAIL FAILED TO SET ON TIME - LINE #  '
7435  031722  053101  044501  020114
7436  031730  040506  046111  042105
7437  031736  052040  020117  042523
7438  031744  020124  047117  052040
7439  031752  046511  020105  020055
7440  031760  044514  042516  021440
7441  031766  020040     000
7442
7443                                  ;INFORMATION FOR MESSAGE 23
7444
7445  031771     102  051501  041511  EM23:   .ASCIZ  'BASIC DATA TEST ERROR - LINE #  '
7446  031776  042040  052101  020101
7447  032004  042524  052123  042440
7448  032012  051122  051117  026440
7449  032020  046040  047111  020105
7450  032026  020043  000040
7451  032032  024040  041520  020051  DH7:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG   WAS     S B'
7452  032040  020040  024040  051520
7453  032046  020051  020040  024040
7454  032054  050123  020051  020040
```

```
7455  032062  052040  051505  020124
7456  032070  020040  042504  040526
7457  032076  051104  020040  044103
7458  032104  046122  043516  020040
7459  032112  053440  051501  020040
7460  032120  020040  051440  041057
7461  032126     000
7462
7463                                  ;INFORMATION FOR MESSAGE 24
7464
7465  032127     101  052125  020117  EM24:   .ASCIZ  'AUTO ECHO TEST ERROR - LINE #  '
7466  032134  041505  047510  052040
7467  032142  051505  020124  051105
7468  032150  047522  020122  020055
7469  032156  044514  042516  021440
7470  032164  020040     000
7471
7472                                  ;INFORMATION FOR MESSAGE 25
7473
7474  032167     102  042522  045501  EM25:   .ASCIZ  'BREAK BIT TEST ERROR - LINE #  '
7475  032174  041040  052111  052040
7476  032202  051505  020124  051105
7477  032210  047522  020122  020055
7478  032216  044514  042516  021440
7479  032224  020040     000
7480
7481                                  ;INFORMATION FOR MESSAGE 26
7482
7483  032227     110  046101  026506  EM26:   .ASCIZ  'HALF-DUPLEX TEST ERROR - LINE #  '
7484  032234  052504  046120  054105
7485  032242  052040  051505  020124
7486  032250  051105  047522  020122
7487  032256  020055  044514  042516
7488  032264  021440  020040     000
7489
7490                                  ;INFORMATION FOR MESSAGE 27
7491
7492  032271     125  042516  050130  EM27:   .ASCIZ  'UNEXPECTED BUS ERROR TRAP'
7493  032276  041505  042524  020104
7494  032304  052502  020123  051105
7495  032312  047522  020122  051124
7496  032320  050101     000
7497  032323     040  050050  024503  DH3:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   TRPPC   TRPPS '
7498  032330  020040  020040  050050
7499  032336  024523  020040  020040
7500  032344  051450  024520  020040
7501  032352  020040  042524  052123
7502  032360  020040  052040  050122
7503  032366  041520  020040  052040
7504  032374  050122  051520  000040
7505                                  .EVEN
7506  032402  001116  001202  001176  DT3:    .WORD   $ERRPC,$TMFC,$REG6,$REG0,$REG1,$REG2,0
7507  032410  001162  001164  001166
7508  032416  000000
7509
7510                                  ;INFORMATION    FOR MESSAGE 30
```

```
                                                    ;INFORMATION FOR MESSAGE 31
032453  000131  052125  020117  EM31:   .ASCIZ  'AUTO ECHO DATA COMPARE ERROR - LINE #  '
032460  041505  047510  040040
032466  052101  020101  047503
032474  050101  051101  020105
032502  051105  047522  020122
032510  020055  044514  042516
032516  021440  020040  000040
032523  000040  050050  024503  DH4:    .ASCIZ  ' (PC)    (PS)    SP)    TEST   WABADR  BBADR    WAB     B B'
032530  020040  020040  050050
032536  020440  020040  020040
032544  020440  020040  020040
032552  020040  042524  052123
032560  020040  053440  051501
032566  020440  020122  051440
032574  020040  051124  020040
032602  020040  020040  027523
032610  020040  020040  027523
032616  000102
                                                    ;INFORMATION FOR MESSAGE 32
032620  052501  047524  042440  EM32:   .ASCIZ  'AUTO ECHO TEST TIMEOUT - LINE #  '
032626  044103  020117  042524
032634  052123  052040  046511
032642  047505  052125  026440
032650  046040  047111  020105
032656  020043  000040
032662  024040  041520  020051  DH5:    .ASCIZ  ' (PC)   (PRG)   TEST'
032670  020040  046050  051120
032676  024507  020040  052040
032704  051505  000124
                                                    .EVEN
032710  001116  001202  001206  DT4:    .WORD   $ERRPC,$TMPC,$TMP2,C
032716  000000
                                                    ;INFORMATION FOR MESSAGE 33
032720  040520  044522  054524  EM33:   .ASCIZ  'PARITY LOGIC TEST ERROR - LINE #  '
032726  046040  043517  041511
032734  052040  051505  020124
032742  051105  047522  020122
032750  020055  044514  042516
032756  021440  020040  000040
                                                    ;INFORMATION FOR MESSAGE 34
032763  000115  046125  044524  EM34:   .ASCIZ  'MULTI-LINE PARITY DATA TEST ERROR - LINE #   - SUBTEST #  '
```

```
                                                ;INFORMATION FOR MESSAGE 35
033056  052515  052114  026511  EM35:   .ASCIZ  'MULTI-LINE PARITY DATA TEST TIMEOUT'
033064  044514  042516  050040
033072  051101  052111  020131
033100  040504  040524  052040
033106  051505  020124  044524
033114  042515  052517  000124
033122  024040  041520  020051  DH14:   .ASCIZ  ' (PC)   (LPRG)  ACTLIN'
033130  020040  046050  051120
033136  024507  020040  041501
033144  046124  047111     000
                033152          .EVEN
033152  001116  001202  001210  DT6:    .WORD   SERRPC,STMPC,STMP3,0
033160  000000
                                                ;INFORMATION FOR MESSAGE 36
033162  044103  051101  040440  EM36:   .ASCIZ  'CHAR AVAILABLE TIMEOUT'
033170  040526  046111  041101
033176  042514  052040  046511
033204  047505  052125     000
                                                ;INFORMATION FOR MESSAGE 37
033211     104  052101  020101  EM37:   .ASCIZ  'DATA COMPARE ERROR - LINE #  '
033216  047503  050115  051101
033224  020105  051105  047522
033232  020122  020055  044514
033240  042516  021440  020040
033246     000
                                                ;INFORMATION FOR MESSAGE 40
033247     102  043125  042506  EM40:   .ASCIZ  'BUFFER ACTIVE REG ERROR - LINE #  '
033254  020122  041501  044524
033262  042526  051040  043505
033270  042440  051122  051117
033276  026440  046040  047111
033304  020105  020043  000040
                                                ;INFORMATION FOR MESSAGE 41
033313  041522  051126  043040  EM41:   .ASCIZ  'RCVR FALSE INTERRUPT'
033321  046101  042523  044440
033326  052116  051105  052522
```

```
 7623  033334  052120     000
 7624
 7625                                ;INFORMATION FOR MESSAGE 42
 7626
 7627  033337     123  046111  020117  EM42:   .ASCIZ  'SILO OVERFLOW ERROR'
 7628  033344  053117  051105  046106
 7629  033352  053517  042440  051122
 7630  033360  051117     000
 7631
 7632                                ;INFORMATION FOR MESSAGE 43
 7633
 7634  033363     123  046111  020117  EM43:   .ASCIZ  'SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT'
 7635  033370  053117  051105  046106
 7636  033376  053517  043040  044501
 7637  033404  042514  020104  047524
 7638  033412  043440  047105  051105
 7639  033420  052101  020105  041522
 7640  033426  051126  044440  052116
 7641  033434  051105  052522  052120
 7642  033442     000
 7643
 7644                                ;INFORMATION FOR MESSAGE 44
 7645
 7646  033443     116  047117  042440  EM44:   .ASCIZ  'NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT'
 7647  033450  020130  042515  047515
 7648  033456  054522  043040  044501
 7649  033464  042514  020104  047524
 7650  033472  043440  047105  051105
 7651  033500  052101  020105  046530
 7652  033506  052111  051124  044440
 7653  033514  052116  051105  052522
 7654  033522  052120     000
 7655
 7656                                ;INFORMATION FOR MESSAGE 45
 7657
 7658  033525     130  044515  020124  EM45:   .ASCIZ  'XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT'
 7659  033532  047504  042516  043040
 7660  033540  044501  042514  020104
 7661  033546  047524  043440  047105
 7662  033554  051105  052101  020105
 7663  033562  046530  052111  051124
 7664  033570  044440  052116  051105
 7665  033576  052522  052120     000
 7666
 7667                                ;INFORMATION FOR MESSAGE 46
 7668
 7669  033603     103  051125  042522  EM46:   .ASCIZ  'CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE #  '
 7670  033610  052116  040440  042104
 7671  033616  042522  051523  046440
 7672  033624  046505  051117  020131
 7673  033632  043520  052124  051105
 7674  033640  051516  052040  051505
 7675  033646  020124  051105  047522
 7676  033654  020122  020055  044514
 7677  033662  042516  021440  020040
 7678  033670     000
```

E07


```
033671  040  050050  024503  CH10:   .ASCIZ  ' (PC)  LINEWR  PATTRN  TEST  DEVADR  REGADR  WAS    S/B'
033676  020040  046040  047111
033704  053505  020122  050040
033712  052101  051124  020116
033720  020040  042524  052123
033726  020040  042040  053105
033734  042101  020122  051040
033742  043505  042101  020122
033750  020040  040527  020123
033756  020040  020040  027523
033764  000102
                                .EVEN
033766  001116  001202  001204  DTS:    .WORD   $ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4,0
033774  001162  001164  001166
034002  001170  001172  000000

                                ;INFORMATION FOR MESSAGE 47

034010  054502  042524  041440  EM47:   .ASCIZ  'BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE #  '
034016  052517  052116  046440
034024  046505  051117  020131
034032  040520  052124  051105
034040  051516  052040  051505
034046  020124  051105  047522
034054  020122  020055  044514
034062  042516  021440  020040
034070      000

                                ;INFORMATION FOR MESSAGE 50

034071      124  051505  020124  EM50:   .ASCIZ  'TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  '
034076  044524  042515  052517
034104  020124  040527  052111
034112  047111  020107  047506
034120  020122  046530  052111
034126  042040  047117  020105
034134  020055  044514  042516
034142  021440  020040      000

                                ;INFORMATION FOR MESSAGE 51

034147      116  051120  046040  EM51:   .ASCIZ  'NPR LOGIC TEST 2 ERROR'
034154  043517  041511  052040
034162  051505  020124  020062
034170  051105  047522  000122
034176  024040  041520  020051  DH11:   .ASCIZ  ' (PC)   LINACT  LINCHK   TEST   DEVADR  REGADR   WAS    S B'
034204  020040  044514  040516
034212  052103  020040  044514
034220  041516  045510  020040
034226  052040  051505  020124
034234  020040  042504  040526
034242  051104  020040  042522
034250  040507  051104  020040
034256  053440  051501  020040
034264  020040  051440  041057
034272      000
```

```
                                        ;INFORMATION FOR MESSAGE 52
                                
034273    102   051501  041511  EM52:   .ASCIZ  'BASIC DATA COMPARE ERROR'
034303  042040  052101  020101
034306  047503  050115  051101
034314  020105  051105  047522
034322  000122

                                        ;INFORMATION FOR MESSAGE 53

034324  024040  041520  020051  DH12:   .ASCIZ  ' (PC)   SPEED    (SP)    TEST   DEVADR  REGADR  WAS     S/B'
034332  020040  050123  042505
034340  020104  020040  024040
034346  050123  020051  020040
034354  052040  051505  020124
034362  020040  042504  040526
034370  051104  020040  042522
034376  040507  051104  020040
034404  053440  051501  020040
034412  020040  051440  041057
034420     000

                                        ;INFORMATION FOR MESSAGE 55

034421     040  050050  024503  DH13:   .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG  SCRWAS  SCRS/B'
034426  020040  020040  050050
034434  024523  020040  020040
034442  051450  024520  020040
034450  020040  042524  052123
034456  020040  042040  053105
034464  042101  020122  041440
034472  051110  047114  020107
034500  051440  051103  040527
034506  020123  051440  051103
034514  027523  000102

                                        ;INFORMATIOPN FOR MESSAGE 56

034520  053117  051105  052522  EM56:   .ASCIZ  'OVERRUN BIT FAILED TO SET - LINE #  '
034526  020116  044502  020124
034534  040506  046111  042105
034542  052040  020117  042523
034550  020124  020055  044514
034556  042516  021440  020040
034564     000

                                        ;INFORMATION FOR MESSAGE 57

034565     123  047524  040522  EM57:   .ASCIZ  'STORAGE OVERFLOW BIT FAILED - LINE #  '
034572  042507  047440  042526
034600  043122  047514  020127
034606  044502  020124  040506
034614  046111  042105  026440
034622  046040  047111  020105
034630  020043  000040

                                .EVEN
```

```
                                        ;MISCELLANEOUS MESSAGES
                                        
034634  005015  040515  047111  TITLE:  .ASCIZ  <15><12>'MAINDEC-11-DZDHM-B  DH11 DIAGNOSTIC'<15><12>
034642  042504  026503  030461
034650  042055  042132  046510
034656  041055  020040  044104
034664  030461  042040  040511
034672  047107  051517  044524
034700  006503  000012
034704  005015  042524  052123  TITLE2: .ASCIZ  <15><12>'TESTING DH11 #  '<15><12>
034712  047111  020107  044104
034720  030461  021440  020040
034736  005015      000
034731      015  052012  050131  INMSG1: .ASCIZ  <15><12>'TYPE SCR ADDRESS FOR FIRST DH11'<15><12
034736  020105  041523  020122
034744  042101  051104  051505
034752  020123  047506  020122
034760  044506  051522  020124
034766  044104  030461  005015
034774      000
034775      015  052012  050131  INMSG2: .ASCIZ  <15><12>'TYPE VECTOR ADDRESS FOR FIRST DH11'<15><12>
035002  020105  042526  052103
035010  051117  040440  042104
035016  042522  051523  043040
035024  051117  043040  051111
035032  052123  042040  030510
035040  006461  000012
035044  005015  054524  042520  INMSG3: .ASCIZ  <15><12>'TYPE DH11 DEVICE SELECTION PARAMETER'<15><12>
035052  042040  030510  020061
035060  042504  044526  042503
035066  051440  046105  041505
035074  044524  047117  050040
035102  051101  046501  052105
035110  051105  005015      000
035115      015  044412  053116  INMSG4: .ASCIZ  <15><12>'INVALID DH11 SCR ADDRESS - TRY AGAIN'<15><12>
035122  046101  042111  042040
035130  030510  020061  041523
035136  020122  042101  051104
035144  051505  020123  020055
035152  051124  020131  043501
035160  044501  006516  000012
035166  005015  047111  040526  INMSG5: .ASCIZ  <15><12>'INVALID DH11 VECTOR ADDRESS - TRY AGAIN'<15><12>
035174  044514  020104  044104
035202  030461  053040  041505
035210  047524  020122  042101
035216  051104  051505  020123
035224  020055  051124  020131
035232  043501  044501  006516
035240  000012
035242  005015  054524  042520  INMSG6: .ASCIZ  <15><12>'TYPE LINE SELECTION PARAMETER'<15><12>
035250  046040  047111  020105
035256  042523  042514  052103
035264  047511  020116  040520
035272  040522  042515  042524
035300  006522  000012
035304  005015  042504  051120  INMSG7: .ASCIZ  <15><12>'DEPRESS "CONTINUE" TO START TESTING'<15><12>
```

```
 7847  035312  051505  020123  041442
 7848  035320  047117  044524  052516
 7849  035326  021105  052040  020117
 7850  035334  052123  051101  020124
 7851  035342  042524  052123  047111
 7852  035350  006507  000012
 7853
 7854  035354  005015  054524  042520  VCWC:   .ASCIZ  <15><12>'TYPE NO. OF ADDRESSES (OCTAL) BETWEEN VECTORS (10 OR 20)'<15><1
 7855  035362  047040  027117  047440
 7856  035370  020106  042101  051104
 7857  035376  051505  042523  020123
 7858  035404  047450  052103  046101
 7859  035412  020051  042502  053524
 7860  035420  042505  020116  042526
 7861  035426  052103  051117  020123
 7862  035434  030450  020060  051117
 7863  035442  031040  024460  005015
 7864  035450     000
 7865  035451     116  020117  044104  MSG1:   .ASCIZ  /NO DH11'S WERE FOUND./<15><12>
 7866  035456  030461  051447  053440
 7867  035464  051105  020105  047506
 7868  035472  047125  027104  005015
 7869  035500     000
 7870  035501     116  020117  044104  MSG2:   .ASCIZ  'NO DH RECEIVER INTERRUPT OCCURRED.'<15><12>
 7871  035506  051040  041505  044505
 7872  035514  042526  020122  047111
 7873  035522  042524  051122  050125
 7874  035530  020124  041517  052503
 7875  035536  051122  042105  006456
 7876  035544  000012
 7877  035546  047516  042040  030515  MSG3:   .ASCIZ  'NO DM11-BB INTERRUPT OCCURRED.'<15><12>
 7878  035554  026461  041102  044440
 7879  035562  052116  051105  052522
 7880  035570  052120  047440  041503
 7881  035576  051125  042522  027104
 7882  035604  005015     000
 7883  035607     040  020040     000  SPACE:  .ASCIZ  '  '
 7884  035613     015  042012  030510  DEVMAP: .ASCII  <15><12>'DH11, DM11-BB DEVICE MAP:'<15><12>
 7885  035620  026061  042040  030515
 7886  035626  026461  041102  042040
 7887  035634  053105  041511  020105
 7888  035642  040515  035120  005015
 7889  035650  005015  044104  030461          .ASCII  <15><12>'DH11   DH11   DM11-BB   DM11-BB'
 7890  035656  020040  042040  030510
 7891  035664  020061  020040  046504
 7892  035672  030461  041055  020102
 7893  035700  020040  046504  030461
 7894  035706  041055     102
 7895  035711     015  040412  051104          .ASCIZ  <15><12>'ADRS   VECT    ADRS      VECT'<15><12><15><12>
 7896  035716  020123  020040  042526
 7897  035724  052103  020040  020040
 7898  035732  042101  051522  020040
 7899  035740  020040  020040  042526
 7900  035746  052103  005015  005015
 7901  035754     000
 7902  035755     015  042012  030515  MSG4:   .ASCIZ  <15><12>'DM11-BB ERROR, RUN DZDHK DIAGNOSTIC'<15><12>
```

```
 7903  035762  026461  041102  042440
 7904  035770  051122  051117  020054
 7905  035776  052522  020116  055104
 7906  036004  044104  020113  044504
 7907  036012  043501  047516  052123
 7908  036020  041511  005015     000
 7909  036025     015  047012  020117  MSG5:   .ASCIZ  <15><12>/NO DM11-BB FOUND/<15><12>
 7910  036032  046504  030461  041055
 7911  036040  020102  047506  047125
 7912  036046  006504  000012
 7913                                  .EVEN
 7914                                  ;SIXTEEN CHAR COUNTERS USED BY THE AUTO ECHO TEST #3
 7915
 7916  036052  000020                  RCNT:   .BLKW   16.
 7917
 7918                                  ;256. WORD RECEIVER INPUT BUFFER
 7919
 7920  036112  000400                  RBUF:   .BLKW   256.
 7921
 7922
 7923                                  ;256(10) BYTE TRANSMITTER OUTPUT DATA BUFFER
 7924
 7925                                  .EVEN
 7926  037112  000400                  TBUF:   .BLKB   256.
 7927
 7928          000001                  .END
```

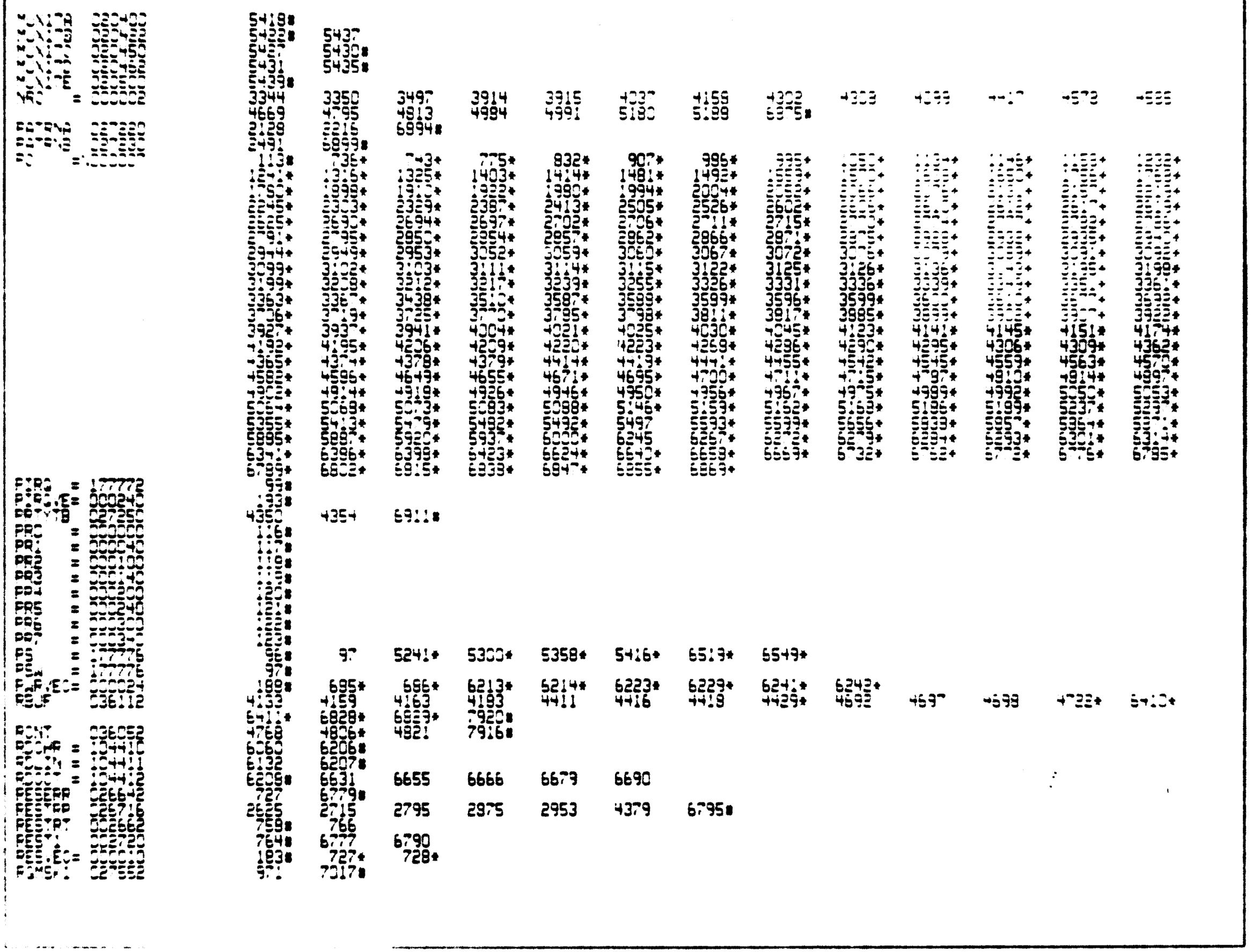
```
ABASE = 000000            258     299
ADDW1 = 000000            258     301
ADDW2 = 000000            258     302
ADFJOP= 000000            258     273
ADDW0 = 000000            259     303
ADDW1 = 000000            258     304
ADDW10= 000000            258     313
ADDW11= 000000            258     314
ADDW12= 000000            258     315
ADDW13= 000000            258     316
ADDW14= 000000            258     317
ADDW15= 000000            258     318
ADDW2 = 000000            258     305
ADDW3 = 000000            258     306
ADDW4 = 000000            258     307
ADDW5 = 000000            258     308
ADDW6 = 000000            258     309
ADDW7 = 000000            258     310
ADDW8 = 000000            258     311
ADDW9 = 000000            258     312
ADEVCT= 000000            258     264
ADEVM = 000000            258     300
ADPTR   030216            752*    759*    767     5456*   7201*
ADRVEC  030176            741     6599*   7190*
AENV  = 000000            258     269
AENVM = 000000            258     270
AETAB   027310           4532    4639    4767    4807    4811    4812    6931*
AETABO  027350           4602    6951*
AFATAL= 000000            258     261
AMADR1= 000000            258     286
AMADR2= 000000            258     290
AMADR3= 000000            258     293
AMADR4= 000000            258     296
AMAMS1= 000000            258     280
AMAMS2= 000000            258     288
AMAMS3= 000000            258     291
AMAMS4= 000000            258     294
AMSGAD= 000000            258     266
AMSGLG= 000000            258     267
AMSGTY= 000000            258     260
AMTYP1= 000000            258     281
AMTYP2= 000000            258     289
AMTYP3= 000000            258     292
AMTYP4= 000000            258     295
APASS = 000000            258     263
APRIOR= 000000            258
APTCSU= 000040           5840    5945*
APTENV= 000001           5596    5833    5901    5943*
APTSIZ= 000200            710    5942*
APTSPC= 000100           5835    5903    5944*
ASWREG= 000000            258     271
ATESTN= 000000            258     262
AUNIT = 000000            258     265
AUSWR = 000000            258     272
AUTOSZ  025076            736    6428*
AVEC*1= 000000            258     297
```

```
AVECT2= 000000   258    298
BAR   = 000012  3065*   3096    3100    3202*   3686*   3779*   3893*   4015*   4135*   4188    4280*   4371*   4439
                4445    4553*   4665*   4780*   4908*   4940*   4944    4953    5058*   5062    5069    5071    5153*
                6879*
BCR   = 000010  1895    2051    2215    23[illegible]   3063*   3119    3124    3227    3238    3683*   3776*   3891*   4011*
                4131*   4202    4277*   45[illegible]7*   4660*   4770*   4906*   4937*   5056*   5[illegible]51*   6381*   6393*   6408*
                6826*   6878*
BEGIN   002146   47     669*
BEGINA  002150   670*   6645    6648
BIT0  = 000001   179*    734     739    6651    6662    6675    6686    6714    6744
BIT00 = 000001   169*    179    6971
BIT01 = 000002   168*    178    6972
BIT02 = 000004   167*    177    6973
BIT03 = 000010   166*    176    6974
BIT04 = 000020   165*    175    6975
BIT05 = 000040   164*    174    6976
BIT06 = 000100   163*    173    2692    3330    6977
BIT07 = 000200   162*    172    2693    3078    3702    3796    6978
BIT08 = 000400   161*    171    5535    6979
BIT09 = 001000   160*    170    1125    1127    1136    1139    1150    2691    2771    2851    2929    4134    55[illegible]3
                5607    6980
BIT1  = 000002   178*   6590
BIT10 = 002000   159*   2853    5222    6981
BIT11 = 004000   158*    903    2603    2688    2768    2848    2926    3061    31[illegible]    3197    3328    3420    3483
                3572    3681    3774    3887    4009    4128    4270    4361    4389    4395    4415    4446    4460
                4531    4654    4765    4901    4934    5052    5148    5550    6982
BIT12 = 010000   157*   2773    6983
BIT13 = 020000   156*   2852    2930    3066    5591    6984
BIT14 = 040000   155*   2772    3897    3903    3919    4392    5179    5521    6985
BIT15 = 100000   154*   2931    3214    3329    3423    3427    3488    3491    3576    3580    3704    3796    4567
                4596    4727    4824    5086    5166    5179    6986
BIT2  = 000004   177*
BIT3  = 000010   176*
BIT4  = 000020   175*
BIT5  = 000040   174*
BIT6  = 000100   173*   3618
BIT7  = 000200   172*
BIT8  = 000400   171*   6696
BIT9  = 001000   170*
BKR   = 000014   894    1300    1407*   1541    1643*   2516*   4279*   4939*   6880*
BPTVEC= 000014   186*
BRKTAB  027512  4887    7000*
BR_VL   027672   754    7072*
BRPTR   030222   754*    761*    769    5458*   7203*
BUSER   026566   725    6766*
CAR   = 000006  1753    1979    2127    2300    2500*   2510*   2517    3107    3113    3243    3254    3634*   3777*
                3890*   4012*   4130*   4216    4278*   4548*   4549*   4658*   4659*   4[illegible]59*   4905*   4936*   5055*
                5150*   6380*   6407*   6825*   6877*
CHKADR  026340  6658    6704*
CHKVCT  026454  6669    6734*
CHPS1   027076  2606    2694    2774    2854    2932    3067    3136    3331    3367    4374    4378    6776    6789
                6844*
CHPS2   027112  2602    2690    2770    2850    2928    3060    3326    4365    6852*
CKSWR = 104407  5518    5583    5606    6205*
CLCABC  024654  3059    3198    4545    4902    5053    6378*
CLRMUX= 002000  5243    7216*
```

```
CO     = 000100          5312    7214*
CR     = 000015            94*   5879    5889
CRLF   = 000200            95*   5850    5889
CS     = 000040          5312    7213*
DDISP  = 177570           101*    222     698
DEVMAP  035613           6593    7884*
DHADR   027172            767*   4127    4186    4357    4653    5520    6340*   6883*
DHADRS  027732           6430    6438    6485    6501    6594    6650    6716    7091*
DHADTB  027570            752    6713    6727    7028*
DHMCSR  030200           5229*   5230    5232    5240*   5243*   5245*   5247*   5250    5261*   5265*   5299*   5303*   5305*
                         5308    5319*   5324*   5357*   5361*   5363*   5366    5377*   5382*   5415*   5419*   5421*   5424
                         5435*   5440*   7191*
DHMLSR  030202           5230*   5231*   5246*   5249    5267*   5270    5296*   5304*   5317    5325*   5328    5354*   5362*
                         5365    5383*   5386    5412*   5420*   5423    5441*   5444    7192*
DHNUM   030210            755*    758*    771    5225    5455*   7196*
DHRLVL  030206            769*   2590    2684    2764    2844    2922    3325    4368    7194*
DHSEL   027200            762    5461    6682*   6886*
DHTLVL  030207           2593    2687    2767    2847    2925    3051    7195*
DHVCT   027174            768*   2588    2682    2762    2842    2920    3048    3131    3323    3359    4366    6795    6884*
DHVCTB  027630            753    6743    6757    7048*
DHVEC   027774           6502    6583    6584    6586    6595    6661    6746    7112*
DH1     030317            341    7244*
DH10    033671            602     609    7679*
DH11    034176            623    7723*
DH12    034324            637     644    7745*
DH13    034421            651    7759*
DH14    033122            539    7585*
DH2     030452            348     355     362     369     376     383     390     397     404     411     418     425     432
                          453     460     475     482     489     525     560     581     588     595     616     630     658
                          665    7264*
DH3     032323            496     503    7497*
DH4     032523            510     532     553    7527*
DH5     032662            517     546     567     574    7547*
DH6     031512            439     446    7401*
DH7     032032            467    7451*
DISPLA  001142            222*    698*    706*   5565*   5586*
DISPRE  000174             43*    706
DMADRS  030034           5225    6437    6543    6596    7132*
DMVEC   030076           6544    6597    7153*
DOIT11  017574           5233    5237*   5272
DOIT15  017764           5296*   5330
DOIT16  020156           5354*   5388
DOIT17  020350           5412*   5446
DSWR   = 177570           100*    221     697
DT1     030376            342    7253*
DT2     030550            349     356     363     370     377     384     391     398     405     412     419     426     433
                          440     447     454     461     468     476     483     490     511     526     533     554     561
                          582     589     596     617     631     638     645     652     659     666    7276*
DT3     032402            497     504    7506*
DT4     032710            518     547     568     575    7552*
DT5     033766            603     610     624    7691*
DT6     033152            540    7590*
EMTVEC= 000030            189*    681*    682*
EM1     030250            340    7237*
EM10    031050            389    1925    2058    4212    7327*
EM11    031117            396    7337*
```

```
EM12    031156            403    7346#
EM13    031217            410    7355#
EM14    031300            417    3142    7367#
EM15    031347            424    3082    7377#
EM16    031415            431    7387#
EM17    031446            438    3728    7395#
EM2     030414            347    7259#
EM20    031610            445    3820    7415#
EM21    031651            452    7424#
EM22    031715            459     643     650    3801    4033    4154    4293    7434#
EM23    031771            466    4048    7445#
EM24    032127            474    4573    4589    7465#
EM25    032167            481    4929    4959    4978    4995    7474#
EM26    032227            488    5076    5091    7483#
EM27    032271            495    7492#
EM3     030572            354    7282#
EM30    032420            502    7512#
EM31    032453            509    4703    4718    4817    7520#
EM32    032620            516    4678    7541#
EM33    032720            524    4312    7557#
EM34    032763            531    4422    4425    7566#
EM35    033056            538    7579#
EM36    033162            545    7595#
EM37    033211            552    4177    7602#
EM4     030650            361    7293#
EM40    033247            559    4198    7611#
EM41    033312            566    7620#
EM42    033337            573    7627#
EM43    033363            580    7634#
EM44    033443            587    7646#
EM45    033525            594    7658#
EM46    033603            601    2160    2332    7669#
EM47    034010            608    2248    2415    7697#
EM5     030706            368    7301#
EM50    034071            615     636    3220    3709    5171    7708#
EM51    034147            622    7719#
EM52    034273            629    7737#
EM56    034520            657    5192    7773#
EM57    034565            664    3910    3930    3944    7783#
EM6     030743            375    7309#
EM7     030776            382    1783    1986    2529    4226    7317#
ENDA    020540           5224    5236    5414    5453#   5462
ERRVEC= 000004            182#    695     696*    707*    725*    726*    822     823*    838*    5526    5527*   5529*   5532*
EXPAR   026334           6697    6701#
GNS     = ****** U         42    6197    6198    6199    6200    6201    6203    6205    6206    6207    6208
GTSWR   = 104406          721    6203#
HT      = 000011           92#    5848    5889
INMSG1  034731           6654    7804#
INMSG2  034775           6665    7811#
INMSG3  035044           6678    7818#
INMSG4  035115           6731    7825#
INMSG5  035166           6761    7832#
INMSG6  035242           6689    7840#
INMSG7  035304           6699    7846#
INPAR   026126            750    6650#
INPARA  026046            743    6629#   6638
```

```
INPARC  026104            48   6644#
INPARX  026114            46   6647#
INPAR1  026162          6657   6661#
INPAR3  026216           749   6668   6672#
INPAR4  026254          6681   6684#
INTMSK  027566          2595   7023#
IOTVEC= 000020           197#   679*   680*   6503*  6504*  6545*  6571*
LDBCR   024716          3199   6391#
LDTBF1  024266           775   6261#
LF      = 000012          93#  5883   5889
LINACT  027450          4370*  4432*  4453   4459   4664*  4724*  4725   4779*  4825*  6836*  6988#
LINBIT  027410          4432   4724   4825   6971#
LINE    030212          1758   1762   1770   1774   1782   1790*  1900   1904   1912   1916   1924   1932*  1983
                        1985   2055   2057   2133   2151   2221   2239   2305   2323   2389   2407   2507   2520
                        3058   3062   3081   3141   3200   3215   3219   3228   3235   3244   3251   3682   3705
                        3708   3727   3775   3797   3800   3819   3889   3904   3909   3924   3929   3943   4010
                        4029   4032   4039   4047   4129   4150   4153   4167   4176   4197   4211   4225   4272
                        4274   4293   4297   4311   4546   4569   4572   4588   4601*  4657   4677   4687   4717
                        4903   4928   4935   4958   4977   4994   5054   5075   5087   5090   5149   5165   5170
                        5176   5191   5242   5301   5359   5417   5519*  6326   6328*  6330*  6332*  6339*  7197#
LINEA   030214          2133*  2142   2221*  2230   2305*  2314   2389*  2398   6307   7198#
LINENA= 000001          5246   5254   5304   5312   5322   5362   5370   5380   5420   5428   5438   7208#
LINMSK  027204          3065   3202   3686   3779   3893   4015   4135   4279   4280   4553   4664   4909   4939
                        4940   5058   5153   6329*  6333*  6335   6888#
LINSEL  027202          1754   1755*  1793*  1896   1897*  1935*  4364   4370   4371   4407   4540   4647   4665
                        4689   4779   4780   4804   4895   6335   6693*  6695*  6836   6887#
LMSK1   027206          4533*  4537*  4540   4640*  4644*  4647   4888*  4892*  4895   6889#
LPR     = 000004          895  1152*  1216   1396*  1463   1631*  2509*  3064*  3201*  3685*  3778*  3892*  4013*
                        4014*  4132*  4276*  4363   4552*  4564*  4583*  4596*  4661*  4675   4727*  4771*  4791
                        4824*  4907*  4938*  5057*  5152*  6409*  6827*  6876#
MSG1    035451          6487   7865#
MSG2    035501          6527   7870#
MSG3    035546          6556   7877#
MSG4    035755          5259   5274   5317   5332   5375   5390   5433   5448   7902#
MSG5    036025          5235   7909#
MSTCLR  027210           888    900   6891#
MULPTB  027452          4430*  6418   6420   6995#
MUX11   017602          5240#
MUX11A  017620          5243#
MUX11B  017650          5248#  5263
MUX11C  017676          5253   5256#
MUX11D  017710          5257   5261#
MUX11E  017724          5265#
MUX15   017776          5299#
MUX15A  020014          5302#
MUX15B  020036          5306#  5311
MUX15C  020064          5311   5314#
MUX15D  020076          5315   5319#
MUX15E  020114          5323#
MUX16   020170          5357#
MUX16A  020206          5360#
MUX16B  020230          5364#  5379
MUX16C  020256          5369   5372#
MUX16D  020270          5373   5377#
MUX16E  020306          5381#
MUX17   020362          5415#
```



```
R3MSK2  027554          1037    7018#
R3MSK3  027556          1218    1465    1471    1475    7019#
R3MSK4  027560          1302    1543    1549    1552    7020#
R3MSK5  027562          1388    1621    1627    1630    7021#
R3MSK6  027564          1119    7022#
R4      =000200         537     7215#
R5      =000004         5362    5370    5383    7210#
R5TMP   =000130          763     767#   5464    6246
R7      =000016          104#    669*    741*    745     747    5225*   5228*   5229    5489*   5492    5628    5629*   5630*
                        5637*   5638*   5639*   5640*   5641*   5642    5647    5652*   5654*   5658    5660    5756    5764*
                        5786    5788    5800*   5831    5832*   5837    5842    5845*   5897    5905*   5909    5910
                        5910*   5913*   5914    5936*   6129    6133*   6134    6137    6157*   6160*   6174    6175*   6176
                        6178*   6179*   6179*   6180*   6215    6240*   6273*   6274    6285*   6296    6298    6308*   6309
                        6364    6365*   6369*   6360    6367*   6368*   6369    6372*   6428    6491*   6492*   6493*   6497
                        6553*   6555*   6552*   6554*   6572*   6632*   6634    6636    6639*   5644*   6673    6684    6756
                        6770*   6783*
R:      =000001          105#    819     889     990     891     892     903*    968     980*    990*    1035    1045*   1116
                        1138*   1140*   1152*   1215    1226*   1236*   1299    1310*   1320*   1385    1396*   1407*   1462
                        1473*   1486*   1540    1553*   1564*   1618    1631*   1643*   1752    1762*   1763*   1774*   1775*
                        1894    1904*   1905*   1916*   1917*   1978    1983*   1988*   1999*   2050    2055*   2060*   2071*
                        2126    2136*   2142*   2143*   2148*   2149*   2214    2224*   2230*   2231*   2236*   2237*   2299
                        2308*   2314*   2315*   2320*   2321*   2383    2392*   2398*   2399*   2404*   2405*   2489    2497*
                        2498*   2499*   2500*   2507*   2508*   2509*   2510*   2514*   2515*   2516*   2517    2587    2603*
                        2605*   2611    2618    2681    2688*   2691*   2692*   2693*   2698    2699*   2700*   2707    2709*
                        2767*   2768*   2771*   2772*   2773*   2778    2779*   2780*   2787    2793*   2790*   2841
                        2884*   2885*   2892*   2893*   2894*   2895*   2896*   2897*   2898*   2899*   2900*   2919    2926*
                        2930*   2931*   2936    2937*   2938*   2945    2947*   2948*   3056    3061*   3062*   3063*   3064*
                        3066*   3066*   3086    3090    3096    3107    3119    3130*   3197*   3200*   3201*   3202*
                        3213    3214    3215    3225*   3227    3233    3243    3253    3328*   3329*   3330*   3337    3343
                        3388*   3389*   3411    3420*   3480    3483*   3497    3503    3572*   3582    3684*   3682*   3683*
                        3885*   3886*   3890    3701    3703    3774*   3775*   3776*   3777*   3778*   3779*   3793
                        3795    3887*   3888    3899*   3890*   3991*   3992*   3993*   3997    3905    3914    3915
                        3925    3934    3940    4009*   4010*   4011*   4012*   4013*   4014*   4015*   4019    4036
                        4127*   4128*   4129*   4130*   4131*   4132*   4134*   4135*   4139    4146    4158    4153*
                        4182*   4183    4186*   4187    4231    4215    4270*   4274*   4276*   4277*   4279*   4283*
                        4284    4291    4294    4302    4307    4357*   4361*   4363    4369*   4371*   4385    4389*
                        4395*   4399    4415*   4417*   4439    4445    4446*   4460*   4531*   4546*   4547*   4548*
                        4552*   4553*   4557    4564*   4565    4566    4578    4583*   4594    4596*   4653*   4654*
                        4658*   4659*   4660*   4661*   4665*   4669    4675    4699*   4714*   4723*   4729    4730*
                        4769*   4770*   4771*   4774*   4780*   4785    4791    4795    4813*   4823*   4824*   4831*   4833*
                        4905*   4906*   4907*   4908*   4912    4920    4934*   4935*   4936*   4937*   4939*   4939*   4940*
                        4944    4952    4964    4969    4984    4990    5052*   5054*   5055*   5056*   5057*   5058*   5063
                        5069    5070    5080    5084    5085    5148*   5149*   5150*   5151*   5152*   5153*   5154*   5160
                        5167    5180    5187    5242*   5246    5252    5265    5266    5301*   5303    5304    5305    5310
                        5359*   5361    5368    5382    5417*   5419    5426    5439    5440*   5441*   5442*   5443*   5470*
                        5771    5775    5799*   5898    5935*   6130    6135*   6143*   6145*   6147*   6150*   6153    6156*
                        6216    6239*   6261*   6263*   6275    6287    6299    6310    6355    6359*   6365*   6369*   6371*
                        6379*   6380*   6381*   6385*   6392*   6393*   6397*   6403*   6407*   6408*   6409*   6412*   6413*
                        6440*   6442    6443    6444*   6446*   6447*   6448    6451*   6453    6457*   6458    6459    6460*
                        6472*   6473    6474    6476    6478*   6532*   6533*   6534    6561*   6562*   6563    6593*   6596*
                        6588*   6589    6594*   6599    6619    6650*   6656*   6661*   6667*   6672*   6680*   6683*   6691*
                        6693    6704    6707    6710    6725    6726*   6734    6737    6740    6755    6756*   6759*   6791*
                        6824*   6825*   6826*   6827*   6830*   6831*
R2      =000002          1068#    819*    820     824*    826*    827*    829     899*    902*    904     968*    979*    981*
                         982     991*    992    1035*   1044*   1046*   1047    1116*   1126*   1127*   1129*   1130    1133*
                        1141*   1142    1151*   1153*   1154    1215*   1216*   1225*   1227*   1229    1237*   1233    1235*
```

1300* 1309* 1311* 1312 1321* 1322 1385* 1386* 1395* 1397* 1398 1408* 1409
1460* 1463* 1472* 1475* 1476* 1477 1487* 1488 1540* 1541* 1551* 1552* 1554*
1555 1565* 1566 1618* 1619* 1629* 1630* 1632* 1633 1644* 1645 1752* 1753*
1894* 1896* 1894* 1895* 1906* 1918 1978* 1979* 1989* 1990 2000* 2001* 2005*
2061* 2061* 2062 2072* 2073 2126* 2127* 2137* 2144* 2150 2214* 2215* 2225*
2238 2238 2299* 2300* 2309* 2316* 2322 2383* 2384* 2393* 2400* 2406 2489*
2518 2518 2587* 2681* 2761* 2841* 2919* 3056* 3100* 3113* 3124* 3216* 3237*
3253* 3253* 3254* 3350* 3362* 3417* 3418* 3423* 3427* 3432 3480* 3481* 3488*
3503 3503 3569* 3570* 3574* 3576* 3580* 3604 3703* 3720* 3721* 3722* 3723*
3795* 3795* 3812* 3813* 3814* 3815* 3816* 3888* 3939* 4007* 4014 4051* 4052*
4147* 4147* 4162* 4164 4181* 4187* 4188* 4189 4193 4201* 4202* 4203 4227*
4216* 4216* 4217* 4221 4294* 4307* 4308* 4366* 4367* 4368* 4400* 4401* 4402*
4403* 4403* 4409 4411 4416 4418* 4429* 4430* 4432 4565* 4584* 4585* 4586*
4699 4699 4692 4697 4698* 4712* 4722* 4724 4767* 4769 4773* 4796* 4797*
4799 4799 4800* 4804 4806* 4807 4811 4812* 4821 4825 4830* 4831* 4832*
4963* 4963* 4954 4969* 4970* 4971 4990* 4991* 5070* 5071* 5084* 5156* 5157*
5244* 5244* 5262* 5302* 5320* 5360* 5378* 5418* 5436* 5458 5459* 5473* 5474*
5784* 5784* 5785 5790* 5798* 6131 6136* 6144* 6146* 6148* 6154 6155* 6156*
6262* 6262* 6263 6264* 6265 6275 6288 6300 6311 6355 6360* 6361* 6362*
6364* 6364* 6365* 6366 6370* 6404* 6410* 6414* 6430* 6431* 6438* 6439* 6440*
6494* 6494* 6495 6501* 6516 6520* 6521* 6528* 6529* 6537* 6538* 6539* 6540*
6551* 6551* 6557* 6558* 6566* 6567* 6595* 6603 6613* 6619* 6625* 6627 6643*
6755* 6755* 6757 6769* 6782* 6821* 6828* 6832*
R3 =000003 907* 982* 983 992* 1047* 1130* 1131 1142* 1143 1154* 1155 1220*
1238* 1238* 1312* 1313 1322* 1398* 1399* 1400 1409* 1410* 1411 1477* 1478*
1489 1489 1555* 1556 1566* 1567 1633* 1634* 1635 1645* 1646* 1647 1765*
1918* 1919 1919 1990* 1991 2001* 2062* 2063 2073* 2150* 2154 2233* 2234*
2326 2326 2611* 2665 2680 2681* 2682* 2683* 2684* 2685 2686* 2687* 2688*
2765* 2766* 2766* 2767* 2768* 2769* 2770* 2771* 2772* 2773* 2774* 2775* 2776*
2821* 2821* 2822* 2823 2824* 2825* 2826* 2827* 2828* 2829* 2830* 2831* 2832*
3090* 3090* 3096* 3107* 3108 3109* 3110* 3111* 3112* 3113* 3114* 3115* 3116*
3243* 3243* 3247 3323* 3324* 3325* 3326* 3327* 3328* 3329* 3330* 3331* 3332*
3369* 3369* 3370* 3371* 3372* 3373* 3374* 3375* 3376* 3377* 3378* 3379* 3380*
3523* 3523* 3504* 3507 3527* 3528* 3529* 3530* 3531* 3532* 3533* 3534* 3535*
3905* 3905* 3906* 3925* 3926* 3927* 3928* 3929* 3930* 3931* 3932* 3933* 3934*
4193* 4193* 4220* 4221* 4289* 4302* 4303* 4304* 4305* 4306* 4307* 4308* 4309*
4669* 4669* 4682 4692 4708 4750* 4751* 4752* 4753* 4754* 4755* 4756* 4757*
4972* 4972* 4984* 5069* 5085* 5086* 5087* 5088* 5089* 5090* 5091* 5092* 5093*
5266* 5266* 5308* 5309* 5310* 5311* 5312* 5313* 5314* 5315* 5316* 5317* 5318*
5399 5399 5414* 5415* 5416* 5417* 5418* 5419* 5420* 5421* 5422* 5423* 5424*
5785* 5785* 5794* 5795* 5796* 5797* 5798* 5799* 5800* 5801* 5802* 5803* 5804*
6086* 6086* 6092 6094* 6095* 6096* 6097* 6098* 6099* 6100* 6101* 6102* 6103*
6432* 6432* 6433 6437* 6438* 6439* 6440* 6441* 6442* 6443* 6444* 6445* 6446*
6716* 6716* 6719 6746* 6749 6795* 6796* 6797* 6798* 6799* 6800* 6801* 6802*
6831* 6831*
R4 =000004 883 989* 1043* 1124* 1125* 1131 1139* 1143 1150* 1151* 1152*
1235* 1235* 1313 1313* 1314* 1315* 1316* 1317* 1318* 1319* 1320* 1321* 1322*
1549* 1549* 1555 1556* 1557* 1558* 1559* 1560* 1561* 1562* 1563* 1564* 1565*
1772* 1772* 1765* 1766* 1767* 1768* 1769* 1770* 1771* 1772* 1773* 1774* 1775*
2000 2000 1938* 1939* 1940* 1941* 1942* 1943* 1944* 1945* 1946* 1947* 1948*
2054* 2054* 2319* 2320* 2321* 2322* 2323* 2324* 2325* 2326* 2327* 2328* 2329*
2325* 2326* 2888* 2889* 2890* 2891* 2892* 2893* 2894* 2895* 2896* 2897* 2898*
2661* 2668* 2888* 2889* 2890* 2891* 2892* 2893* 2894* 2895* 2896* 2897* 2898*
2922* 2923* 2924* 2925* 2926* 2927* 2928* 2929* 2930* 2931* 2932* 2933* 2934*
3107* 3108 3109* 3110* 3111* 3112* 3113* 3114* 3115* 3116* 3117* 3118* 3119*
3323* 3324* 3325* 3326* 3327* 3328* 3329* 3330* 3331* 3332* 3333* 3334* 3335*
3371* 3372* 3373* 3374* 3375* 3376* 3377* 3378* 3379* 3380* 3381* 3382* 3383*
3507 3508 3509* 3510* 3511* 3512* 3513* 3514* 3515* 3516* 3517* 3518* 3519*
3925* 3926* 3927* 3928* 3929* 3930* 3931* 3932* 3933* 3934* 3935* 3936* 3937*
4221* 4289* 4302* 4303* 4304* 4305* 4306* 4307* 4308* 4309* 4310* 4311* 4312*
4682 4708 4750* 4751* 4752* 4753* 4754* 4755* 4756* 4757* 4758* 4759* 4760*
4984* 5069* 5085* 5086* 5087* 5088* 5089* 5090* 5091* 5092* 5093* 5094* 5095*
5308* 5309* 5310* 5311* 5312* 5313* 5314* 5315* 5316* 5317* 5318* 5319* 5320*
5414* 5415* 5416* 5417* 5418* 5419* 5420* 5421* 5422* 5423* 5424* 5425* 5426*
5794* 5795* 5796* 5797* 5798* 5799* 5800* 5801* 5802* 5803* 5804* 5805* 5806*
6092 6094* 6095* 6096* 6097* 6098* 6099* 6100* 6101* 6102* 6103* 6104* 6105*
6433 6437* 6438* 6439* 6440* 6441* 6442* 6443* 6444* 6445* 6446* 6447* 6448*
6719 6746* 6749 6795* 6796* 6797* 6798* 6799* 6800* 6801* 6802* 6803* 6804*
6831* 6831*

```
                    3924*  3938*  4028*  4023*  4038*  4039*  4040*  4041*  4042   4149*  4150*  4164*  4165*
                    4166*  4167*  4168*  4169*  4171   4194*  4202*  4222*  4271*  4272*  4273*  4292*  4293*
                    4303   4416*  4539*  4568*  4569*  4579   4682*  4683*  4684*  4685   4687   4696   4697*
                    4713*  4811*  4894*  4919   4924*  4925*  4930*  4951   4955*  4960*  4968   4973*  4974*
                    4979*  4985   5072*  5097*  5165*  5166*  5176*  5177*  5178*  5179*  5183   5249*  5256
                    5270*  5271   5307*  5314   5329*  5329   5365*  5372   5386*  5397   5423*  5430   5444*
                    5445   5500   5702*  5703*  5704*  5705   5706*  5720   5722*  5730*  5733*  6219   6236*
                    6290   6313   6406*  6411*  6415*  6418*  6419*  6420   6506*  6509   6510*  6511   6512
                    6574*  6576   6579*  6579   6580   6597*  6613   6718*  6721*  6722   6748*  6751*  6752
                    6823*  6829*  6833*
RS      =000005      105*   106*   820*   821*   827    888*   893*   890*   891*   892*   893*   894*   895*
                     898    899    900    969*   971    973*   978    981    991    1036*  1037   1039*  1046
                    1117*  1118   1120*  1124   1129   1141   1153   1217*  1218   1220*  1224   1227   1237
                    1301*  1302   1304*  1308   1311   1321   1387*  1388   1390*  1394   1397   1408   1464*
                    1465   1466*  1474   1476   1485   1487   1542*  1543   1545*  1550   1554   1563   1565
                    1620*  1621   1623*  1628   1632   1642   1644   1781*  1923*  1984*  2056*  2128*  2129
                    2158*  2216*  2217   2246*  2301*  2302   2309   2325   2330*  2385*  2386   2393   2409
                    2414*  2491*  2492   2493   2525*  2594*  2595   2597*  2604   2605   3080*  3140*  3218*
                    3342*  3365   3421*  3429*  3484*  3485*  3505   3573*  3574   3575*  3581*  3584   3593
                    3697   3700*  3726*  3799*  3816*  3908*  3916*  3917*  3928*  3942*  4006*  4008   4031*
                    4046*  4133*  4156*  4158*  4159   4175*  4196*  4210*  4224*  4296*  4310*  4350*  4352*
                    4354   4358   4359   4420*  4423*  4532*  4539   4548   4571*  4587*  4602*  4639*  4646
                    4658   4676*  4701*  4716*  4766*  4775*  4815*  4887*  4894   4927*  4957*  4976*  4993*
                    5074*  5099*  5169*  5190*  5248*  5254*  5256   5264*  5306*  5312*  5314   5322*  5329
                    5364*  5370*  5372   5380*  5387   5422*  5428*  5430   5438*  5445   5701   5707*  5709*
                    5711*  5712*  5713*  5714   5732*  5760   5762*  5764*  5771*  5775*  5790   5796*  6220
                    6235*  6357   6358   6373*  6502*  6534*  6544*  6563*
R6      =000006      111*   113*   674*   675    6277   6291
R7      =000007      111*
SP%S    =000006      2610   2617   2697   2706   2777   2786   2957   2966   2935   2944   3076   3089   3099
                     3100   3108   3232   3336   3349   3361   3587   3595   3719   3811   3902   3922   3937
                     4085   4145   4182   4206   4220   4230   4306   4414   4563   4582   4695   4711   4910
                     4958   4959   4967   4989   5068   5083   5162   5196   6272   6284   6863*
                     5408   5418*
                     5420   5428   5441   7211*
                     5766   5769   5898   1910   1980   2052   2131   2219   2303   2387   2505   3052   3135
                     3677   3700   3885   4004   4123   4268   4542   4649   4897   5050   5146   5237   5297
                     5355   5413   6326*
                     5566*  5622   5664*  5459*  5461   6885*
                     6565   6820*
                     1128*  6777*  6795*  7203*  7207   9228*  9338   2601*  2614*  2621*  2624*  2693*
                     2744*  2769*  2794*  2849*  2874*  2927*  3352*  3055*  3137*  3327*  3366*  4356*  4373*
                     5486*  5526*  5529   5531   5532   5561   5562   5566*  5588   5609*  5613*  5628*  5633*
                     5654   5658*  5691*  5692   5693   5694*  5699*  5700*  5701*  5732   5733   5790*  5734
                     5735*  5736*  5756*  5757*  5758*  5759*  5760*  5761*  5762   5765*  5778   5930*  5742*
                     5792   5794   5796   5797   5798   5799   5800   5802*  5803*  5831*  5832   5842*  5843*
                     5845   5846*  5848   5850   5852   5859   5860*  5862*  5870*  5874   5878   5879   5893*
                     5897*  5898*  5905   5906*  5917   5918*  5919*  5929   5930*  5935   5936   5960*  5966*
                     5962   5969*  5972*  5973*  5977*  5978*  5982   5985*  5989   5991   5993   5994*  6033
                     6003   6005*  6006   6008*  6009*  6010*  6011*  6012*  6027*  6028*  6031*  6032*  6033*
                     6037*  6038*  6039   6042   6044   6046*  6055*  6056*  6061   6064   6068*  6075   6079*
                     6098   6099   6100*  6101*  6102*  6127*  6128*  6129*  6130*  6131*  6133   6137*  6139*
                     6141   6149*  6150   6152   6153*  6155   6156   6157   6159   6174*  6175   6185*  6186*
                     6215*  6216*  6217*  6218*  6219*  6220*  6221*  6222   6230*  6234   6235   6236   6237*
                     6238   6239   6240   6245*  6337*  6354*  6355*  6356*  6370   6371   6372   6429*  6430*
                     6466*  6481*  6484   6532   6535   6536*  6561   6564   6565*  6572   6593*  6603*  6609*
```

```
                    6613*  6632   6656   6667   6680   6691   6712*  6742*  6767   6768   6769   6771*  6780
                    6791   6782   6794*  6814*  6844*  6845*  6852*  6953*  6980*  6981*  6984*  6986*  6987
                    6988
SPACE   = 035607    6607   6612   7883#
SSR     = 000016     693    980*   990*  1045*  1128*  1140*  1226*  1236*  1310*  1320*  1386   1473*  1486*
                    1553*  1564*  1619   1763*  1775*  1905*  1917*  1988*  1999*  2060*  2071*  2143*  2149*
                    2231*  2237*  2315*  2321*  2399*  2405*  2490   3329*  3343   3362   3418   3481   3570
                    3934   3939   3940   4728   4912   4921   4964   4970   6881#
STACK   = 001100     87#    677   2601   2624   2689   2714   2769   2794   2849   2874   2927   2952   3055
                    3137   3327   3366   4356   4377   6771   6784
START1  002464       725#
START2  002634       746    752#   765   5498   6701
STEP    = 000400    5261   5319   5377   5435   7207#
STK_MT  = 177774      98#
SUBR1   024310       932   6272#
SUBR2   024350       907    986    995   1050   1134   1146   1158   1232   1241   1316   1325   1403   1414
                    1481   1492   1559   1570   1638   1650   1780   1922   1994   2004   2066   2076   2526
                    3438   3510   4045   4174   6284#
SUBR2A  024354      2612   2619   2722   2711   2782   2791   2862   2871   2940   2949   3079   3091   3102
                    3114   3125   3217   3339   3351   3363   3589   3600   3706   3725   3798   3817   3937
                    3927   3941   4030   4151   4195   4209   4223   4295   4309   4419   4570   4586   4700
                    4715   4814   4926   4956   4975   4992   5073   5088   5168   5189   6285#
SUBR3   024420      6298#  6772   6785
SUBR4   024436      2157   2245   2329   2413   3239   3255   6306#
SUNUM   024574       770   1781   1923   1984   2056   2158   2246   2330   2414   2527   3080   3140   3218
                    3707   3726   3799   3818   3908   3928   3942   4031   4046   4152   4175   4196   4213
                    4224   4296   4310   4420   4423   4571   4597   4676   4701   4716   4815   4927   4957
                    4976   4993   5074   5089   5169   5190   6353#
SUPPAR  024756      4362   6402#
SWR     001140       221#   675    697*   699    705*   712*   719    734    739   5222   5521   5535   5537
                    5543   5550   5591   5603   5607   5956   5993*  6221   6234*  6590   6651   6662   6675
                    6686   6696   6714   6744
S_REG   000176        44#   705    719   5956   5969
SW00    = 000001      151#
SW00    = 000001      141#   151
SW01    = 000002      140#   150
SW02    = 000004      139#   149
SW03    = 000010      138#   148
SW04    = 000020      137#   147
SW05    = 000040      136#   146
SW06    = 000100      135#   145
SW07    = 000200      134#   144
SW08    = 000400      133#   143
SW09    = 001000      132#   142
SW1     = 000002      150#
SW10    = 002000      131#
SW11    = 004000      130#
SW12    = 010000      129#
SW13    = 020000      128#
SW14    = 040000      127#
SW15    = 100000      126#
SW2     = 000004      149#
SW3     = 000010      148#
SW4     = 000020      147#
SW5     = 000040      146#
SW6     = 000100      145#
```

```
SW7    = 000200          144#
SW8    = 000400          143#
SW9    = 001000          142#
TBITVE= 000014           184#
TBUF    037112  4130    4162    4217    4222    4936    5055    5150    6261    6407    6825    7926#
TDATA1  030224  4275*   4278    7218#
TDATA2  030226  4006    7219#
TIMEA   030240  3069*   3204*   3688*   3781*   3895*   4017*   4137*   4282*   4437*   4451*   4555*   4667*   4782*
                4910*   4942*   5060*   5155*   6812*   7224#
TIMEB   030242  3070*   3205*   3689*   3713    3731    3782*   3805    3823    3896*   4018*   4138*   4293*   4438*
                4452*   4556*   4668*   4783*   4911*   4943*   5061*   5156*   6810*   7225#
TIMEC   030244  3680*   3714    3731*   3773*   3806    3823*   7226#
TIMEIT  026744  3072    3208    3692    3785    3899    4021    4141    4286    4441    4455    4559    4671    4737
                4914    4946    5064    5159    6810#
TITFLG  030236   670*    729     733*    7223#
TITLE   034634   732    7793#
TITLE2  034704   772     774    7800#
TNVEC = 000060           191#
TNULL   033246  4905    7227#
TPVEC = 000064           192#
TRAPVE= 000034           190#    683*    684*
TRMRDY= 000002  5304    5312    5325    7209#
TRTVEC= 000014           185#
TST1    003000           781#
TST10   004000  1305    1331#
TST11   004126  1391    1420#
TST12   004256  1468    1498#
TST13   004406  1546    1576#
TST14   004546  1624    1656#
TST15   004736  1797#
TST16   005126  1939#
TST17   005252  1981    2011#
TST2    003070           842#
TST20   005376  2053    2083#
TST21   005604  2130    2171#
TST22   006012  2218    2259#
TST23   006212  2304    2343#
TST24   006412  2388    2427#
TST25   006652  2494    2540#
TST26   007036  2629#
TST27   007216  2719#
TST3    003170           901     914#
TST30   007376  2799#
TST31   007556  2879#
TST32   007736  2957#
TST33   010370  3138    3147#
TST34   010716  3196    3265#
TST35   011150  3377#
TST36   011272  3449#
TST37   011446  3522#
TST4    003300           975    1001#
TST40   011662  3623#
TST41   012170  3678    3740#
TST42   012472  3771    3833#
TST43   013002  3886    3951#
TST44   013242  4005    4059#
```

```
TST45   013714          4235#
TST46   014146          4269    4318#
TST47   014706          4380    4466#
TST5    003362          1040    1057#
TST50   015260          4600    4609#
TST51   015730          4651    4735#
TST52   016260          4793    4819    4830#
TST53   017020          5003#
TST54   017234          5051    5097#
TST55   017476          5147    5199#
TST56   017762          5238    5281#
TST57   020154          5298    5339#
TST6    003550          1121    1164#
TST60   020346          5356    5397#
TST7    003664          1221    1247#
TYPDS = 104405          5487    6201#
TYPE  = 104401           731     773    5234    5258    5273    5316    5331    5374    5389    5432    5447    5485    5488
                        5594    5627    5644    5646    5649    5651    5655    5662    5726    5801    5853    5967    5968
                        5971    5984    5995    6014    6067    6073    6079    6082    6087    6099    6090    6093    6097
                        6161    6163    6197#   6243    6487    6527    6556    5592    6607    6612    6617    6623    6629
                        6653    6664    6677    6688    6698    6730    6760
TYPOC = 104402          5635    5659    5970    6198#
TYPON = 104404          6200#
TYPOS = 104403          5199#   6600    6604    6609    6614
VCFLG   027670           737     744*   6647*   7065#
VCPTR   030220           753*    760*    768    5457*   7202#
VCWC    035354          6630    7854#
$APTHD  000214            69      75#
$ASTAT= ******          5923    5938
$ATYC   022536          5894    5896#
$ATY1   022512          5892#
$ATY3   022520          5838    5893#
$ATY4   022530          5599    5895#
$AUTOB  001134           218#    723*   5964    6113
$BASE   001306           299#
$BDADR  001122           213#
$BDDAT  001126           215#
$CDW1   001312           301#
$CDW2   001314           302#
$CHARC  022506          5855*   5865*   5872    5881*   5886#
$CKSWR  022760          5956#   6205
$CMTAG  001100           201#    672     673     681     687     688     689
$CM1  = 000010           233#    234#    235#    236#    237#    238#    239#    240#    241#
$CM2  = 000020           233#    234#    235#    236#    237#    238#    239#    240#    241#
$CM3  = 000010           231#    233
$CM4  = 000010           241#    242#    243#    244#    245#    246#    247#    248#    249#
$CNTLG  023633          5967    6108#
$CNTLU  023626          5984    6082    6107#
$CPUOP  001260           273#
$CR_F   001227           252#   5594    5619    5627    5646    5651    5655    5854    5889    5995    6087    6107    6166
                        6618    6623
$CB_K   022220          5767    5801    5809#
$DDW0   001316           303#
$DDW1   001320           304#
$DDW10  001342           313#
$DDW11  001344           314#
```

```
$DDW12  001346            315#
$DDW13  001350            316#
$DDW14  001352            317#
$DDW15  001354            318#
$DDW2   001322            305#
$DDW3   001324            306#
$DDW4   001326            307#
$DDW5   001330            308#
$DDW6   001332            309#
$DDW7   001334            310#
$DDW8   001336            311#
$DDW9   001340            312#
$DEVCT  001242            264#
$DEVM   001310            300#
$DHSEL  030204           6497*   6672    6718    6748    7193#
$DOAGN  020722           5481    5490    5496#
$DTBL   022210           5770    5805#
$ENDAD  020712             56    5492*   5614
$ENDCT  020660            687    5483#
$ENDMG  020731           5485    5500#
$ENULL  020726           5488    5499#
$ENV    001252            269#    717    5596    5633    5901    5925
$ENVM   001253            270#    710    5835    5840    5903
$EOP    020624           5454*   5460    5473#
$EOPCT  020652            687*   5480#   5484
$ERFLG  001103            204#   5508    5539    5541    5547*   5569    5584*   5619
$ERMAX  001115            210#    690*   5541    5564*   5569
$ERROR  021236            681    5582#
$ERRPC  001116            211#   5588*   5589*   5590    5619    5633    7253    7276    7506    7552    7590    7691
$ERRTB  001356            337#   5641
$ERRTY  021422           5593    5626#
$ERTTL  001112            208#   5587*   5619
$ESCAP  001224            250#    689*   5563*   5610    5612    5619
$ETABL  001252            268#
$ETEND  001356             81     321#
$FATAL  001234            261#   5929*
$FFLG   022756           5892*   5895*   5923    5932*   5940#
$FILLC  001156            229#   5858    5889
$FILLS  001155            228#   5889
$GDADR  001120            212#
$GDDAT  001124            214#
$GET42  020702           5489#
$GTSWR  023030           5968#   6203
$HD     =  000001          25      26
$HIBTS  000214             76#
$HIOCT  024020           6154*   6165#
$ICNT   001104            205#   5554*   5555    5557*   5568
$ILLUP  024250           6213    6229    6248#
$INTAG  001135            219#   5996    6113
$ITEMB  001114            209#   5590*   5598    5619    5630
$LF     001230            253#   5619    5889    6097    6107    6166
$LFLG   022755           5933*   5939#
$LNSEL  030136           4407    4689    4804    7172#
$LPADR  001106            206#    691*    777*   5545*   5561*   5566    5568
$LPERR  001110            207#    692*    834*    908*    967*   1051*   1115*   1214*   1298*   1384*   1461*   1539*   1617*
                         1784*   1926*   1995*   2005*   2067*   2077*   2161*   2249*   2333*   2417*   2503*   2586*   2680*
```

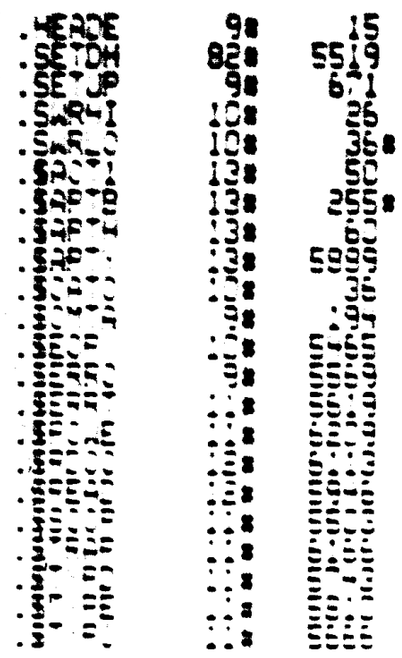
```
                    2760*  2840*  2918*  3047*  3194*  3322*  3439*  3511*  3568*  3676*  3769*  3884*  4003*
                    4122*  4267*  4349*  4529*  4638*  4764*  4886*  5049*  5145*  5545   5562*  5568   5609
                    6773*  6786*
$MADR1  001264       286*
$MADR2  001270       290*
$MADR3  001274       293*
$MADR4  001300       296*
$MAIL   001232        77     81    259*   709    717   5560   5596   5833
$MAMS1  001262       280*
$MAMS2  001266       288*
$MAMS3  001272       291*
$MAMS4  001276       294*
$MBADR  000216        77*
$MFLG   022754      5893*  5899   5934*  5938*
$MNEW   023651      5971   6111*
$MSGAD  001246       266*  5909*  5912
$MSGLG  001250       267*  5914*
$MSGTY  001232       260*  5907   5915*  5927   5931*
$MSWR   023640      5968   6109*
$MTYP1  001263       281*
$MTYP2  001267       289*
$MTYP3  001273       292*
$MTYP4  001277       295*
$MXCNT  021234      5558   5568*
$NULL   001154       227*  5860   5889
$NWTST= 000001       778*   839*   911*   998*  1054*  1161*  1244*  1328*  1417*  1495*  1573*  1653*  1794*
                    1936*  2008*  2080*  2168*  2256*  2340*  2424*  2537*  2626*  2716*  2796*  2876*  2954*
                    3144*  3262*  3374*  3446*  3519*  3620*  3737*  3830*  3948*  4055*  4232*  4315*  4463*
                    4605*  4732*  4827*  5000*  5094*  5196*  5278*  5336*  5394*
$OCNT   022000      5698*  5727*  5740*
$OMODE  022002      5693*  5697*  5702   5705*  5716*  5742*
$OVER   021220      5522   5538   5546   5556   5565*
$PASS   001240       263*   709*  5477*  5478*  5486   5499   5552   5569
$PASTM  000222        79*
$POWER  024256      6244   6251*
$PWRAD  024244      6246*
$PWRDN  024104       685   6213*  6241
$PWRMG  024240      6244*
$PWRUP  024156      6223   6229*
$QUES   001226       251*  5619   5889   6014   6090   6107   6163   6166
$RDCHR  023242      6027*  6206
$RDDEC= ****** U    6209
$RDLIN  023362      6055*  6207
$RDOCT  023662      6127*  6208
$RDSZ = 000010      6048*
$REGAD  001160       231*
$REG0   001162       233*  6274*  6286*  6298*  6309*  7253   7276   7506   7691
$REG1   001164       234*  6275*  6287*  6299*  6310*  7253   7276   7506   7691
$REG2   001166       235*  6276*  6288*  6300*  6311*  7253   7276   7506   7691
$REG3   001170       236*  6289*  6312*  7276   7691
$REG4   001172       237*  6290*  6313*  7276   7691
$REG5   001174       238*
$REG6   001176       239*  6277*  6278*  6291*  6292*  6767*  6780*  7253   7276   7506
$REG7   001200       240*
$RTNAD  020724      5498*
$R2R  = ****** U    6209
```

```
$SAVRE= ****** U        6209
$SAVR6  024254          6222*   6230    6231*   6232*   6250*
$SCOPE  020746           679    5517*   6571
$SETUP= 000137           671*    678     679     681     683     685     687     688     689     691     714    5475    5518
                        5583    5606    5614    5951    6113
$STUP = 177777           671*
$SVLAD  021164          5530    5559*
$SVPC = 000214            54*     59
$SWR  = 165400             8*     25      30      31      32      33      34      35     249     250     251     688     689
                         691     692     782     843     915    1002    1058    1165    1248    1332    1421    1499    1577
                        1657    1798    1940    2012    2084    2172    2260    2344    2428    2541    2630    2720    2900
                        2880    2959    3148    3266    3378    3450    3523    3624    3741    3834    3952    4059    4236
                        4319    4467    4609    4736    4831    5004    5098    5200    5282    5340    5398    5470    5476
                        5491    5497    5499    5509    5510    5511    5512    5513    5521    5533    5535    5536    5539
                        5540    5541    5548    5549    5550    5562    5565    5568    5575    5576    5577    5578    5587
                        5591    5603    5607    5619    6247
$SWREG  001254           271*    712
$SWRMK= 000000            35      36    5513    5514    5537
$TESTN  001236           262*   5560*
$TIMES  001222           249*    688*   5476*   5548*   5555    5558*   5568
$TKB    001146           224*   5949    5960    5977    6031    6037
$TKS    001144           223*   5949    5958    5974    5998*   6029    6035
$TMPO   001202           241*   3234*   3235*   3250*   3251*   3696*   3697*   3698*   3699*   3700*   3789*   3790*   3791*
                        3792*   3793*   4363*   4675*   4791*   5226*   5227*   5229    6306*   6307*   6766*   6779*   6868*
                        7253    7276    7506    7552    7590    7691
$TMP1   001204           242*   2129*   2144    2147    2217*   2232    2235    2302*   2386*   2492*   2508    3236*   3252*
                        4919*   4930    4951*   4960    4968*   4979    5175*   5181*   7691
$TMP2   001206           243*   2146*   2148    2151    2159    2164*   2165    2234*   2236    2239    2247    2252*   2253
                        2318*   2320    2323    2331    2336*   2337    2402*   2404    2407    2415    2420*   2421    2493*
                        2513    4372*   4373*   4662*   4663*   4777*   4778*   7552
$TMP3   001210           244*   2135*   2136    2138*   2139    2223*   2224    2226*   2227    2307*   2308    2310*   2311
                        2391*   2392    2394*   2395    2496*   2498    2501*   2502    4364*   4445*   4459*   4530*   4599*
                        7590
$TMP4   001212           245*   2512*   2515    2520    2528    2533*   2534    4384*   4404*   4405    4403*   4410*   4421
                        4784*   4801*   4802    6402*   6416*
$TMP5   001214           246*   4359*   4696*   4702    6409
$TMP6   001216           247*   4358*   4551*   4577*   4799*   4816    4823    6408    6419    6920*   6934*
$TMP7   001220           248*   1754*   1793    1896*   1935    3224*   3225    3228    3236    3244    3252    3258*   3259
                        3419*   3421    3442*   3443    3482*   3485    3495    3514*   3515    3571*   3573    3605    3609
                        3612    3617*   3618    3679*   3685    3696    3720    3732*   3733    3772*   3778    3789    3912
                        3824*   3825    3914*   3915*   4008*   4012    4041    4351*   4353*   4360*   4424    4550*   4593*
                        4646*   4708    4712    4713    6378*   6379    6382*   6383    6391*   6392    6394*   6395
$TN     = 000061           25*    778     782*    839     843*    901     911     915*    975     998    1002*   1040    1054
                        1058*   1121    1161    1165*   1221    1244    1248*   1305    1328    1332*   1391    1417    1421*
                        1468    1495    1499*   1546    1573    1577*   1624    1653    1657*   1794    1798*   1936    1940*
                        1981    2006    2012*   2053    2080    2084*   2130    2168    2172*   2219    2256    2260*   2304
                        2340    2344*   2388    2424    2428*   2494    2537    2541*   2626    2630*   2716    2720*   2796
                        2800*   2876    2880*   2954    2958*   3138    3144    3148*   3196    3262    3266*   3374    3378*
                        3446    3450*   3519    3523*   3620    3624*   3678    3737    3741*   3771    3830    3934*   3996
                        3948    3952*   4005    4055    4059*   4232    4236*   4269    4315    4319*   4380    4463    4467*
                        4600    4605    4609*   4651    4732    4736*   4793    4819    4827    4831*   5000    5004*   5051
                        5094    5098*   5147    5196    5200*   5238    5278    5282*   5298    5336    5340*   5356    5394
                        5398*
$TPB    001152           226*   5878*   5889
$TPFLG  001157           230*   5827    5889
$TPS    001150           225*   5876    5889
```

```
$TRAP   024022          683    6174*
$TRAP2  024044         6195*   6196
$TRP  = 000013         6189*   6198*   6199*   6200*   6201*   6202*   6203    6204*   6205    6206*   6207*   6208*   6209*
$TRPAD  024056         6179    6196*
$TSTM   000220           79*
$TSTNM  001102          203*   4372    4662    4777    5463*   5475*   5508    5537    5559*   5560    5565    5569    5596
                       5619    6273    6285    6308    6770    6783
$TTYIN  023616         6057    6058    6070    6088    6102    6106*
$TYPBN= ****** U       6202
$TYPDS  022004         5755*   6201
$TYPE   022230         5827*   5920    6189    6197
$TYPEC  022442         5857    5864    5871    5876*   5877    6000
$TYPEX  022510         5882    5884    5887*
$TYPOC  021602         5696*   6198
$TYPON  021616         5695    5698*   6200
$TYPOS  021556         5691*   6199
$UNIT   001244          265*
$UNITM  000224           80*
$USWR   001256          272*
$VECT1  001302          297*
$VECT2  001304          298*
$XTSTR  020770         5524*
$$GET4= 000000         5491*
$OFILL  022001         5692*   5696*   5706    5741*
$40CAT= ****** U       5521    5593
.     = 037512           38*     42*     54      55*     57*     59*     65      66*     68*     70*    200*    254     676
                        691     692     777    5269    5327    5385    5443    5499    5503    5568    5569    5619    5665*
                       5809*   5889    5941*   5949    6106*   6107    6113    6166    6225    6249    6891*   6995*   7275*
                       7589*   7916*   7920*   7926*
.$ASTA= ****** U       5893    5896
.$X   = 000214           65*     70
```

```
COMMEN    194#
ENDCOM    194#
ERROR      88#    835    909    987    996   1052   1135   1147   1159   1233   1242   1317   1326   1404   1415
          1482   1493   1560   1571   1639   1651   1785   1927   1996   2006   2068   2078   2162   2250   2334
          2418   2531   2613   2620   2703   2712   2783   2792   2863   2872   2941   2950   3083   3093   3104
          3116   3127   3221   3240   3256   3340   3352   3364   3440   3512   3590   3601   3710   3729   3802
          3821   3911   3931   3945   4034   4049   4155   4178   4199   4213   4227   4239   4313   4339   4396
          4426   4447   4461   4574   4590   4679   4704   4719   4792   4818   4931   4961   4980   4996   5077
          5092   5172   5193   6774   6787
ESCAPE    194#
GETPRI    194#
GETSWR     10#    194#    714
MHDRSO      1#   3834
MHDR1       1#    782
MHDR10      1#   1332
MHDR11      1#   1421
MHDR12      1#   1499
MHDR13      1#   1577
MHDR14      1#   1657
MHDR15      1#   1798
MHDR16      1#   1940
MHDR17      1#   2012
MHDR2       1#    843
MHDR20      1#   2094
MHDR21      1#   2172
MHDR22      1#   2260
MHDR23      1#   2344
MHDR24      1#   2428
MHDR25      1#   2541
MHDR26      1#   2630
MHDR27      1#   2720
MHDR3       1#    915
MHDR30      1#   2800
MHDR31      1#   2880
MHDR32      1#   2958
MHDR33      1#   3148
MHDR34      1#   3266
MHDR35      1#   3378
MHDR36      1#   3450
MHDR37      1#   3523
MHDR4       1#   1002
MHDR40      1#   3624
MHDR41      1#   3741
MHDR42      1#   3952
MHDR43      1#   4059
MHDR44      1#   4236
MHDR45      1#   4319
MHDR46      1#   4467
MHDR47      1#   4609
MHDR5       1#   1058
MHDR50      1#   4736
MHDR51      1#   4831
MHDR52      1#   5004
MHDR53      1#   5098
MHDR54      1#   5200
MHDR6       1#   1165
```

```
MHDR60      1*   5292
MHDR61      1*   5340
MHDR62      1*   5398
MHDR7       1*   1248
MULT      194*
NEWTST    194*    778    839    911    998   1054   1161   1244   1328   1417   1495   1573   1653   1794   1936
         2009   2080   2168   2256   2340   2424   2537   2626   2716   2796   2876   2954   3144   3262   3374
         3446   3519   3620   3737   3830   3948   4055   4232   4315   4463   4605   4732   4827   5000   5094
         5196   5278   5336   5394
POP       194*  5796   5935   5936   6155   6234   6235   6370
PUSH      194*  5755   5896   5898   5919   6129   6215   6221   6353
REPORT    194*
SCOPE      89*   781    842    914   1001   1057   1164   1247   1331   1420   1498   1576   1656   1797   1939
         2011   2083   2171   2259   2343   2427   2540   2629   2719   2799   2879   2957   3147   3265   3377
         3449   3522   3623   3740   3833   3951   4058   4235   4318   4466   4608   4735   4830   5003   5097
         5199   5281   5339   5397   5453   5474
SETPRI    194*
SETTRA   6189*  6198   6199   6200   6201   6203   6205   6206   6207   6208
SETUP      10*   194*   671
SKIP      194*   901    905    974   1040   1121   1221   1305   1391   1468   1546   1624   1757   1769   1899
         1911   1981   1992   2053   2064   2130   2218   2304   2388   2494   3138   3196   3678   3771   3886
         4005   4124   4204   4269   4355   4380   4543   4595   4600   4650   4793   4819   4898   5051   5147
         5238   5298   5356
SLASH     194*
SPACE     194*
STARS      52     62     64     71    194*   196    254    257    778    780    839    841    911    913    998
         1000   1054   1056   1161   1163   1244   1246   1328   1330   1417   1419   1495   1497   1573   1575
         1653   1655   1794   1796   1936   1938   2008   2010   2080   2082   2168   2170   2256   2258   2340
         2342   2424   2426   2537   2539   2626   2628   2716   2718   2796   2798   2876   2878   2954   2956
         3144   3146   3262   3264   3374   3376   3446   3448   3519   3521   3620   3622   3737   3739   3830
         3832   3948   3950   4055   4057   4232   4234   4315   4317   4463   4465   4605   4607   4732   4734
         4827   4829   5000   5002   5094   5096   5196   5198   5278   5280   5336   5338   5394   5396   5467
         5505   5571   5621   5668   5745   5812   5891   5948   5951   6019   6048   6115   6168   6211   6227
         6254   6256   6871   6873   7231   7233
SWRSU     194*   693*
TRMTRP   6189*
TYPBIN    194*
TYPDEC    194*  5486
TYPNAM    194*
TYPNUM    194*
TYPOCS    194*
TYPOCT    194*  5633   5657   5969
TYPTXT    194*
$$CMRE    194*   233    234    235    236    237    238    239    240
$$CMTM    194*   241    242    243    244    245    246    247    248
$$ESCA    194*
$$NEWT    194*   778    839    911    998   1054   1161   1244   1328   1417   1495   1573   1653   1794   1936
         2008   2080   2168   2256   2340   2424   2537   2626   2716   2796   2876   2954   3144   3262   3374
         3446   3519   3620   3737   3830   3948   4055   4232   4315   4463   4605   4732   4827   5000   5094
         5196   5278   5336   5394
$$SET    6189*  6198   6199   6200   6201   6203   6205   6206   6207   6208
$$SETM    709*
$$SKIP    194*   901    975   1040   1121   1221   1305   1391   1468   1546   1624   1981   2053   2130   2218
         2304   2388   2494   3138   3196   3678   3771   3886   4005   4269   4380   4600   4651   4793   4819
         5051   5147   5238   5298   5356
.EQUAT      10*    84
```



ADD	759	760	761	821	826	893	894	895	1216	1300	1386	1463	1541	1613	1753
	1765	1787	1895	1907	1929	1979	2051	2127	2215	2300	2384	2490	3049	3100	3113
	3124	3132	3134	3238	3254	3350	3362	3371	3418	3481	3570	3732	3824	3939	4122
	4198	4202	4216	4309	4417	4418	4585	4698	4773	4812	4813	4921	4953	4970	4931
	5071	5188	5228	5231	5456	5457	5458	5641	5694	5704	5775	5846	5906	5918	5930
	5985	5994	6150	6278	6292	6337	6365	6368	6414	6415	6463	6478	6712	6726	6742
	6756	6797	6800	6814	6832	6833									
ASL	764	973	1039	1120	1220	1304	1390	1467	1545	1623	2597	3790	3791	3813	3814
	4403	4696	4800	5227	5459	5638	5639	5640	6008	6009	6010	6143	6145	6147	6179
	6333														
ASLB	[illegible]														
ASR	3698	3699	3722	3723	4410	5913	6361	6362	6363	6490	6721	6751			
ASRB	[illegible]														
BCC	[illegible]														
BCS	[illegible]														
BEQ	711	719	738	746	765	828	901	905	975	984	993	1040	1048	1121	1132
	[illegible]	1156	1221	1230	1239	1305	1314	1323	1391	1401	1412	1468	1479	1490	1546
	[illegible]	1568	1624	1626	1648	1778	1920	1992	2002	2064	2074	2130	2152	2155	2166
	2218	2240	2243	2254	2324	2327	2338	2408	2411	2422	2494	2521	2524	2535	2598
	[illegible]	3109	3120	3232	3248	3347	3359	3436	3508	3594	3598	3619	3920	3935	4043
	4172	4190	4204	4218	4304	4355	4393	4406	4408	4412	4440	4454	4538	4541	4580
	4600	4645	4648	4689	4690	4693	4709	4803	4805	4808	4993	4996	4913	4945	4965
	5063	5184	5257	5272	5315	5333	5373	5398	5431	5446	5460	5462	5490	5536	5538
	[illegible]	5544	5553	5585	5608	5611	5643	5648	5661	5721	5836	5849	5884	5900	5904
	5924	5926	5965	5992	6007	6076	6138	6334	6336	6449	6462	6477	6517	6549	6585
	6591	6633	6635	6637	6652	6657	6663	6668	6676	6681	6687	6692	6697		
BGE	5556	6705	6735												
BGT	4594	5491	5728	5789	6004	6045	6140								
BHI	5542														
BHIS	[illegible]														
BIC	991	1138	1139	1141	1153	1237	1321	1399	1408	1410	1474	1476	1550	1554	1628
	1632	1634	1646	2519	3078	3356	3427	3433	3491	3504	3580	3700	3702	3724	3733
	3816	3906	3926	4027	4149	4165	4373	4402	4432	4567	4596	4663	4664	4684	4724
	4727	4778	4798	4924	4925	4923	4972	5096	5164	5241	5251	5300	5309	5325	5359
	5367	5383	5416	5425	5441	5478	5718	5961	5978	6005	6032	6038	6046	6149	6364
	6367	6457	6529	6533	6539	6558	6567								
BICB	2497	2499	2514	6340	6385	6397									
BIS	903	1125	1127	1129	1151	1311	1397	1495	1497	1563	1565	1642	1644	2605	2692
	2693	2772	2773	2852	2853	2930	2931	3065	3066	3197	3329	3423	3488	3575	4014
	4134	5166	5179	5261	5319	5377	5435	5723	5724	5783	5794	6012	6444	6446	6447
	6451	6520	6521	6528	6537	6550	6551	6557	6566						
BISB	1762	1774	1904	1916	1983	2055	2498	2507	2508	2515	3058	3062	3200	3215	3603
	3682	3705	3775	3797	3904	3924	4010	4015	4029	4039	4041	4129	4150	4157	4168
	4272	4274	4293	4569	4657	5054	5178	5630							
BIT	734	739	762	971	1037	1118	1218	1302	1398	1465	1543	1621	2595	3619	3897
	3919	4392	4407	4540	4647	4689	4804	4895	5222	5461	5521	5535	5543	5550	5591
	5607	6235	6448	6453	6590	6651	6662	6675	6696	5696	6710	6714	6740	6744	
BITB	710	5835	5840	5872	5903										
BLO	2716	3808	6071												
BLOS	6059	6514	6581												
BLT	5729	5772	5788	5863	6002	6043	6142	6708	6738						
BMI	3087	3207	3583	3691	3784	4020	4140	4285	4386	4558	4595	4670	4786	5159	5779
	6674	6685													
BNE	676	700	716	720	730	735	740	748	763	972	1038	1119	1219	1303	1389
	1466	1544	1622	1759	1771	1901	1913	2140	2228	2312	2396	2503	2596	3229	3245
	3260	3334	3426	3430	3444	3490	3493	3499	3501	3516	3579	3585	3613	3734	3826
	3898	3918	4053	4160	4184	4431	4723	4726	4729	4776	4822	4826	4986	5192	5223
	5233	5253	5263	5269	5311	5321	5327	5369	5379	5385	5427	5437	5443	5522	5551
	5592	5597	5615	5631	5653	5719	5777	5834	5841	5843	5851	5859	5873	5880	5902

```
                5908    5911    5926    5957    5963    5983    5990    5997    6034    6040    6063    6065    6081    6085    6095
                6233    6266    6321    6384    6396    6417    6421    6434    6456    6486    6496    6525    6555    6620    6622
                6711    6715    6723    6728    6741    6745    6753    6758    6811    6813    6835
SP              5089    5604    5714    5763    5793    5828    5877    5959    5975    6030    6036
SR                 722     742     766     830     837     910     976     997    1041    1053    1122    1136    1148    1160
                1203    1243    1306    1321    1392    1416    1469    1483    1494    1547    1561    1578    1605    1640    1650
                1756    1769    1758    1791    1899    1908    1911    1930    1933    1981    2002    2053    2079    2130
                2255    2304    2339    2389    2423    2506    2536    2599    2608    2704    2784    2864    2934    2980
                3054    3084    3094    3105    3117    3128    3139    3196    3209    3222    3261    3341    3353    3531
                3678    3688    3693    3711    3735    3771    3786    3803    3827    3896    3900    3912    3932    3947
                4033    4056    4124    4125    4142    4156    4269    4287    4300    4314    4390    4390    4397
                4456    4535    4544    4560    4575    4591    4597    4642    4651    4672    4680    4788    4793
                4898    4899    4915    4932    4947    4962    5051    5065    5078    5093    5147    5160    5173
                5298    5356    5414    5524    5530    5533    5546    5549    5602    5636    5663    5695    5710
                5751    5830    5856    5866    5875    5882    5894    5916    5986    6013    6035    6041    6074
                6089    6091    6151    6164    6225    6249    6331    6338    6464    6479    6530    6559    6638    6659
                6706    6709    6724    6729    6736    6739    6754    6759
                4536    4643    4891
                 670     674     688     689     709     744     997     979     989    1043    1044    1126    1225    1235
                1309    1396    1406    1472    1551    1625    1760    1772    1790    1902    1914    1932    1998    2070
                2153    2225    2241    2316    2319    2400    2403    2505    2522    2699    2700    2709    2710    2779
                2789    2790    2859    2860    2869    2870    2937    2938    2947    2948    3070    3101    3123    3135
                3224    3230    3234    3242    3250    3368    3372    3422    3571    3684    3689    3777    3782    3890
                3923    4007    4018    4138    4194    4208    4283    4351    4384    4438    4452    4530    4533    4551
                4564    4583    4601    4640    4668    4772    4783    4784    4898    4911    4943    4955    5061    5073
                5247    5248    5264    5267    5296    5299    5305    5306    5354    5357    5363    5364    5412
                5422    5475    5476    5519    5548    5563    5629    5708    5766    5769    5972    5973    6056
                6136    6231    6262    6306    6339    6378    6380    6381    6391    6404    6429    6431    6491
                6523    6549    6552    6798    6801    6821    6824    6837    6860
CLRB            2146    2223    2234    2307    2318    2391    2402    2496    2512    3608    5463    5547    5795    5855
                5932    5933    5934    6086    6096    6160    6330    6403    6410    6422    6829
CR               699     719     747     827     833     900     993    1131    1143    1155    1229    1313    1400    1411
                1489    1556    1567    1635    1647    1777    1919    1991    2063    2154    2242    2326    2413    2523
                3231    3247    3259    3346    3358    3435    3443    3507    3515    3597    3715    3733    3807    3835
                4052    4159    4171    4183    4217    4303    4354    4405    4411    4579    4692    4708    4802    4907
                4985    5183    5252    5256    5310    5314    5329    5368    5372    5387    5426    5430    5445    5531
                5614    5787    5956    5962    5982    5989    6001    6003    6033    6039    6042    6044    6058    6070
                6383    6395    6420    6433    6461    6476    6511    6512    6535    6564    6579    6580    6634    6636
                6727    6734    6737    6757
CMPB            2139    2151    2165    2227    2239    2253    2311    2323    2337    2395    2407    2421    2502    2520
                3228    3244    3934    4687    4912    4964    5537    5541    5596    5933    5949    5950    5959    5979
                5901    5964    5996    6062    6080    6084    6094    6139    6141
CMP              733    4599
CMPB             638
DEC             3333    3425    3429    3489    3492    3498    3500    3578    3581    3917    4353    4593    4775    5181    5262
                5320    5378    5436    5479    5637    6069    6416    6494    6812    6834
DECB            5727    5862    5865
EMT              488
HALT             488    5260    5275    5318    5333    5376    5391    5434    5449    5605    5616    5829    6224    6249    6439
                6700
INC              758    3246    3258    3428    3442    3514    3575    3606    4051    4181    4360    4404    4430    4577    4774
                4801    4806    5455    5477    5554    5587    5722    5730    5773    5931    6011    6232    6264    6382    6394
                6413    6432    6493    6525    6554    6810    6830    6831
INCB            2138    2164    2226    2252    2310    2336    2394    2420    2501    2533    4429    4722    5268    5326    5384
                5442    5559    5564    5885    6332
INT              89
```

JMP	46	47	48	749	750	4126	4229	4375	4448	4462	4603	4705	4720	4730	4731
	4900	4981	4997	4997	5224	5236	5464	5497	6645	6648	6701	6777	6790		
JSR	736	743	770	775	832	907	996	998	1050	1134	1146	1158	1232	1241	1316
	1325	1403	1414	1481	1492	1559	1570	1639	1650	1756	1768	1780	1791	1898	1910
	1922	1923	1980	1984	1994	2004	2052	2056	2066	2076	2131	2157	2158	2219	2245
	2246	2303	2329	2330	2387	2413	2414	2505	2526	2527	2602	2606	2610	2612	2617
	2619	2625	2690	2694	2697	2702	2706	2711	2715	2770	2774	2777	2782	2786	2791
	2795	2850	2854	2857	2862	2866	2871	2875	2928	2932	2935	2940	2944	2949	2953
	3052	3059	3060	3067	3072	3076	3079	3080	3089	3091	3092	3099	3102	3103	3111
	3114	3115	3122	3125	3126	3136	3140	3195	3198	3199	3208	3212	3217	3218	3239
	3255	3326	3331	3336	3339	3349	3351	3361	3363	3367	3438	3510	3587	3588	3589
	3595	3599	3600	3677	3692	3706	3707	3719	3725	3726	3770	3785	3798	3799	3811
	3817	3818	3895	3899	3902	3907	3908	3922	3927	3928	3937	3941	3942	4004	4021
	4025	4030	4031	4045	4046	4123	4141	4145	4151	4152	4174	4175	4192	4195	4196
	4206	4209	4210	4220	4223	4224	4268	4286	4290	4295	4296	4306	4309	4310	4362
	4365	4374	4378	4379	4414	4419	4420	4423	4441	4455	4542	4545	4559	4563	4570
	4571	4582	4586	4597	4649	4655	4671	4676	4695	4700	4701	4711	4715	4716	4787
	4810	4814	4815	4897	4902	4914	4918	4926	4927	4946	4950	4956	4957	4967	4975
	4976	4989	4992	4993	5050	5053	5064	5069	5073	5074	5083	5088	5089	5146	5153
	5162	5168	5169	5186	5189	5190	5237	5297	5355	5413	5492	5593	5599	5838	5857
	5864	5871	5920	6000	6272	6284	6658	6669	6772	6776	6785	6789			
MOV	673	677	679	680	681	682	683	684	685	686	687	691	692	695	696
	697	698	703	705	706	707	712	725	726	727	728	741	752	753	754
	755	756	767	768	769	777	819	820	822	823	834	838	888	889	890
	891	892	899	902	904	908	967	968	969	978	981	982	992	1035	1036
	1046	1047	1051	1115	1116	1117	1124	1130	1142	1150	1152	1154	1214	1215	1217
	1224	1227	1228	1238	1298	1299	1301	1308	1312	1322	1384	1355	1357	1394	1396
	1358	1407	1409	1461	1462	1464	1471	1475	1477	1499	1539	1540	1542	1549	1552
	1555	1566	1617	1618	1620	1627	1630	1631	1633	1643	1645	1752	1754	1755	1764
	1776	1784	1793	1894	1896	1897	1906	1918	1926	1935	1979	1982	1989	1990	1995
	2000	2001	2005	2050	2054	2061	2062	2067	2072	2073	2077	2126	2128	2129	2144
	2147	2150	2161	2214	2216	2217	2232	2235	2238	2249	2299	2301	2302	2309	2322
	2325	2333	2383	2385	2386	2393	2406	2409	2417	2489	2491	2492	2493	2509	2510
	2513	2516	2517	2518	2530	2586	2587	2588	2589	2592	2594	2601	2603	2604	2611
	2614	2618	2621	2624	2680	2681	2682	2683	2686	2688	2689	2691	2698	2701	2707
	2708	2714	2760	2761	2762	2763	2766	2768	2769	2771	2779	2781	2787	2788	2794
	2840	2841	2842	2843	2846	2848	2849	2851	2858	2861	2867	2868	2914	2918	2919
	2920	2921	2924	2926	2927	2929	2936	2939	2945	2946	2952	3047	3048	3050	3055
	3056	3057	3061	3063	3064	3069	3077	3090	3096	3107	3112	3119	3130	3131	3133
	3137	3194	3201	3202	3204	3213	3214	3216	3226	3227	3235	3237	3243	3252	3253
	3322	3323	3324	3327	3328	3330	3332	3337	3338	3343	3344	3345	3355	3357	3366
	3369	3370	3417	3419	3420	3421	3424	3432	3439	3480	3482	3483	3494	3496	3497
	3495	3496	3503	3505	3511	3568	3569	3572	3573	3574	3577	3604	3605	3676	3679
	3680	3681	3683	3685	3686	3688	3696	3701	3703	3704	3713	3714	3720	3731	3739
	3772	3773	3774	3776	3778	3779	3791	3793	3794	3795	3796	3805	3806	3812	3823
	3884	3887	3888	3891	3892	3893	3895	3903	3905	3914	3915	3916	3925	3939	3940
	4003	4006	4008	4009	4011	4012	4013	4017	4026	4028	4037	4038	4122	4127	4129
	4130	4131	4132	4133	4135	4137	4146	4147	4149	4158	4162	4163	4170	4186	4190
	4193	4201	4207	4215	4221	4222	4267	4270	4271	4275	4276	4277	4278	4279	4280
	4282	4291	4292	4294	4302	4307	4349	4350	4356	4357	4358	4359	4361	4363	4364
	4366	4367	4369	4370	4371	4377	4388	4395	4399	4400	4409	4415	4416	4437	4445
	4446	4451	4459	4460	4529	4531	4532	4539	4547	4548	4550	4552	4553	4555	4565
	4566	4568	4578	4584	4602	4638	4639	4646	4653	4654	4658	4660	4661	4665	4667
	4669	4675	4682	4685	4696	4697	4699	4712	4713	4714	4764	4765	4766	4767	4768
	4769	4770	4771	4779	4780	4782	4791	4795	4796	4799	4811	4823	4886	4887	4894
	4901	4905	4906	4907	4908	4910	4919	4920	4922	4924	4930	4934	4936	4937	4938

CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
                4939  4940  4942  4951  4952  4954  4960  4968  4969  4971  4973  4979  4984  4990  5049
                5052  5055  5056  5057  5058  5060  5069  5070  5084  5085  5145  5148  5150  5151  5152
                5153  5155  5163  5167  5175  5180  5187  5225  5226  5229  5230  5243  5244  5245  5246
                5249  5254  5265  5266  5270  5302  5303  5304  5307  5312  5322  5323  5324  5328  5360
                5361  5362  5365  5370  5380  5381  5382  5386  5418  5419  5420  5423  5428  5438  5439
                5440  5444  5454  5482  5486  5489  5520  5526  5527  5529  5532  5545  5557  5558  5561
                5562  5565  5566  5586  5588  5609  5612  5628  5633  5642  5647  5652  5654  5658  5691
                5699  5700  5701  5707  5714  5732  5733  5734  5735  5736  5756  5757  5758  5759  5760
                5766  5762  5767  5770  5790  5796  5797  5798  5799  5800  5802  5803  5831  5832  5837
                5845  5660  5897  5898  5905  5909  5914  5915  5917  5919  5929  5935  5936  5969  5993
                5998  6023  6028  6055  6057  6068  6099  6100  6101  6102  6127  6128  6129  6130  6131
                6133  6134  6153  6154  6155  6156  6157  6174  6175  6179  6185  6186  6213  6214  6215
                6216  6217  6218  6219  6220  6221  6222  6223  6229  6230  6234  6235  6236  6237  6238
                6239  6240  6241  6242  6245  6261  6274  6275  6276  6277  6286  6287  6288  6289  6290
                6291  6298  6299  6300  6309  6310  6311  6312  6313  6329  6354  6355  6356  6357  6358
                6360  6370  6371  6372  6393  6402  6405  6406  6407  6408  6409  6418  6419  6428  6430
                6435  6436  6437  6438  6440  6458  6466  6471  6472  6474  6481  6484  6497  6501  6502
                6503  6504  6505  6506  6508  6510  6532  6534  6536  6543  6544  6545  6561  6563  6565
                6566  6572  6573  6574  6576  6578  6583  6589  6594  6595  6596  6597  6599  6603  6608
                6613  6632  6639  6644  6647  6650  6656  6661  6667  6672  6680  6682  6691  6693  6695
                6713  6716  6719  6719  6725  6743  6746  6748  6749  6755  6766  6767  6768  6769  6771
                6773  6779  6780  6781  6782  6784  6786  6795  6796  6799  6820  6822  6823  6825  6826
                6828  5836  6844  6845  6852  6853  6961  6962  6864  6865  6867  6868
MCVB             590   723   980   990  1045  1128  1140  1226  1236  1310  1320  1473  1486  1553  1564
                1763  1775  1905  1917  1988  1999  2060  2071  2133  2136  2142  2143  2148  2149  2221
                2228  2230  2231  2236  2237  2305  2308  2314  2315  2320  2321  2389  2392  2398  2399
                2404  2506  2590  2593  2684  2687  2764  2767  2844  2847  2922  2925  3051  3225  3235
                3251  3325  3889  4164  4368  4372  4546  4662  4777  4903  4935  5087  5149  5165  5176
                5228  5250  5301  5308  5359  5366  5417  5424  5560  5564  5590  5598  5692  5693  5696
                5697  5698  5702  5705  5706  5725  5765  5768  5782  5795  5794  5842  5870  5878  5892
                5893  5895  5963  5977  6031  6037  6061  6066  6072  6077  6092  6137  6177  6263  6273
                6285  6307  6309  6359  6366  6369  6379  6392  6411  6770  6783  6829
MCVB             573   764  6588
MCVB            2607  2695  2775  2855  2933  3071  4231  4998  5493  5494  5495
RESET           5491  6775  6788
RCVB            3617  4537  4644  4892  5709  5711  5712  5713  5715  6144  6146  6148  6492
RTT              704  2615  2622  4433  5567  5618  5737  5804  5847  5999  6047  6103  6158  6187  6247
                6467  6482  6539  6569  6846  6854  6866
RTS             3143  3610  5656  5887  5937  6180  6267  6279  6293  6301  6314  6341  6373  6386  6398
                6423  6624  6640  6732  6762  6802  6815  6838  6847  6855  6869
SEC             3614  4534  4641  4889
SCVB             824  3485  4352  4549  4659  5589  5771  5912  6485  6562  6586
SLAB            3434  3506  3607  3697  3721  3792  3815  4040  4166  4169  4273  4401  4693  4797  4925
                4974  5177
TRAP            6189  6198  6199  6200  6201  6203  6205  6206  6207  6208  6963
TST              715   729   737   745   829   896  3086  3206  3497  3584  3593  3612  3690  4189  4203
                4439  4453  4725  4944  5062  5157  5232  5271  5528  5552  5603  5610  5660  5720  5776
                5786  5844  5852  5874  5907  5925  5927  5991  6006  6064  6075  6098  6152  6159  6176
                6442  6443  6473  6495  6516  6547  6584  6619  6621  6673  6684  6722  6752
TST2            1758  1770  1900  1912  2591  2685  2765  2845  2923  3582  3783  4019  4139  4284  4395
                4557  4728  4785  5080  5539  5778  5792  5827  5876  5899  5910  5923  5958  5974  6029
                6035  6326
.ASCII           251   252  7884  7889
.ASCIZ           253  5500  5664  6107  6108  6109  6111  6251  7237  7244  7259  7264  7282  7293  7301
                7309  7317  7327  7337  7346  7355  7367  7377  7387  7395  7401  7415  7424  7434  7445
                7451  7465  7474  7483  7492  7497  7512  7520  7527  7541  7547  7557  7566  7579  7585
```

```
                7595    7602    7611    7620    7627    7634    7646    7658    7669    7679    7697    7708    7719    7723    7737
                7745    7759    7773    7783    7793    7800    7904    7811    7818    7925    7932    7840    7846    7854    7865
                7870    7877    7883    7895    7902    7909
.BLKB           6106    7926
.BLKW           5909    6991    6995    7916    7920
.BYTE            203     204     209     210     218     219     227     228     229     230     269     270     280     281     289
                 289     291     292     294     295    5499    5600    5601    5738    5739    5740    5741    5938    5939    5940
                6104    6105    6601    6602    6605    6606    6610    6611    6615    6616    7194    7195
.CSAB           5016
.ENABL            14    5949
.ENDC           7928
.ENDC             20      33      34      35      36      47      53      57      59      63      65      72      88     180     194
                 197     201     203     231     241     249     250     251     255     258     280     288     291     294     297
                 298     299     300     301     302     303     304     305     306     307     308     309     310     311     312
                 313     314     315     316     317     318     319     323     671     677     678     681     683     685     687
                 688     689     691     693     714     719     725     779     780     781     782     840     841     842     843
                 902     906     912     913     914     915     976     999    1000    1001    1002    1041    1055    1056    1057
                1058    1122    1162    1163    1164    1165    1222    1245    1246    1247    1248    1306    1329    1330    1331
                1332    1392    1418    1419    1420    1421    1469    1496    1497    1498    1499    1547    1574    1575    1576
                1577    1625    1654    1655    1656    1657    1756    1770    1795    1796    1797    1798    1900    1912    1937
                1938    1939    1940    1982    1993    2009    2010    2011    2012    2054    2065    2081    2082    2083    2084
                2131    2169    2170    2171    2172    2219    2257    2258    2259    2260    2305    2341    2342    2343    2344
                2389    2425    2426    2427    2428    2495    2538    2539    2540    2541    2627    2628    2529    2630    2717
                2718    2719    2720    2797    2798    2799    2800    2877    2878    2879    2880    2955    2956    2957    2958
                3139    3145    3146    3147    3148    3197    3263    3264    3265    3266    3375    3376    3377    3378    3447
                3448    3449    3450    3520    3521    3522    3523    3621    3622    3623    3624    3679    3738    3739    3740
                3741    3772    3831    3832    3833    3834    3887    3949    3950    3951    3952    4006    4056    4057    4058
                4059    4125    4205    4233    4234    4235    4236    4270    4316    4317    4318    4319    4356    4381    4464
                4465    4466    4467    4545    4596    4601    4606    4607    4608    4609    4652    4733    4734    4735    4736
                4794    4820    4828    4829    4830    4831    4899    5001    5002    5003    5004    5052    5095    5096    5097
                5098    5148    5197    5198    5199    5200    5239    5279    5280    5281    5282    5299    5337    5338    5339
                5340    5357    5395    5396    5397    5398    5468    5469    5470    5472    5475    5481    5484    5485    5499
                5491    5497    5499    5500    5503    5506    5509    5514    5521    5523    5534    5537    5538    5539    5541
                5543    5550    5554    5559    5561    5565    5568    5569    5572    5575    5584    5598    5593    5594    5595
                5603    5614    5618    5619    5622    5637    5666    5669    5746    5813    5842    5892    5893    5896    5923
                5938    5949    5950    5952    5980    6016    6020    6048    6049    6057    6059    6062    6090    6107    6113
                6116    6122    6166    6169    6175    6178    6197    6198    6199    6200    6201    6202    6203    6204    6205
                6206    6207    6209    6209    6212    6221    6222    6228    6234    6235    6245    6247    6254    6255    6257
                6872    6874    7232    7234
.EQUIV            88      89      97     142     143     144     145     146     147     148     149     150     151     170     171
                 172     173     174     175     176     177     178     179
.EVEN             258    5665    5941    6253    7252    7275    7505    7551    7589    7690    7790    7913    7925
.IF                16      33      34      35      36      45      52      55      57      62      64      71      86     152     180
                 196     200     202     231     241     249     250     251     254     255     257     280     288     291     294
                 297     298     299     300     301     302     303     304     305     306     307     308     309     310     311
                 312     313     314     315     316     317     318     319     323     671     672     677     679     681     683
                 685     687     688     689     691     709     714     717     778     780     782     939     941     943     901
                 905     911     913     915     975     998    1000    1002    1040    1054    1056    1058    1121    1161    1163
                1165    1221    1244    1246    1248    1305    1328    1330    1332    1391    1417    1419    1421    1459    1495
                1497    1499    1546    1573    1575    1577    1624    1653    1655    1657    1757    1769    1794    1796    1798
                1899    1911    1936    1938    1940    1981    1992    2008    2010    2012    2053    2064    2080    2082    2084
                2130    2168    2170    2172    2218    2256    2258    2260    2304    2340    2342    2344    2399    2424    2426
                2428    2494    2537    2539    2541    2626    2628    2630    2716    2718    2720    2796    2798    2820    2976
                2878    2880    2954    2956    2958    3138    3144    3146    3148    3196    3252    3264    3266    3274    3276
                3378    3446    3448    3450    3519    3521    3523    3620    3622    3624    3678    3736    3739    3741    3771
                3830    3832    3834    3886    3948    3950    3952    4005    4055    4057    4059    4124    4204    4232    4234
```

```
        4236  4269  4315  4317  4319  4355  4380  4463  4465  4467  4544  4595  4600  4605  4607
        4609  4651  4732  4734  4736  4793  4819  4827  4823  4931  4998  5000  5002  5004  5051
        5094  5096  5098  5147  5196  5198  5200  5238  5278  5280  5282  5298  5336  5338  5340
        5356  5394  5396  5398  5467  5468  5469  5470  5471  5472  5474  5480  5483  5485  5489
        5491  5497  5499  5500  5505  5508  5513  5519  5521  5533  5535  5536  5537  5539  5540
        5541  5550  5552  5560  5562  5567  5568  5569  5571  5574  5584  5587  5591  5593  5594
        5596  5603  5607  5614  5618  5619  5621  5636  5652  5668  5745  5812  5833  5891  5893
        5896  5923  5938  5948  5950  5951  5952  5980  6019  6020  6048  6056  6058  6062  6063
        6106  6107  6113  6115  6118  6134  6168  6174  6178  6189  6198  6199  6200  6201  6202
        6203  6205  6206  6207  6208  6209  6211  6221  6222  6227  6234  6235  6243  6245  6247
        6251  6254  6256  6871  6873  7231  7233
.IFF      33    34    35    36    53    57    63    65    72    96   197   200   203   231   255
         259   677   779   780   781   782   840   841   842   843   902   905   912   913   914
         915   976   999  1000  1001  1002  1041  1055  1056  1057  1058  1122  1162  1163  1164
        1165  1222  1245  1246  1247  1248  1306  1329  1330  1331  1332  1392  1418  1419  1420
        1421  1469  1496  1497  1498  1499  1547  1574  1575  1576  1577  1625  1654  1655  1656
        1657  1757  1769  1795  1796  1797  1798  1899  1911  1937  1938  1939  1940  1992  1992
        2009  2010  2011  2012  2054  2064  2081  2082  2083  2084  2131  2169  2170  2171  2172
        2219  2257  2258  2259  2260  2305  2341  2342  2343  2344  2389  2425  2426  2427  2428
        2495  2538  2539  2540  2541  2627  2628  2629  2630  2717  2718  2719  2720  2797  2798
        2799  2800  2877  2878  2879  2880  2955  2956  2957  2958  3139  3145  3146  3147  3148
        3197  3263  3264  3265  3266  3375  3376  3377  3378  3447  3448  3449  3450  3520  3521
        3522  3523  3621  3622  3623  3624  3679  3738  3739  3740  3741  3772  3831  3832  3833
        3834  3887  3949  3950  3951  3952  4006  4056  4057  4058  4059  4124  4204  4233  4234
        4235  4236  4270  4316  4317  4318  4319  4355  4381  4464  4465  4466  4467  4544  4595
        4601  4606  4607  4608  4609  4652  4733  4734  4735  4736  4794  4820  4828  4829  4830
        4831  4838  5001  5002  5003  5004  5052  5095  5096  5097  5098  5148  5197  5198  5199
        5200  5239  5279  5280  5281  5282  5299  5337  5338  5339  5340  5357  5395  5396  5397
        5398  5468  5471  5475  5481  5484  5499  5506  5534  5537  5538  5541  5568  5572  5574
        5588  5614  5619  5622  5637  5666  5669  5746  5813  5892  5949  5952  6020  6022  6027
        6048  6049  6059  6090  6107  6116  6169  6175  6212  6228  6245  6255  6257  6872  6874
        7232  7234
.IFT     5549  5594  6022  6027  6139  6159  6166
.IFTF    5547  5593  5967  6020  6023  6135  6143  6165
.IIF       15    20    25    26    30    31    32    33    35    36    42   254   258   678   681
         687   688   689   691   692  5469  5475  5476  5487  5499  5503  5509  5510  5511  5512
        5513  5514  5518  5548  5549  5565  5568  5569  5575  5576  5577  5578  5583  5606  5614
        5619  5634  5659  5889  5949  5970  6098  6107  6113  6166  6197  6198  6199  6200  6201
        6203  6205  6206  6207  6208
.IRP     671   778   839   911   998  1054  1161  1244  1328  1417  1495  1573  1653  1794  1936
        2008  2080  2168  2256  2340  2424  2537  2626  2716  2796  2876  2954  3144  3262  3374
        3446  3519  3620  3737  3830  3948  4055  4232  4315  4463  4605  4732  4827  5000  5094
        5196  5278  5336  5394  5519  5756  5796  5897  5898  5919  5935  5936  6129  6155  6215
        6221  6234  6235  6354  6370
.LIST       1     7    35    42   194   231   233   234   235   236   237   238   239   240   241
         242   243   244   245   246   247   248   249   255   258   671   693   779   782   839
         843   911   915   998  1002  1054  1058  1161  1165  1244  1248  1328  1332  1417  1421
        1495  1499  1573  1577  1653  1657  1794  1798  1936  1940  2008  2012  2080  2084  2168
        2172  2256  2260  2340  2344  2424  2428  2537  2541  2626  2630  2716  2720  2796  2800
        2876  2880  2954  2958  3144  3148  3262  3266  3374  3378  3446  3450  3519  3523  3630
        3624  3737  3741  3830  3834  3948  3952  4055  4059  4232  4236  4315  4319  4463  4467
        4605  4609  4732  4736  4827  4831  5000  5004  5094  5098  5196  5200  5279  5282  5336
        5340  5394  5398  5475  5491  5513  5614  6048  6189  6197  6198  6199  6200  6201  6202
        6203  6204  6205  6206  6207  6208  6209
.MACRO     1    36    82   194   709  6189
.MCALL     9    10    11    12    13   194   255   693
```

```
.NLIST      1     6    35    42   194   231   233   234   235   236   237   238   239   240   241
          242   243   244   245   246   247   248   249   255   258   671   693   778   782   839
          943   911   915   998  1002  1054  1058  1161  1165  1244  1248  1328  1332  1417  1421
         1495  1499  1573  1577  1653  1657  1794  1798  1936  1940  2008  2012  2080  2084  2168
         2172  2256  2260  2340  2344  2424  2428  2537  2541  2626  2630  2716  2720  2796  2800
         2876  2880  2954  2958  3144  3148  3262  3266  3374  3378  3446  3450  3519  3523  3620
         3624  3737  3741  3830  3834  3948  3952  4055  4059  4232  4236  4315  4319  4463  4467
         4605  4609  4732  4736  4827  4831  5000  5004  5094  5098  5196  5200  5278  5282  5336
         5340  5394  5398  5475  5491  5513  5614  6048  6189  6197  6198  6199  6200  6201  6202
         6203  6204  6205  6206  6207  6208  6209
.PAGE     194   323   669   725   778   839   911   998  1054  1161  1244  1328  1417  1495  1573
         1653  1794  1936  2008  2080  2168  2256  2340  2424  2537  2626  2716  2796  2876  2954
         3144  3262  3374  3446  3519  3620  3737  3830  3948  4055  4232  4315  4463  4605  4732
         4827  5000  5094  5278  5336  5394  6254  7231
.REM      782   843   915  1002  1058  1165  1248  1332  1421  1499  1577  1657  1798  1940  2012
         2084  2172  2260  2344  2428  2541  2630  2720  2800  2880  2958  3148  3266  3378  3450
         3523  3624  3741  3834  3952  4059  4236  4319  4467  4609  4736  4831  5004  5098  5200
         5282  5340  5398
.REPT     142   233   241  7092  7113  7133  7154
.SBTTL     26    36    45    50    60    84   194   255   323   671   714   778   839   911   939
         1054  1161  1244  1328  1417  1495  1573  1653  1794  1936  2008  2080  2168  2256  2340
         2424  2537  2626  2716  2796  2876  2954  3144  3262  3374  3446  3519  3620  3737  3830
         3948  4055  4232  4315  4463  4605  4732  4827  5000  5094  5196  5278  5336  5394  5465
         5503  5569  5619  5666  5743  5810  5889  5946  6113  6166  6189  6209
.TITLE     15
.WORD      42    43    44    58    76    77    78    79    80    81   202   205   206   207   208
          211   212   213   214   215   216   217   220   221   222   231   233   234   235   236
          237   238   239   240   241   242   243   244   245   246   247   248   260   261   262
          263   264   265   266   267   271   272   273   286   290   293   296   297   298   299
          300   301   302   303   304   305   306   307   308   309   310   311   312   313   314
          315   316   317   318  5480  5483  5498  5645  5650  5742  5839  5886  5921  6162  6165
         6196  6244  6246  7092  7093  7094  7095  7096  7097  7098  7099  7100  7101  7102  7103
         7104  7105  7106  7107  7108  7113  7114  7115  7116  7117  7118  7119  7120  7121  7122
         7123  7124  7125  7126  7127  7128  7133  7134  7135  7136  7137  7138  7139  7140  7141
         7142  7143  7144  7145  7146  7147  7148  7149  7154  7155  7156  7157  7158  7159  7160
         7161  7162  7163  7164  7165  7166  7167  7168  7169  7173  7174  7175  7176  7177  7179
         7179  7180  7181  7182  7183  7184  7185  7186  7187  7188  7253  7276  7506  7552  7590
         7691
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DZDHMB=DHMMAC,DZDHMB
RUN-TIME:  92 74 12 SECONDS
RUN-TIME RATIO: 349/179=1.9
CORE USED:  42K  (83 PAGES)
```

J09

Spooler runtime 47 Seconds, 1380 KCS, 1505 disk reads, 4 disk writes, 319 pages