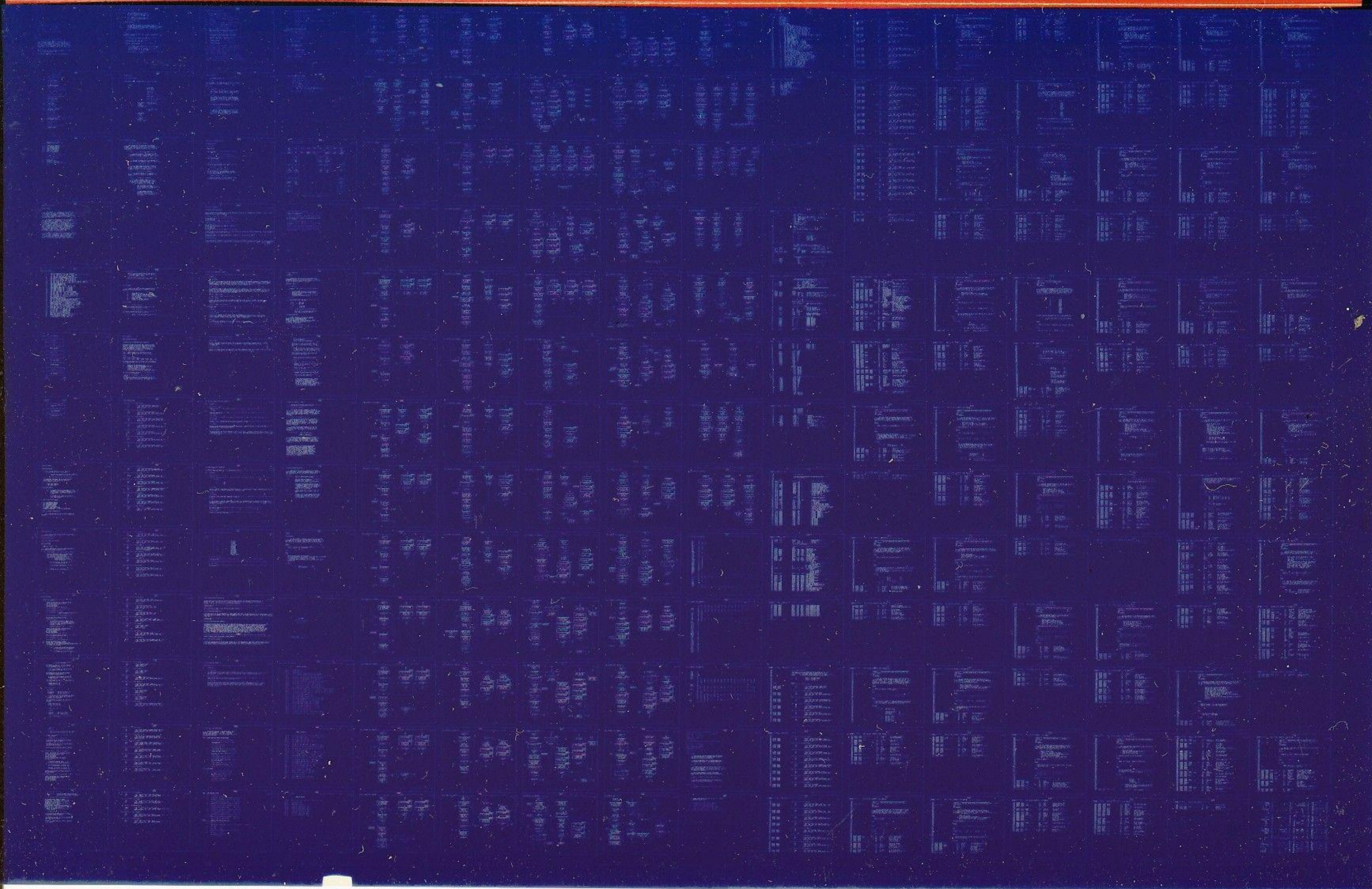


DH11

DIAGNOSTIC
MD-11-DZDHM-B

EP-DZDHM-B-DL-A NOV 1976
COPYRIGHT © 1976
FICHE 1 OF 2 MADE IN U.S.



DH11

DIAGNOSTIC
MD-11-DZDHM-B

EP-DZDHM-B-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 2 OF 2

MADE IN USA

PRODUCT CODE: MAINDEX-11-DZDMH-2
PRODUCT NAME: DH11 DIAGNOSTIC
DATE: 21 SEPTEMBER-1976
AUTHOR: E. CHAWLEY
MAINTAINED BY: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1976

DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

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

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

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

TABLE OF CONTENTS

: SEQ 0002

- 1.0 GENERAL PROGRAM DESCRIPTION
 - 1.0.1 PROGRAM PURPOSE
 - 1.1.1 LOGIC TEST SUMMARY
 - 1.1.2 MD-11-DZCHM CORE MEMORY MAP
 - 1.2 SYSTEM REQUIREMENTS
 - 1.2.1 HARDWARE REQUIREMENTS
 - 1.2.2 SOFTWARE REQUIREMENTS
 - 1.3 RELATED DOCUMENTS AND STANDARDS
 - 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
 - 1.5 FAILURE ASSUMPTIONS
- 2.0 OPERATING INSTRUCTIONS
 - 2.1 LOADING AND STARTING PROCEDURES
 - 2.1.1 LOADING PROCEDURES
 - 2.1.2 STARTING PROCEDURES
 - 2.2 SPECIAL ENVIRONMENTS
 - 2.2.1 ACT11/APT11
 - 2.2.2 "XXDP" SYSTEMS
 - 2.2.3 SWITCHLESS FEATURE
 - 2.3 PROGRAM OPTIONS
 - 2.3.1 CONSOLE SWITCH REGISTER
 - 2.3.2 CORE MEMORY LOCATIONS
 - 2.3.3 REGISTER USAGE
 - 2.4 EXECUTION TIMES
- 3.0 ERROR INFORMATION
 - 3.1 ERROR REPORTING PROCEDURES
 - 3.1.1 STANDARD SYSMAC SML ERROR REPORTING CONVENTIONS
 - 3.1.2 ERROR MESSAGE TABLE
 - 3.1.3 DATA HEADER MNEUMONIC DEFINITIONS
 - 3.2 POWER FAIL PRINTOUT
 - 3.3 ERROR HALTS
- 4.0 PERFORMANCE AND PROGRESS REPORTS
 - 4.1 PERFORMANCE REPORTS
 - 4.2 PROGRESS REPORTS

- 5.0 DH11 DEVICE INFORMATION
- 5.1 ADDRESS AND VECTOR ASSIGNMENTS
- 5.2 REGISTER DEFINITIONS
 - 5.2.1 SYSTEM CONTROL REGISTER
 - 5.2.2 NEXT RECEIVED CHARACTER REGISTER
 - 5.2.3 LINE PARAMETER REGISTER
 - 5.2.4 CURRENT ADDRESS REGISTER
 - 5.2.5 BYTE COUNT REGISTER
 - 5.2.6 BUFFER ACTIVE REGISTER
 - 5.2.7 BREAK CONTROL REGISTER
 - 5.2.8 SILO STATUS REGISTER
- 5.3 DH11 FUNCTIONAL LOGIC PARTITIONING
- 5.4 DH11 MODULE ALLOCATION CHART
- 6.0 MAINTENANCE PROCEDURES
 - 6.1 INTRODUCTION
 - 6.2 PRELIMINARY CHECKS
 - 6.3 MAINTENANCE CONNECTORS
 - 6.4 COMPLETE DH11 SUB-SYSTEM CHECKOUT
 - 6.5 MAINTENANCE HEADER DESCRIPTION

1.0 GENERAL PROGRAM DESCRIPTION

SEQ 0004

1.1 PROGRAM PURPOSE

"MD-11-DZCHM" IS A COMPREHENSIVE DIAGNOSTIC TEST PROGRAM DESIGNED TO AID IN THE ACCEPTANCE TESTING, INSTALLATION CHECKOUT, AND CORRECTIVE MAINTENANCE OF THE DM11 16. LINE ASYNCHRONOUS SERIAL LINE MULTIPLEXOR. IT CONSISTS OF 44 LOGICALLY SEQUENCED DIAGNOSTIC TESTS DESIGNED TO TEST AND VERIFY THAT THE DM11 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 DM11'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 DM11 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 OPERATOR INTERVENTION (SRIO=1) OTHERWISE, THE TESTS ARE BYPASSED. (REFER TO SECTION 6.3)

T1 CHECK SSYN RESPONSE FROM ALL DM11 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 TRANSMITTER 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)


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


H01

SEQ 0007

* CONT. *

* CONT. *

EMI:

*
* 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. DH11-0 ENGINEERING DRAWINGS
- B. DH11 MANUAL EK-DH11-MM-002
- C. PDP11 PERIPHERALS HANDBOOK
- D. PDP11 PROCESSOR HANDBOOK
- E. MD-11-DZQAC-C1 SYSMAC.SML
- F. MD-11-DZQXA "XXDP" USER'S GUIDE
- G. DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS PROGRAMMING PRACTICES DOC NO. 175-003-009-00

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

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

- A. CPU/CORE MEMORY DIAGNOSTICS

1.5 FAILURE ASSUMPTIONS

SEQ 0009

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 POP11 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

SEQ 0010

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 = 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 PROGRAM
WILL TYPE AN ERROR MESSAGE AND ASK YOU TO
TRY AGAIN.

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

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

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

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

..

BIT15=1 TEST DH11 #15

EXAMPLES:

177777<CR> TEST ALL 16. DH11'S
100000<CR> TEST ONLY DH11 #17(9)
000005<CR> TEST DH11 #00 AND 02

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

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

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

..

BIT15=1 TEST LINE #15

EXAMPLES:

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

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

NOTE

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

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

"DEPRESS CONTINUE TO START TESTING"

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

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

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

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

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

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

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

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

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

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

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

NO1

SEQ 0013

SET THE SR=004000 (QUICK PASS)

SET THE SR=0020C0 (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 DM11 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 SRB=1, THE PROGRAM WILL HALT AND PRINT THE
FOLLOWING MESSAGE:

"DEPRESS CONTINUE TO START TESTING"

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

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

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

2.2 SPECIAL ENVIRONMENTS

- 2.2.1 ACT11 WHEN UNDER CONTROL OF THE ACT11 APT11
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 (16): 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 (10) 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 TYPEDS.
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. DM11 CONFIGURATION TABLES AND VARIABLES

WHEN THE AUTOSIZER OPTION IS USED, THIS PROGRAM CAN RUN NON-STANDARD DM11 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 DM11 #00 - WORD 00, DM11 #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. BRVL: 16. WORD BR LEVEL TABLE

THIS TABLE STORES THE BR LEVELS ASSUMED BY THE INTERRUPT SERVICE ROUTINES FOR EACH DM11. 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 DM11 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.

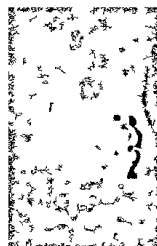
5. LINSEL: LINE SELECTION PARAMETER

THIS WORD IS PROGRAM LOADED AS A 177777(9) 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

SEQ 0018

THERE IS A LOCATION TAGGED "SMXONT:" 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.C 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 9 OCTAL OR DECIMAL NOS.)

EXAMPLE:

SYSTEM CONTROL REGISTER ERROR
 (FC) (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-DZDWM AND ARE INTERPRETED AS FOLLOWS:

EM ADDRESS OF THE MESSAGE FOR LINE 1
 DH ADDRESS OF THE DATA HEADER MESSAGE FOR LINE 2
 DT ADDRESS OF THE TABLE OF ADDRESSES THAT POINT
 TO THE DATA WORDS TO BE PRINTED
 DF ADDRESS THAT POINTS TO THE DATA DESCRIPTOR
 TABLE THAT DEFINES WHETHER AN ITEM IS OCTAL OR DECIMAL.
 IF THIS ENTRY IS "O" 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, DZDWM, 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      : $ERRPC,$TMPD,$REG6,$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      : $ERRPC,$TMPD,$REG6,$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      : $ERRPC,$TMPD,$REG6,$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      : $ERRPC,$TMPD,$REG6,$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      : $ERRPC,$TMPD,$REG6,$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      : $ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
3        : 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      : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2
0         : PRINT ALL OCTAL

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 31

```

EM31      : "AUJ ECHO DATA COMPARE ERROR - LINE # "
DH4       : " (PC) (PS) (SP) TEST WASADR SBADR WAS S/B "
DT2       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$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       : $ERRPC,$TMPD,$REG6,$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       : $ERRPC,$TMPD,$REG6,$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

```


M02

SEQ 0025

;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       : $ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
0         : PRINT ALL OCTAL

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 45

```

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

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 46

```

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

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 47

```

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

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 50

```

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

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 51

```

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

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 52

```

EM52      : "BASIC DATA COMPARE ERROR"
DH2       : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B"
DT2       : $ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
0         : PRINT ALL OCTAL

```

:ERROR TABLE ITEM FOR ERROR MESSAGE 53

SEQ 0026

```

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

```

3.1.3 DATA HEADER MNEUMONIC DEFINITIONS

SEQ 0027

ALL NUMBERS PRINTED AS ERROR DATA ARE IN OCTAL

(PC) ADDRESS OF THE ERROR CALL (ERROR PC)
 (PS) CONTENTS OF THE PSW AT THE TIME OF THE ERROR
 (SP) CONTENTS OF THE STACK POINTER AT THE TIME OF THE ERROR
 TEST TEST NUMBER
 DEVADR DEVICE ADDRESS - 1ST ADDRESS IN THE SELECTED DH11
 REGADR ADDRESS OF THE DH11 REGISTER BEING TESTED
 WAS WHAT THE ACTUAL DATA READ WAS (DH11 REG OR CORE LOC.)
 S B WHAT THE DATA READ SHOULD HAVE BEEN
 SPEED SPEED CODE IN THE "LPR" REG AT THE TIME OF THE ERROR
 REFER TO SECTION 5.2.3 FOR SPEED CODE TABLES
 TIMEB CONTENTS OF SOFTWARE COUNTER USED IN TIMING TESTS
 TIMEC CONTENTS OF SOFTWARE COUNTER USED IN TIMING TESTS
 NOTE: "TIMEB" SHOULD ALWAYS BE LESS THAN "TIMEC"
 CHARLG CHARACTER LENGTH CODE IN THE "LPR" AT THE TIME OF THE ERROR
 00=5 BITS, 01=6 BITS, 02=7 BITS, 03= 8 BITS
 TRPFC 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

SEC 0028

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
PDF11 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 SRC=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 (COTAL). 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 INITIALIZATION 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 INTERRUPT 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 00

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

02 TWO STOP BITS

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

03 NOT USED

04 PARITY ENABLED

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

05 ODD PARITY

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

06-09 RECEIVER SPEED

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

10-13 TRANSMITTER SPEED

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

SPEED TABLE FOR RECEIVER AND TRANSMITTER SPEEDS:

SEQ 0035

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

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

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

5.2.7 BREAK CONTROL REGISTER ADDRESS X14

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

5.2.8 SILO STATUS REGISTER ADDRESS X16

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.

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

M7277 CURRENT ADDR REG MEMORY AND ADDR SELECT

- SH3 CONTROL STROBE MUX FOR THE "LPR" REGISTER
- TRANSMITTER DATA MUX WITH AUTO-ECHO CONTROL LOGIC SELECTION
- MSYN / SSYN TIMING CHAIN
- DH11 MASTER CLEAR LOGIC
- SH4: UNIBUS ADDRESS SELECTION LOGIC WITH JUMPERS
- TRANSMITTER SCAN COUNTER WITH XMITTER STATUS MULTIPLEXOR (BAR N * TBMT N)
- SH5: BYTE COUNT AND CURRENT ADDRESS MEMORY WRITE TIMING LOGIC
- CURRENT ADDRESS MEMORY LOGIC BITS<17:08>
- UNIBUS ADDRESS DRIVERS BITS <17:08>
- SH6: BYTE COUNT AND CURRENT ADDRESS MEMORY ADDRESS SELECT MULTIPLEXOR
- CURRENT ADDRESS MEMORY LOGIC FOR BITS <07:00>
- UNIBUS ADDRESS DRIVERS FOR BITS <07:00>
- TRANSMITTER EVEN/ODD BYTE DATA MULTIPLEXOR

M7279 FIFO BUFFER

- SH1: INPUT DATA MULTIPLEXOR FOR SILO MEMORY
- SH2: SILO MEMORY CHIPS (FOUR 64 X 4 CHIPS)
- SILO MEMORY READ/WRITE TIMING LOGIC
- "SSR" REGISTER BITS <13:08>
- SILO ALARM LEVEL COMPARATOR
- 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

M7280 #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 MULTIPLEXORS (MASTER FE AND MASTER PE)
SH5: UART CHIPS BIT<15:14>
TRANSMITTER SCAN MULTIPLEXOR
-12 VOLT DC REGULATOR

M7289 SYSTEM CONTROL AND RCVR SCAN

SH3: TRANSMITTER AND RECEIVER SCAN LOGIC
SH4: TRANSMITTER AND RECEIVER SCAN TIMING
SH5: HALF-DUPLEX CONTROL LOGIC
UART DATA MULTIPLEXORS
SH6: SYSTEM CONTROL REGISTER

M4540 DC11 - CH11 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

SEQ 0041

SH3: CLEAR "BAR" REG MULTIPLEXOR

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

UNIBUS RECEIVERS BIT<15:08>

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

UNIBUS DATA RECEIVERS BIT<07:00>

SH5 - SH8: "BAR", "BCR", "LPR", AND "SSR" REGISTERS PLUS THE
DATA OUTPUT MUX AND UNIBUS DRIVERS FOR DATA LINES.SH5: BIT<15:12> SH7: BIT<07:04>
SH6: BIT<11:09> SH8: BIT<03:00>

D04

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 A	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		
B		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

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. EQ2 MUST BE G727 GRANT CONTINUITY IF MODEM CONTROL MODULE SET IS NOT INSTALLED. * DENOTES DM11-BB MODEM CONTROL OPTION, WITH DH11-AA OR AC.
5. MODULE SLOTS PROVIDE FOR ADDITIONAL CLOCK RATES.
6. FOR DIAGNOSTIC CHECKOUT OF DH11-AA, AB, OR AC, REPLACES M971 WITH M974.
7. THIS SLOT CONTAINS MODEM CONTROL MODULE M7807 WITH DH11-AD.
8. THIS SLOT CONTAINS MODEM CONTROL MODULE M7808 WITH DH11-AD.
9. THIS SLOT CONTAINS EIA CONVERTER AND PRIORITY MODULE M5906 FOR DH11-AD OR AE.

5.0 MAINTENANCE PROCEDURES
-----5.1 INTRODUCTION

THIS SECTION DESCRIBES HOW TO USE MD-DZDMM 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.

5.2 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.

5.3 MAINTENANCE CONNECTORS

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

A. DH11-AA, AB, OR AC CONFIGURATIONS

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

- A. REMOVE THE DATA CABLE FROM SLOT B7 IN EACH DH11 TO BE TESTED.
 - B. INSTALL AN M974 MAINT JUMPER MODULE INTO SLOT B7 OF EACH DH11 TO BE TESTED.
- 2) TESTING ALL 16. LINES INCLUDING DATA CABLES WHICH CONNECT TO DISTRIBUTION PANEL. DOES NOT TEST LEVEL CONVERTER CIRCUITS LOCATED IN DISTRIBUTION PANEL.
 - A. INSTALL THE M974 MAINT JUMPER MODULE INTO SLOT B3 OF THE MULTIPLEXOR DISTRIBUTION PANEL FOR EACH DH11 TO BE TESTED. ALL LEVEL CONVERTERS IN THE DISTRIBUTION PANEL MUST BE REMOVED FOR THIS TEST.
 - 3) TESTING ONE OR MORE SINGLE LINES INCLUDING EIA LEVEL CONVERTERS AND DEVICE CABLES WHICH ARE NOT TESTED IN 1 AND 2 ABOVE.

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

B. DH11-AD CONFIGURATION

1. TESTING ALL 16. LINES WITHOUT DATA CABLES
 - A. DISCONNECT THE DATA CABLES (2) FROM THE TWO CONNECTORS ON THE M5906 MODULE (SLOT AB7 OF THE DH11 BACKPLANE).
 - B. INSTALL TWO H8611 TEST CONNECTORS ON THE M5906 IN PLACE OF THE CABLES.
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-DZDMM 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.



J04

EXAMPLE: MAINTENANCE HEADER FOR TEST 1

SEQ 0048

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 11 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	EO9-11

!

FLOW CHART

MD-11-DZDHM-B DH11 DIAGNOSTIC

COPYRIGHT 1976
DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

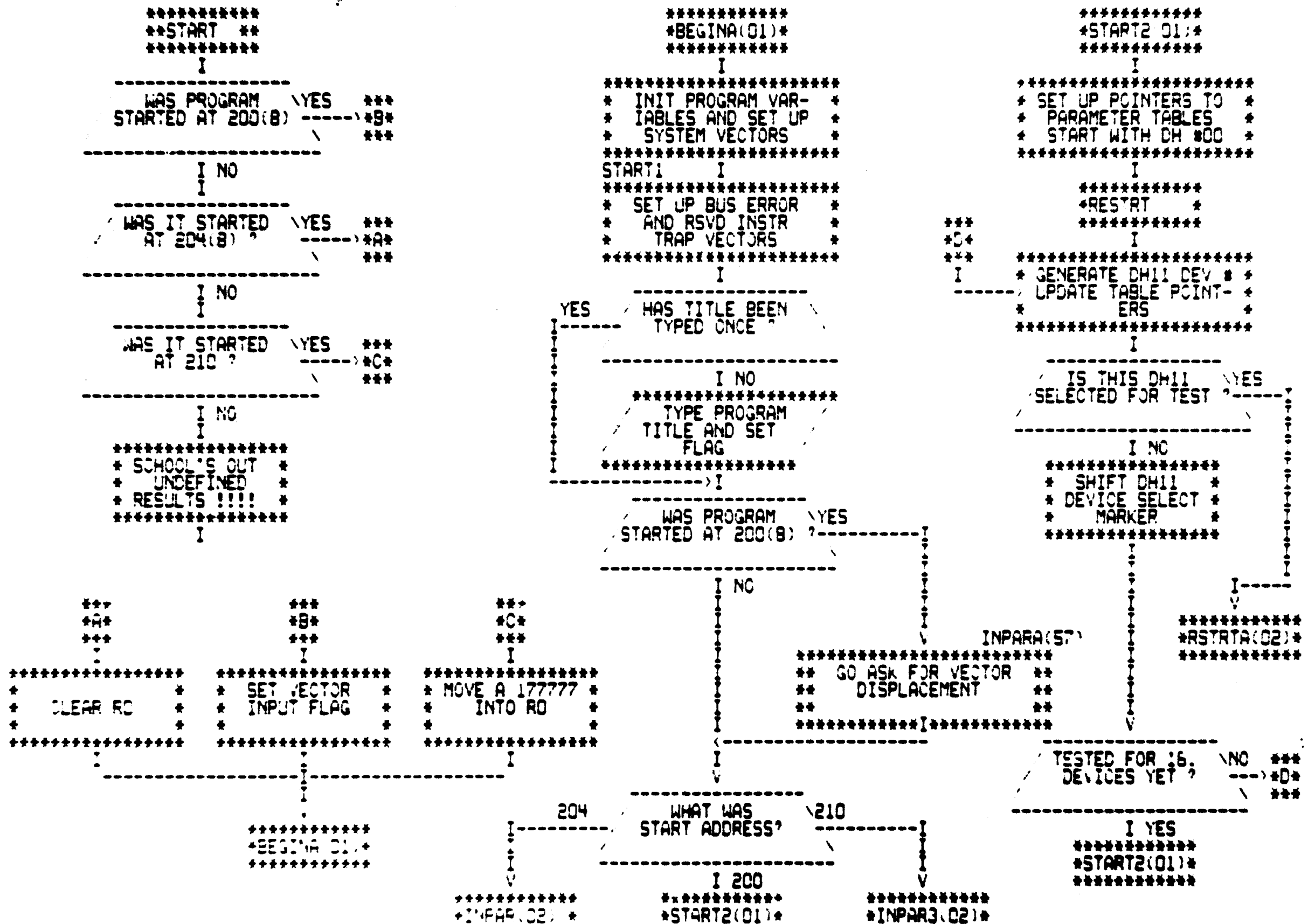
PAGE 01	PROGRAM START-UP 1
PAGE 02	PROGRAM START-UP 2
PAGE 03	TEST 01 CHECK SSYN RESPONSE FROM ALL DH11 REGISTERS
PAGE 04	TEST 02 TEST THAT "MSTCLR" CAN CLEAR "SCR", "LPR", "BKR", AND "SSR" REGS
PAGE 05	TEST 03 TEST "SCR" REG R/W BITS CAN SET/CLR (NORMAL MODE)
PAGE 06	TEST 04 TEST "SCR" REG READ ONLY BITS (NORMAL MODE)
PAGE 07	TEST 05 TEST "SCR" REG BITS THAT CAN BE SET/CLR IN MAINT MODE
PAGE 08	TEST 06 TEST THAT ALL R/W BITS IN "LPR" CAN BE SET/CLR
PAGE 09	TEST 07 TEST THAT ALL R/W BITS IN "BKR" CAN BE SET/CLR
PAGE 10	TEST 10 TEST THAT ALL R/W BITS IN "SSR" CAN BE SET/CLR
PAGE 11	TEST 11 TEST THAT CLR/SET OF BIT "N" IN "LPR" DOES NOT CLEAR ANY OTHER BITS
PAGE 12	TEST 12 TEST THAT CLR/SET OF BIT "N" IN "BKR" DOES NOT CLEAR ANY OTHER BITS
PAGE 13	TEST 13 TEST THAT CLR/SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER BITS
PAGE 14	TEST 14 "CAR" MEMORY ADDRESSING TEST
PAGE 15	TEST 15 "BCR" MEMORY ADDRESSING TESTS
PAGE 16	TEST 16 "CAR" REGISTER TEST - ALL 1'S/ALL 0'S/ALL LINES
PAGE 17	TEST 17 "BCR" REGISTER TEST - ALL 1'S/ALL 0'S/ALL LINES
PAGE 18	TEST 20 "CAR" MEMORY PATTERNS TEST - 0'S DISTURB
PAGE 19	TEST 21 "BCR" MEMORY PATTERNS TEST - 0'S DISTURB
PAGE 20	TEST 22 "CAR" MEMORY PATTERNS TEST - 1'S DISTURB
PAGE 21	TEST 23 "BCR" MEMORY PATTERNS TEST - 1'S DISTURB
PAGE 22	TEST 24 "CAR" EXT MEMORY BITS TEST
PAGE 23	TEST 25 TEST INTR. ENAB. BITS - INTR. CONDITION DISABLED
PAGE 24	TEST 26 TEST CHAR. AVAIL. I.E. WITH INTR. CONDITION ACTIVE

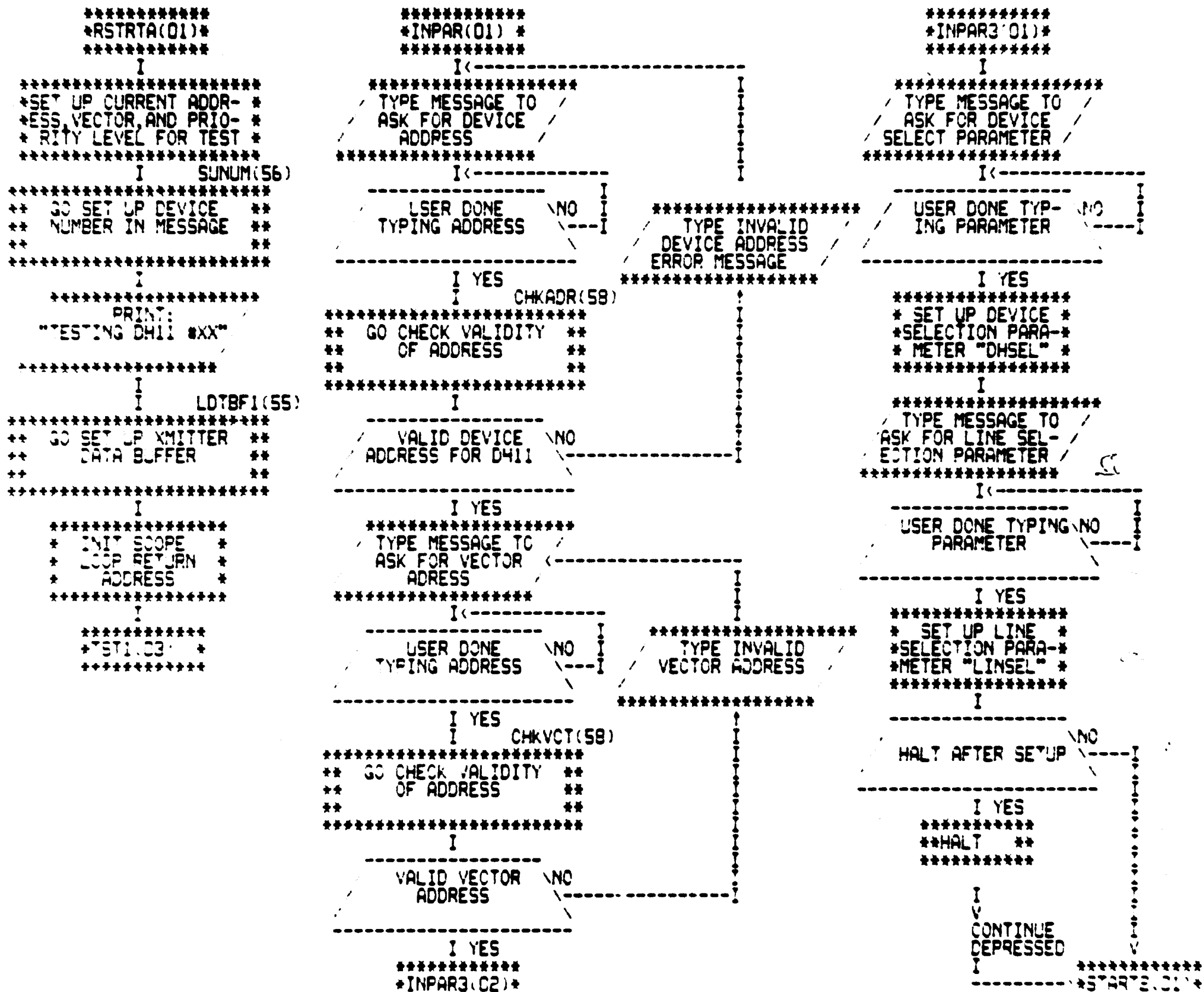
TABLE OF CONTENTS

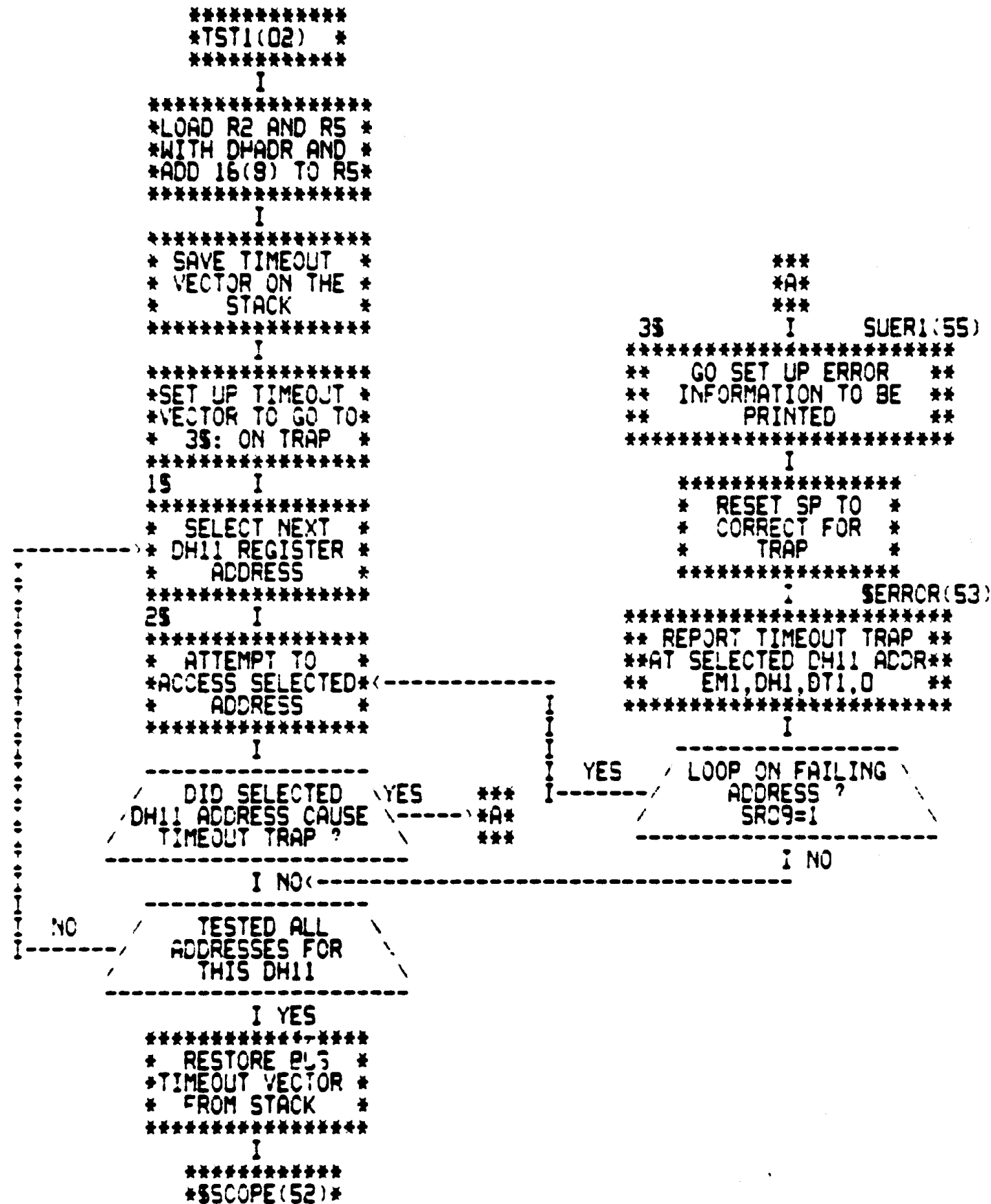
PAGE 26	TEST 30 TEST NON-EX MEM I.E. WITH INTR. CONDITION ACTIVE
PAGE 27	TEST 31 TEST TRANSMIT DONE I.E. WITH INTR. CONDITION ACTIVE
PAGE 28	TEST 32 BASIC "NPR" LOGIC TEST 1
PAGE 29	TEST 32 BASIC "NPR" LOGIC TEST 1 (CONT.)
PAGE 30	TEST 33 BASIC "NPR" LOGIC TEST 2
PAGE 31	TEST 34 TEST THAT CHAR. AVAIL. CAN GENERATE RCVR. INTR.
PAGE 32	TEST 35 TEST THAT SILO STATUS REGISTER COUNTS UP CORRECTLY
PAGE 33	TEST 36 TEST THAT SILO STATUS REG DOWN COUNTS CORRECTLY
PAGE 34	TEST 37 TEST SILO ALARM LEVEL
PAGE 35	TEST 40 VERIFY STORAGE OVERFLOW - NON MAINT MODE - ALL LINES
PAGE 36	TEST 41 TRANSMITTER TIMING TEST - ALL LINES - ALL SPEEDS
PAGE 37	TEST 42 RECEIVER TIMING TEST - ALL LINES - ALL SPEEDS
PAGE 38	TEST 43 BASIC DATA TEST - ALL LINES - ALL CHAR. LENGTHS
PAGE 39	TEST 44 SINGLE LINE DATA TEST-ALL LINES (PAGE 1)
PAGE 40	TEST 44 SINGLE LINE DATA TEST-ALL LINES (PAGE 2)
PAGE 41	TEST 45 BASIC PARITY LOGIC TEST - ALL LINES - ODD PARITY
PAGE 42	TEST 46 MULTI-LINE PARITY-DATA TEST - PAGE 1
PAGE 43	TEST 46 MULTI-LINE PARITY TEST - PAGE 2
PAGE 44	TEST 47 AUTO-ECHO TEST 1-ALL LINES
PAGE 45	TEST 50 AUTO-ECHO TEST 2-ALL LINES (PAGE 1)
PAGE 46	TEST 50 AUTO ECHO TEST 2-ALL LINES (PAGE 2)
PAGE 47	TEST 51 AUTO-ECHO TEST-3
PAGE 48	TEST 52 BREAK BIT TEST-ALL LINES (PAGE 1)
PAGE 49	TEST 52 BREAK BIT TEST-ALL LINES (PAGE 2)

TABLE OF CONTENTS

PAGE 51	TEST 54 VERIFY THAT OVERRUN CAN SET PROPERLY-ALL LINES.
PAGE 52	SCOPE LOOP ROUTINE
PAGE 53	ERROR SERVICE ROUTINE
PAGE 54	COMMON DH11 SUB-ROUTINES - PAGE 1
PAGE 55	COMMON DH11 SUB-ROUTINES - PAGE 2
PAGE 56	COMMON DH11 SUB-ROUTINES - PAGE 3
PAGE 57	COMMON DH11 SUB-ROUTINES - PAGE 4
PAGE 58	COMMON DH11 SUB-ROUTINES - PAGE 5
PAGE 59	COMMON DH11 SUBROUTINES - PAGE 6







E05

NO-11-DZDMM-B DH11 DIAGNOSTIC
TEST 02 TEST THAT "MSTCLR" CAN CLEAR "SCR", "LPR", "BKR", AND "SSR" REGS

DECFL0 VER 00.12 23-SEP-76 14:43 PAGE 04

SEQ 0056

```
*****
* TST2
*****
I
*****
* SET UP TABLE OF TEST *
* ADDRESSES FOR "SCR" *
* "LPR", "BKR", AND "SSR" *
*****
```

```
I
*****
* MAKE S/B = 0 *
*
*****
1$ I
*****
* LOAD R2 WITH *
* TEST ADDRESS *
* FROM TABLE *
*****
```

```
I
-----
YES / TESTED ALL FOUR \
:----- TEST ADDRESSES ?
```

```
*****
*SSOCFE(52)*
*****
```

```
I NO
23 I
*****
* SET ALL WRITE *
* BITS AT TEST *
* ADDRESS *
*****
I
*****
* SET "MSTCLR" *
* BIT IN SCR REG *
*****
```

```
I
-----
/ DID "MSTCLR" \ NO ***
/ CLEAR ALL BITS AT \--> *A*
/ TEST ADDRESS \
-----
I YES
```

```
***
*A*
***
I SUER2(55)
```

```
*****
** GO SET UP ERROR **
** INFORMATION TO BE **
** PRINTED **
*****
I $ERROR(53)
*****
** REPORT "MSTCLR" **
** FAILED TO CLEAR REG **
** EM3,DH2,DT2,0 **
*****
```

```
I
-----
YES / LOOP ON FAILING \
:----- ADDRESS SR09=1
```

```
I NO
```

```

***
**A*
***
I      SUER2(55)
*****
**SET UP ERROR INFORM- **
** ATION TO BE PRINTED **
**                          **
*****
I      $ERROR(53)
*****
** REPORT "SCR" REG **
** BIT FAILURE **
**                          **
*****
I
-----
*** NO / LOOP ON FAILING \ ***NO / LOOP ON FAILING \
*C*<---- / BIT ? SR09=1 \ *C*<---- / BIT ? SRC9=1 \
*** / EM2,DH2,DT2,C \ *** / EM2,DH2,DT2.C \
-----
I YES
|
|
|
|
|
|
|
|
|
|
V

```

```

/ DID SELECTED \ NO ***
BIT SET PROPERLY \ ----> **
                  \ ***

```

```

***** I YES *****
* NOW CLEAR THE *
* SELECTED BIT *
*               *
*****

```

```

*** YES / DID SELECTED \ NO ***
*C* -----/ BIT CLEAR \----->*B*
*** / PROPERLY ? \ ***

```

```

*****
*TST4      *
*****
I
*****
* SET UP REGADR, AND *
* BIT TEST MARKER   *
*****
-----> I
IS THIS A READ ONLY BIT ?
***
** NO ***
-----> I NO
2$
*****
*SELECT NEXT BIT*
*POSITION TO THE*
* LEFT          *
*****
I
YES ANY BITS LEFT TO TEST ?
NO
-----> I
3$
*****
*TRY TO SET THE *
* READ ONLY BIT *
*               *
*****
I
*** NO ***
**A** DID THE READ ONLY BIT SET ?
***
-----> I IS
***
*3*

```

```

*****  

*TST5  

*****  

I  

*****  

*SET UP REGADR. ERROR *  

* LOOP RETURN, AND A *  

* BIT TEST MARKER *  

*****  

-----> I  

IS  

CAN BIT BE SET IN MAINT. MODE ?  

YES  

***  

*B*  

***  

2S  

I  

*****  

*SELECT NEXT BIT *  

*POSITION TO THE *  

* LEFT *  

*****  

I  

ANY BITS LEFT TO TEST ?  

YES  

I NO  

V  

*****  

*SCOPE(52)*  

*****  

E--> I  

3S  

V  

*****  

* ENABLE MAINT. *  

* MODE AND SET *  

* THE BIT *  

*****  

I  

DID SELECTED BIT GET SET IN MAINT. MODE ?  

NO  

C*  

***  

4S  

I YES  

*****  

*DISABLE MAINT. *  

*MODE AND TRY TO *  

* CLEAR THE BIT *  

*****  

I  

DID THE BIT CLEAR WITH MAINT. MODE TURNED OFF ?  

YES  

D*  

***  

A*  

***  

*****  

**A**  

***  

5S  

I  

*****  

* ENABLE MAINT *  

*MODE AND CLEAR *  

* THE BIT *  

*****  

I  

DID THE BIT GET CLEARED ?  

NO  

B*  

***  

I YES  

B*  

***  

V  

SJER2(55)  

*****  

**SET UP ERROR INFORM- **  

** ACTION TO BE PRINTED **  

*****  

I  

$ERROR(53)  

*****  

** REPORT THE "SCR" **  

**REGISTER BIT FAILURE **  

** EM2,DH2,DT2,0 **  

*****  

I  

LOOP ON FAILING BIT ?  

YES  

V  

E*  

B*  

***  

NO  

B*  

***  

I  

SUER2(55)  

*****  

** GO SET UP ERROR **  

** INFORMATION TO BE **  

** PRINTED **  

*****  

I  

$ERROR(53)  

*****  

** REPORT THE "SCR" **  

** REG BIT FAILURE **  

** EM2,DH2,DT2,0 **  

*****  

I  

LOOP ON FAILING BIT ?  

YES  

V  

E*  

B*  

***  

NO  

B*  

***  

I  

SUER2(55)  

*****  

** GO SET UP ERROR **  

** INFORMATION TO BE **  

** PRINTED **  

*****  

I  

$ERROR(53)  

*****  

** REPORT THE "SCR" **  

** REG BIT FAILURE **  

** EM2,DH2,DT2,0 **  

*****  

I  

LOOP ON FAILING BIT ?  

YES  

V  

E*  

B*  

***  

NO  

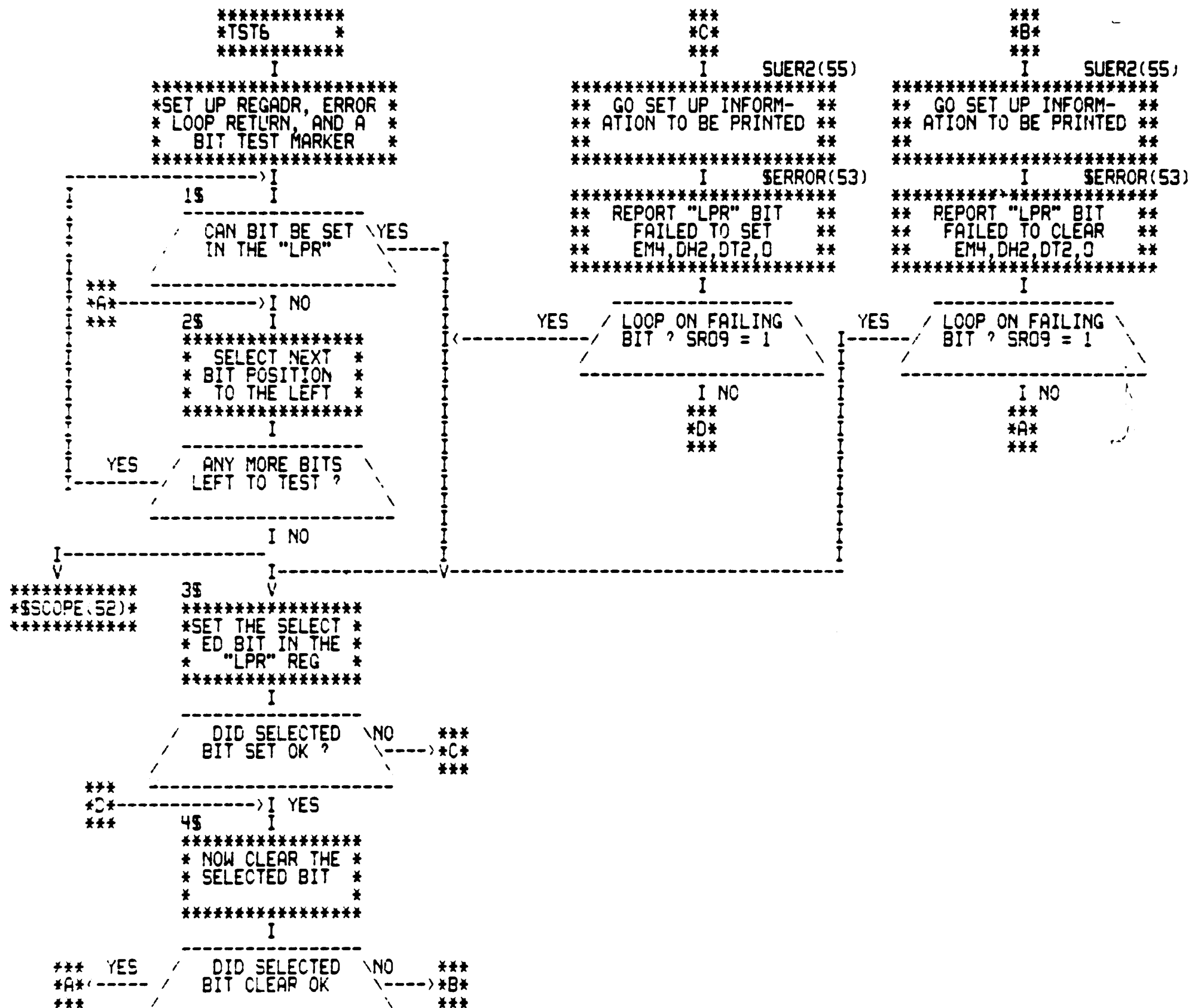
B*  

***  

I  

SUER2(55)

```



```

***
**C*
***
I SUER2(55)
*****
** GO SET UP INFORM- **
** ATION TO BE PRINTED **
**
*****
I $ERROR(53)
*****
** REPORT "LPR" BIT **
** FAILED TO SET **
** EM4,DH2,DT2,0 **
*****

```

```

***
**B*
***
I SUER2(55)
*****
** GO SET UP INFORM- **
** ATION TO BE PRINTED **
**
*****
I $ERROR(53)
*****
** REPORT "LPR" BIT **
** FAILED TO CLEAR **
** EM4,DH2,DT2,0 **
*****

```

YES / LOOP ON FAILING
BIT ? SRO9 = 1 \

YES / LOOP ON FAILING
BIT ? SRO9 = 1 \

```

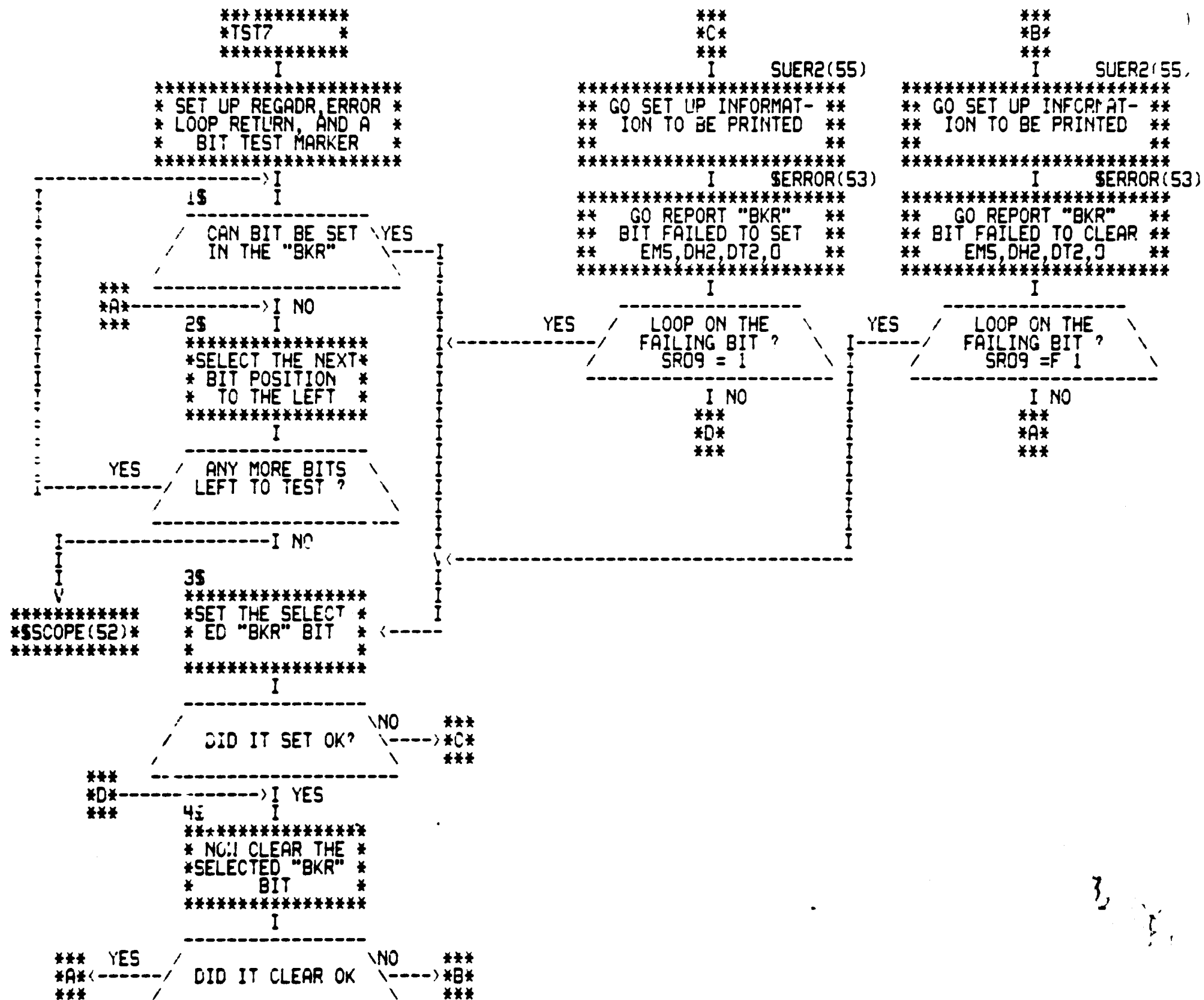
I NO
***
**D*
***

```

```

I NO
***
**A*
***

```




```

*****
*TST10 *
*****
I
*****
*SET UP REGADR, ERROR *
* LOOP RETURN, AND A *
* BIT TEST MARKER *
*****
----->I
I
-----
CAN THIS "SSR" \ YES
BIT BE SET ? /-----I
***
***A*----->I NO
*** *****
*** *SELECT THE NEXT*
*** *BIT POSITION TO*
*** * THE LEFT *
*** *****
I
-----
YES / ANY BITS LEFT \
I----- TO TEST ? /-----I
I
I NO
I
I-----<-----
V
*****
*SSCOPE(52)*
*****
*****
*SET THE SELECT-*
* ED "SSR" BIT *
* *
*****
I
-----
/ DID IT SET OK ? \ NO ***
***C*
***
***D*----->I YES
*** *****
*** * NOW CLEAR THE *
*** *SELECTED "SSR" *
*** * BIT *
*** *****
I
-----
*** YES / \ NO ***
***A*----- DID IT CLEAR OK? \----->*B*
***

```

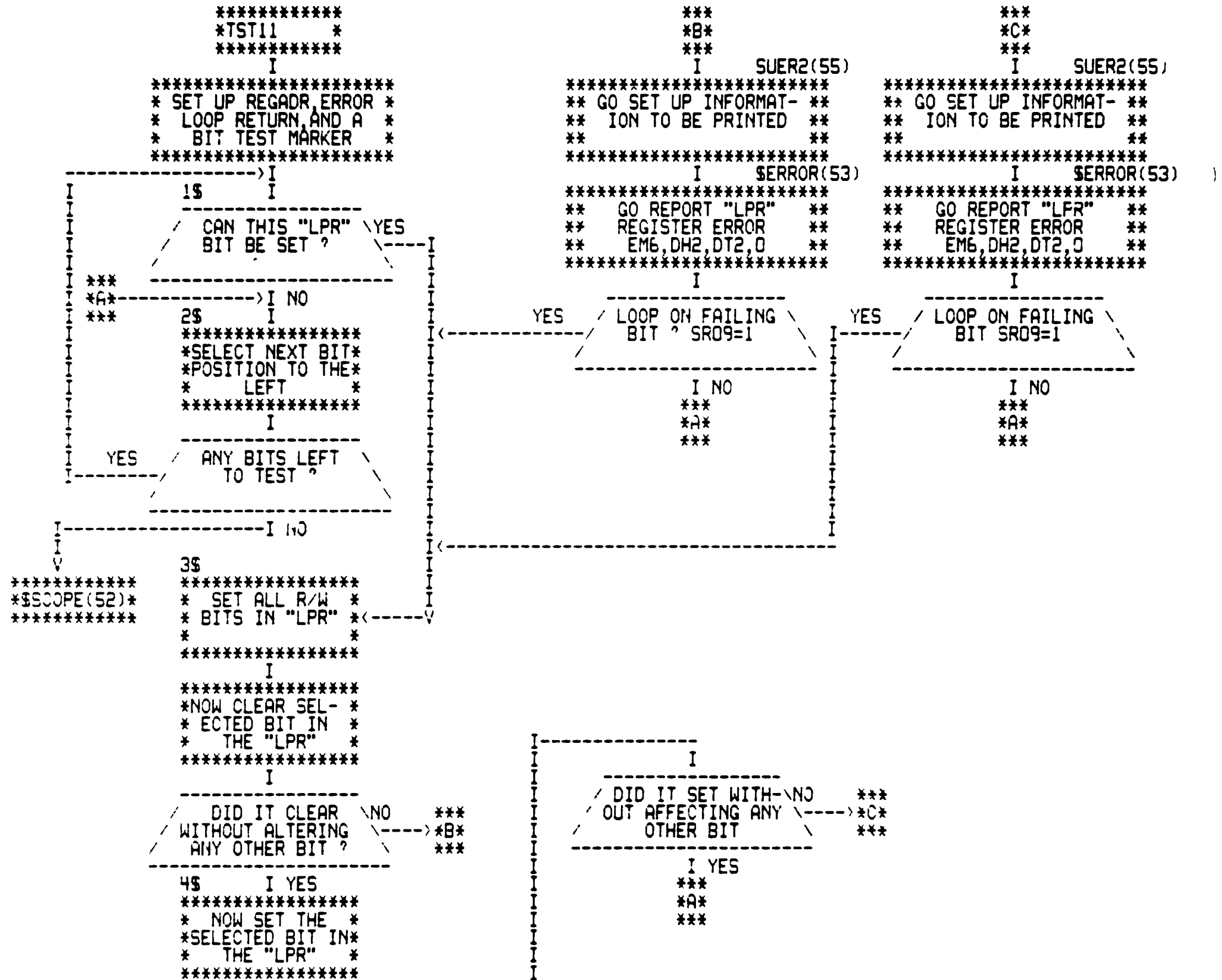
L05

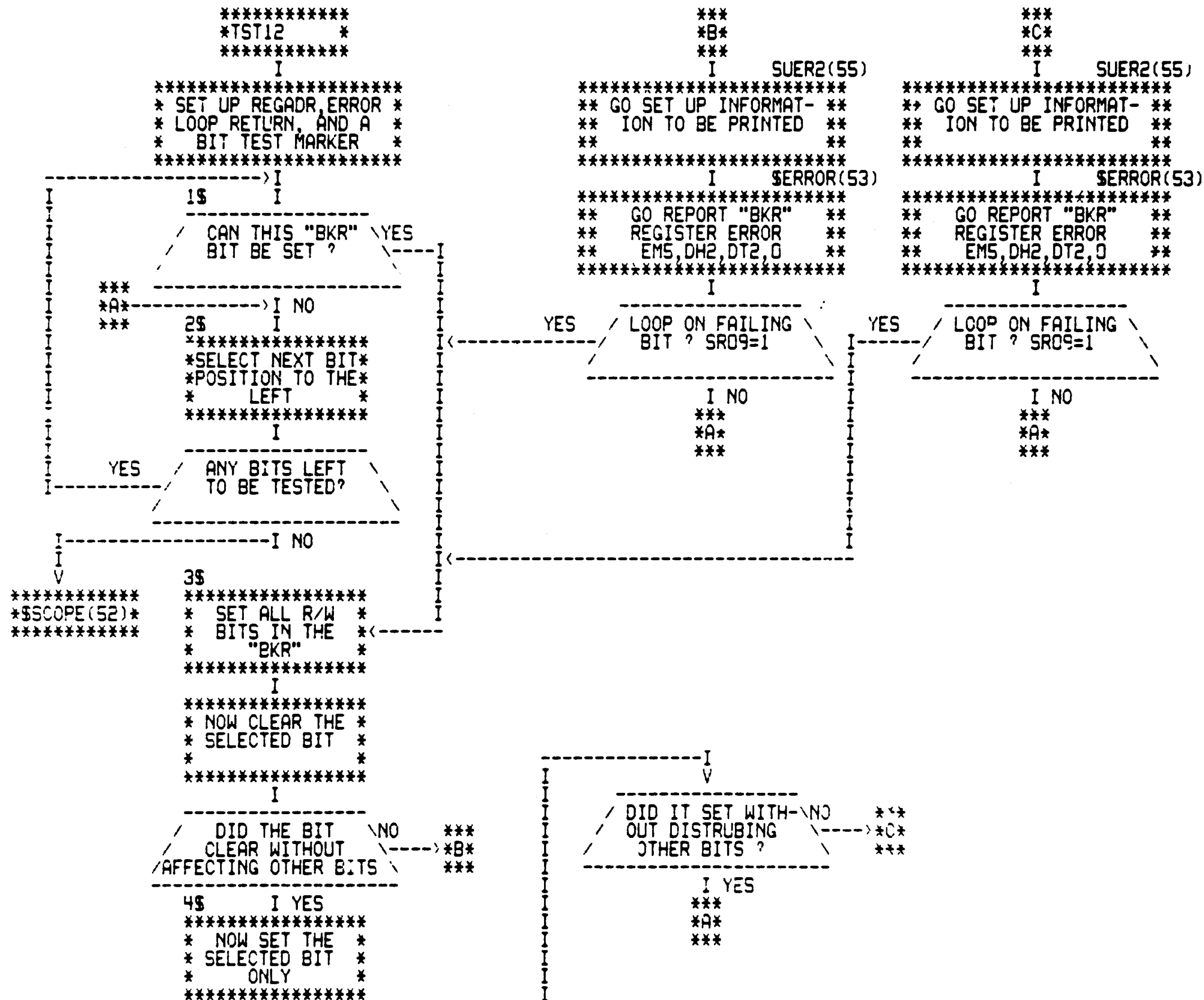
MD-11-DZDMM-8 DH11 DIAGNOSTIC

DECFL0 VER 00.12 23-SEP-76 14:43 PAGE 11

TEST 11 TEST THAT CLR/SET OF BIT "N" IN "LPR" DOES NOT CLEAR ANY OTHER BITS

SEQ 0063





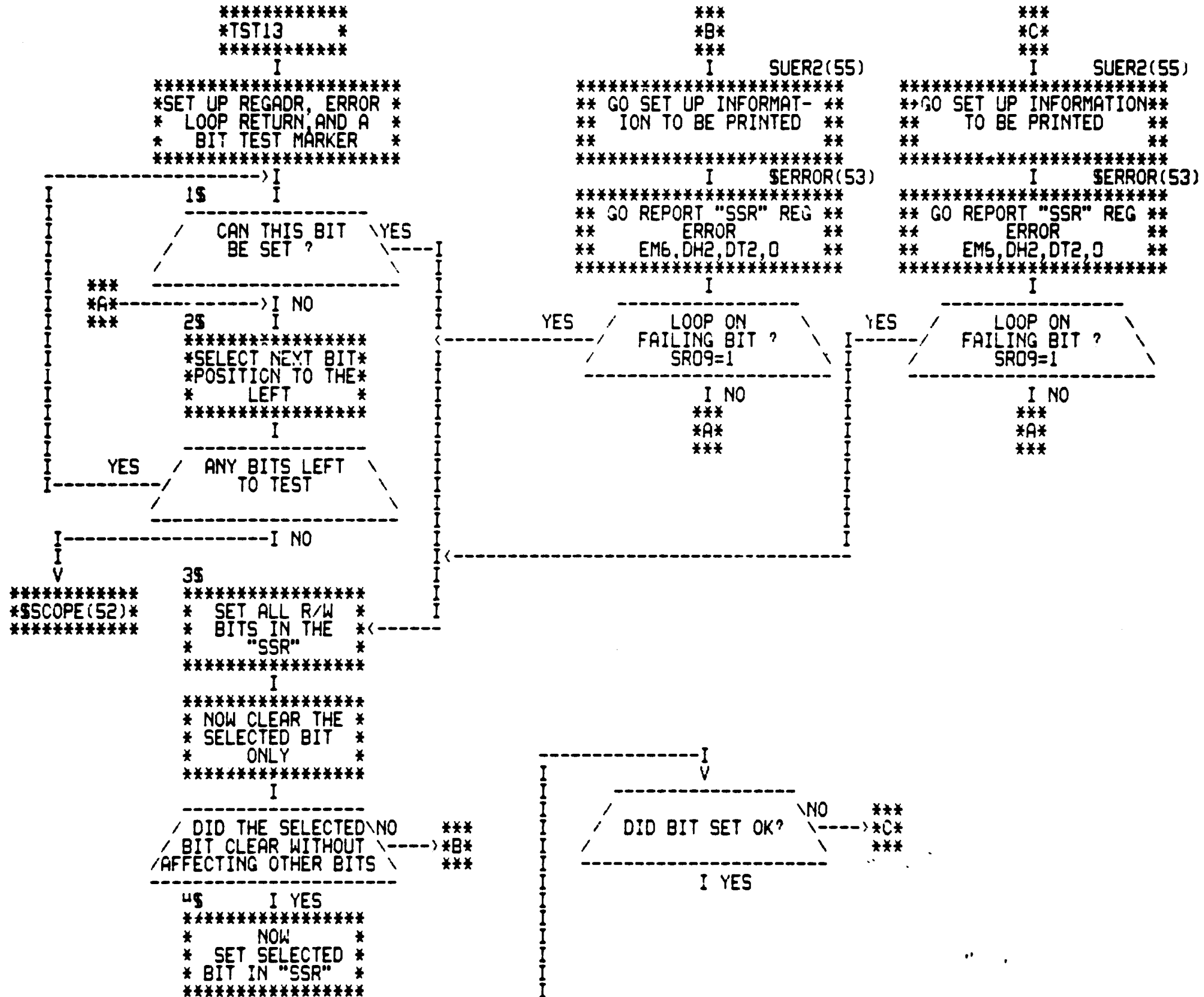
N05

MD-11-DZDMM-B DH11 DIAGNOSTIC

DEC FLO VER 00.12 23-SEP-76 14:43 PAGE 13

TEST 13 TEST THAT CLR/SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER BITS

SEQ 0065



```

*****
*TST14
*****
I
*****
* SET UP REGADR, AND *
* SAVE LINE SELECT *
* PARAMETER IN STMP7 *
*****

```

```

*****
*SET UP TO CHECK*
* ALL 16. LINES *
*
*****

```

```

15
*****
* LOAD "CAR" MEMORY *
* WITH ADDRESS TEST *
* PATTERN *
*****

```

```

35
*****
*GO SELECT A LINE NO. *
* TO CHECK *
*****

```

```

YES
DONE CHECKING
ALL 16. LINES

```

```

*****
* RESTORE LINE *
* SELECTION PARA- *
* METER FM STMP7 *
*****

```

```

*****
*SCOPE (52)*
*****

```

```

45
*****
*READ "CAR" FOR *
* SELECTED LINE *
*****

```

```

*****
/ CORRECT PATTERN \ NO
/ READ FROM SELECT- \
ED LINE ? \
*****

```

```

55
*****
*UPDATE EXPECTED*
* TEST PATTERN *
* FOR NEXT LINE *
*****

```

```

***
*A*
***
I
SUEP2(55)
*****
** GO SET UP INFORMAT- **
** ION TO BE PRINTED **
**
*****
I
SUNUM(56)
*****
**GO SET LINE NO. INTO **
** ERROR MESSAGE **
**
*****

```

```

*****
* SET UP ERROR *
* LOOP RETURN *
*
*****

```

```

*****
I
SERROR(53)
*****
** GO REPORT "CAR" **
** ADDRESSING ERROR **
** EM7,DH2,DT2,0 **
*****

```

```

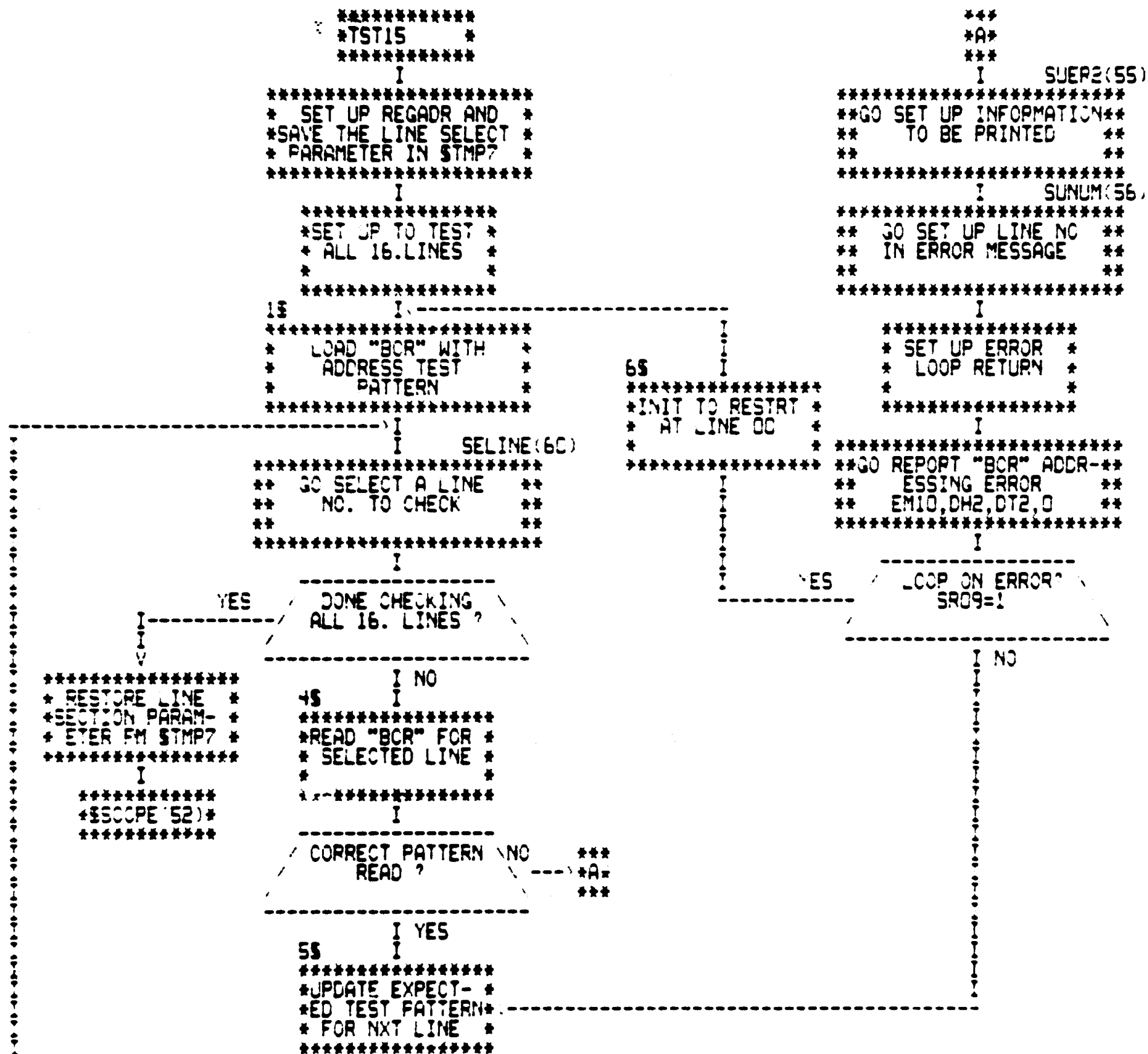
YES
LOOP ON ERROR
SRC9=1

```

```

NO

```



006

MD-11-DZDMM-B DH11 DIAGNOSTIC
TEST 16 "CAR" REGISTER TEST - ALL 1'S ALL 0'S ALL LINES

DECFL0 VER 00.12 23-SEP-76 14:43 PAGE 16

SEQ 0068

TST16

I

* SET UP REGADR *

15 I SELINE(60)

**GO SELECT A LINE NO. **
** TO TEST **

YES
I-----
*****:*****
SCOPE.521 25 I NO

*** *WRITE ALL ONES *
C---->* INTO SELECTED *
*** * "CAR" *

I

*READ BACK SE- *
* ECTED "CAR" *
* *

NO
/ WAS IT ALL ONES? \----> *A*

35 I YES

*** * NOW CLEAR THE *
D---->* SELECTED "CAR" *
*** * *

I

* READ BACK THE *
* SELECTED "CAR" *
* *

YES / WAS IT ALL ZEROS \----> *B*

A

I SUER2(55)

GO SET UP INFORMATION
** TO BE PRINTED **
** **

I SERROR(53)

** GO REPORT "CAR" **
** DATA ERROR **
** **

***YES
C----> LOOP ON FAILING
*** LINE ? SRO9=1

I NO

C

B

I SUER2(55)

GO SET UP INFORMATION
** TO BE PRINTED **
** **

I SERROR(53)

** GO REPORT "CAR" **
** DATA ERROR **
** **

***YES
C----> LOOP ON FAILING
*** LINE ? SRO9=1

I NO

C

E06

MC-11-DZDMM-B DH11 DIAGNOSTIC
TEST 17 "BCR" REGISTER TEST - ALL 1'S ALL 0'S ALL LINES

DECFL0 VER 00.12 23-SEP-76 14:43 PAGE 17

SEQ 0069

```

*****
*TST17
*****
I
*****
* SET LP REGADR *
*****
> I<
1$ I SELINE(60)
*****
**GO SELECT A LINE NO. **
** TO TEST **
*****
I
-----
YES / TESTED ALL \
I / SELECTED LINES? \
V
***** 2$ I NO
**SCOPE(52)* *****
**WRITE ALL ONES *
** INTO SELECTED *
** "BCR" *
**C*----->*
*****
I
*****
* READ BACK *
*SELECTED "BCR" *
*
*****
I
-----
/ WAS IT ALL ONES \ NO ***
/ ? \ ***
/
3$ I YES
*****
**WRITE ALL ZEROS*
** INTO SELECTED *
** "BCR" *
**D*----->*
*****
I
*****
* READ BACK *
*SELECTED "BCR" *
*
*****
I
-----
YES / WAS IT ALL ZEROS \ NO ***
/ ? \ ***

```

```

***
**A*
***
I SUER2(55)
*****
**GO SET UP INFORMATION**
** TO BE PRINTED **
*****
I $ERROR(53)
*****
** GO REPORT "BCR" **
** DATA ERROR **
** EM10,DH2,DT2,0 **
*****
I
-----
*** YES / LOOP ON FAILING \ ***YES / LOOP ON FAILING \
**C*-----> LINE ' SRO9=1 **C*-----> LINE ' SRO9=1 \
***
I NO
***
**D*
***
I NO

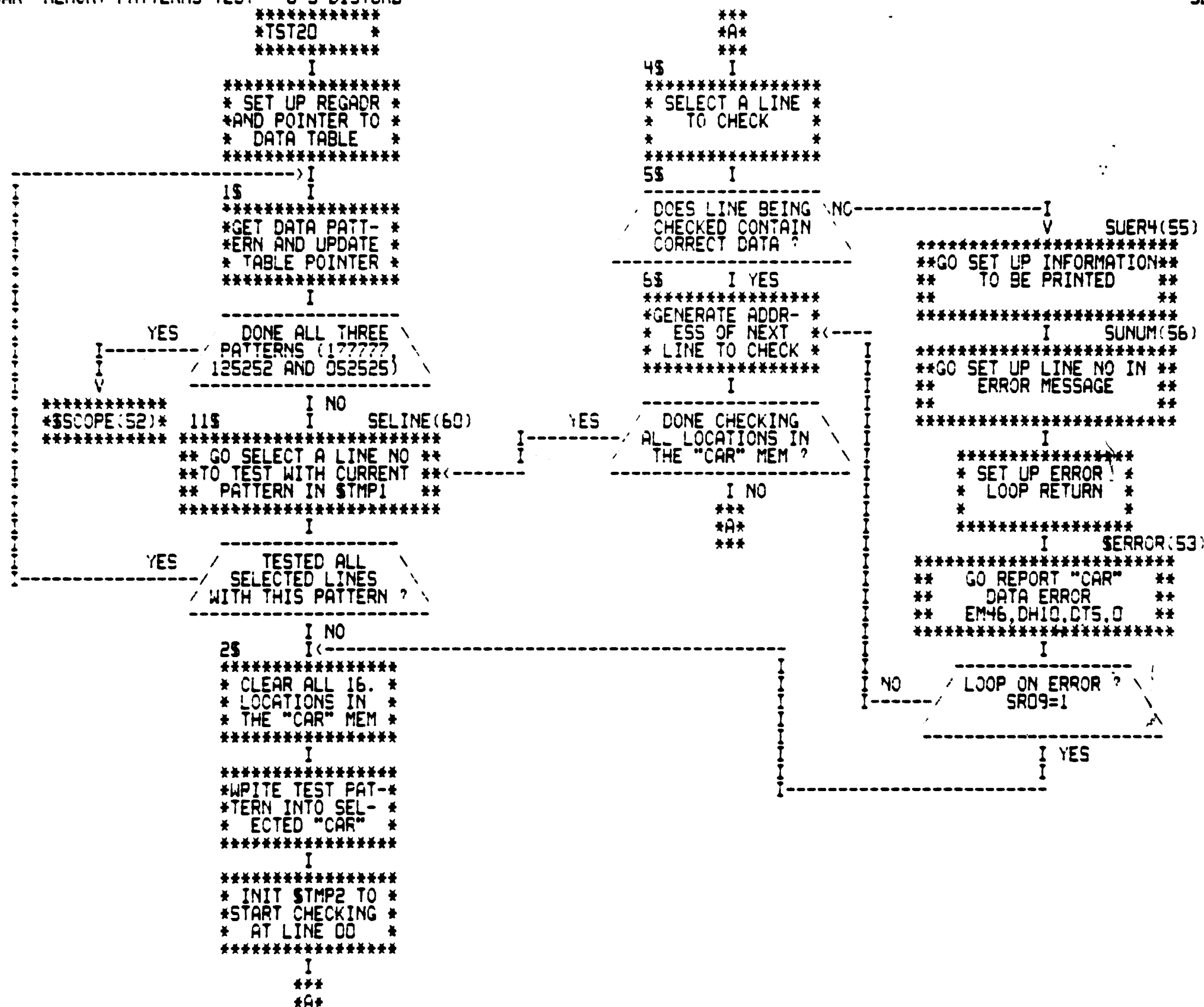
```

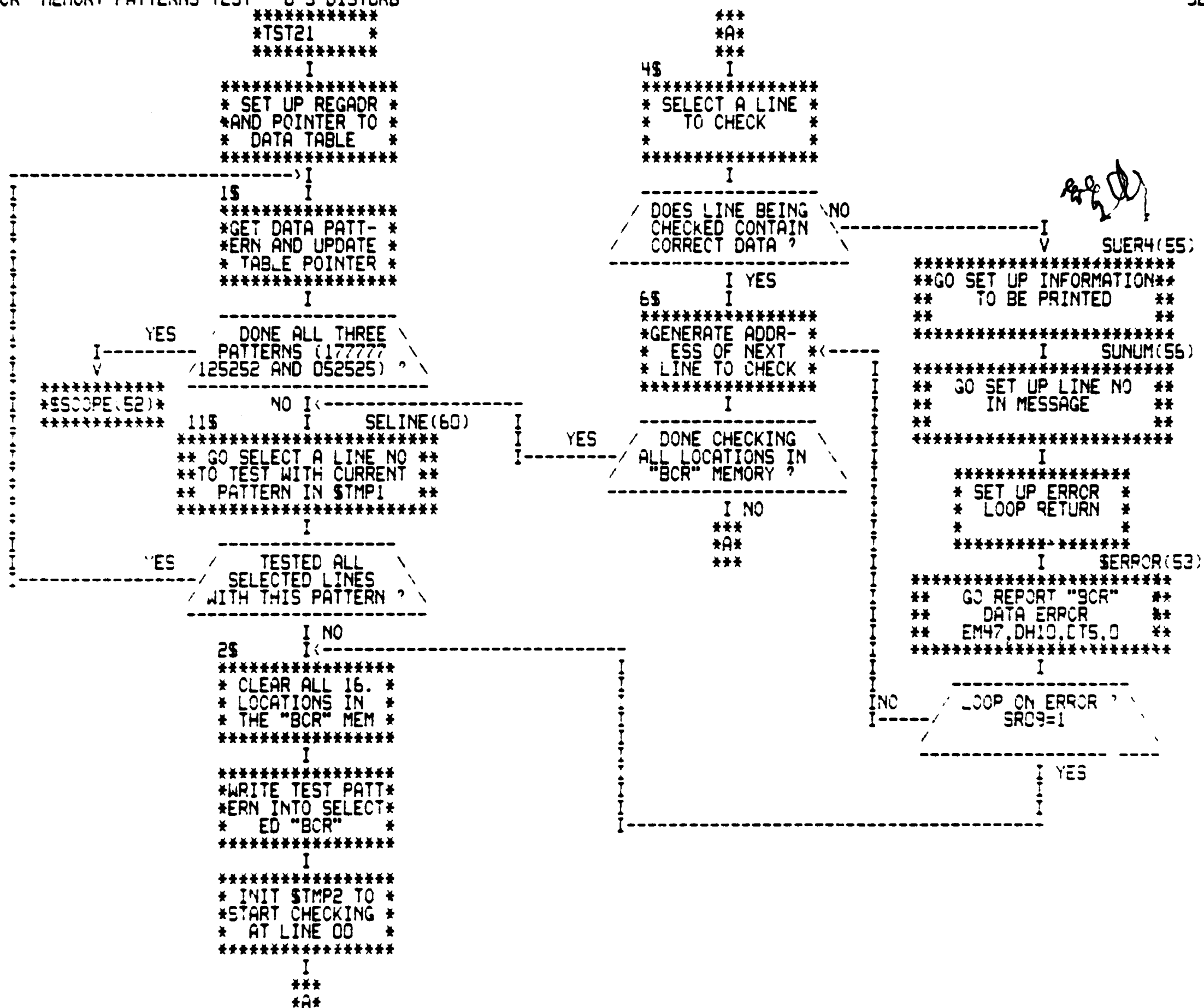

F06

MD-11-DZDMM-8 DH11 DIAGNOSTIC
TEST 20 "CAR" MEMORY PATTERNS TEST - O'S DISTURB

DECFL0 VER 00.12 23-SEP-76 14:43 PAGE 18

SEQ 0070






```

*****
* TST22 *
*****
I
*****
* SEI UP REGADR *
* AND 177777 TEST *
* PATTERN IN R5 *
*****
1$ I SELINE(60)
*****
** GO SELECT A LINE NO **
** TO TEST **
*****
I
YES / TESTED ALL
I / SELECTED LINES ?
I /
I NO
*****
** SCOPE(52) **
*****
2$ I
*****
* WRITE ALL ONES *
* INTO ALL 16. *
* "CAR" LOCATIONS *
*****
I
*****
* ZERO THE SELECT *
* ED "CAR" LOCAT- *
* ION *
*****
I
*****
* INIT $TMP2 TO *
* START CHECKING *
* AT LOC. 00 *
*****
4$ I
*****
* READ BACK A *
* "CAR" LOCATION *
* *
*****
5$ I
*****
/ DID IT CONTAIN \ NO ***
/ THE CORRECT DATA \ ----> *B*
/
I YES
***
*A*

```

```

***
*A*
***
6$ I
*****
* GENERATE NEXT *
* "CAR" ADDRESS *
* TO CHECK *
*****
I
YES / CHECKED ALL 16.
I / "CAR" MEMORY
I / LOCATIONS ?
I NO

```

```

***
*B*
***
I SUER4(55)
*****
** GO SET UP INFORMATION **
** TO BE PRINTED **
**
*****
I SUNUM(56)
*****
** GO SET JP LINE NO. **
** IN ERROR MESSAGE **
**
*****
I
*****
* SET UP ERROR *
* LOOP RETURN *
*
*****
I $ERROR(53)
*****
** GO REPORT "CAR" **
** DATA ERROR **
** EM46,DH10,DTS,0 **
*****
I
NO / LOOP ON ERROR
I / SRO9=1
I YES
***
*C*
***

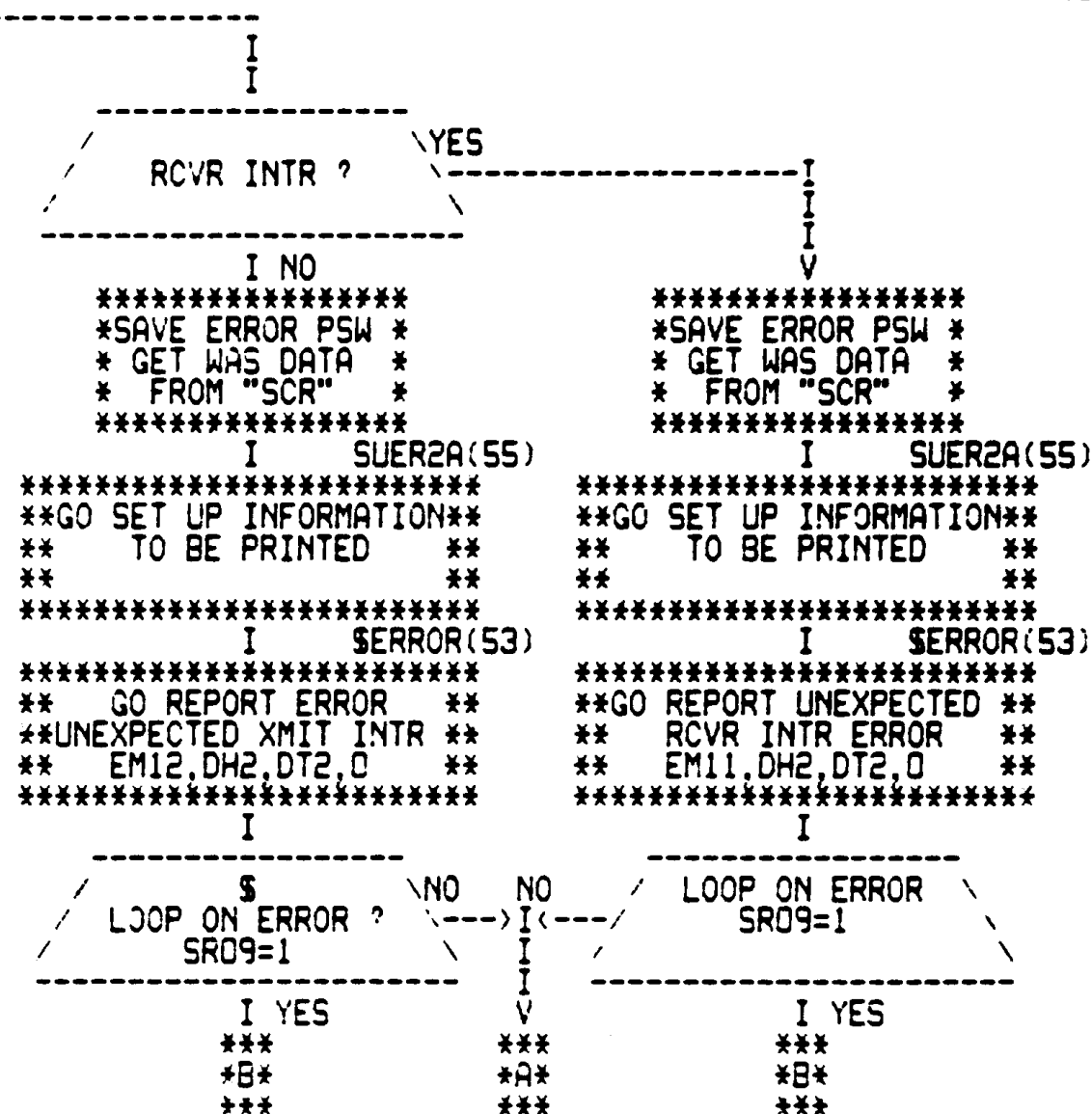
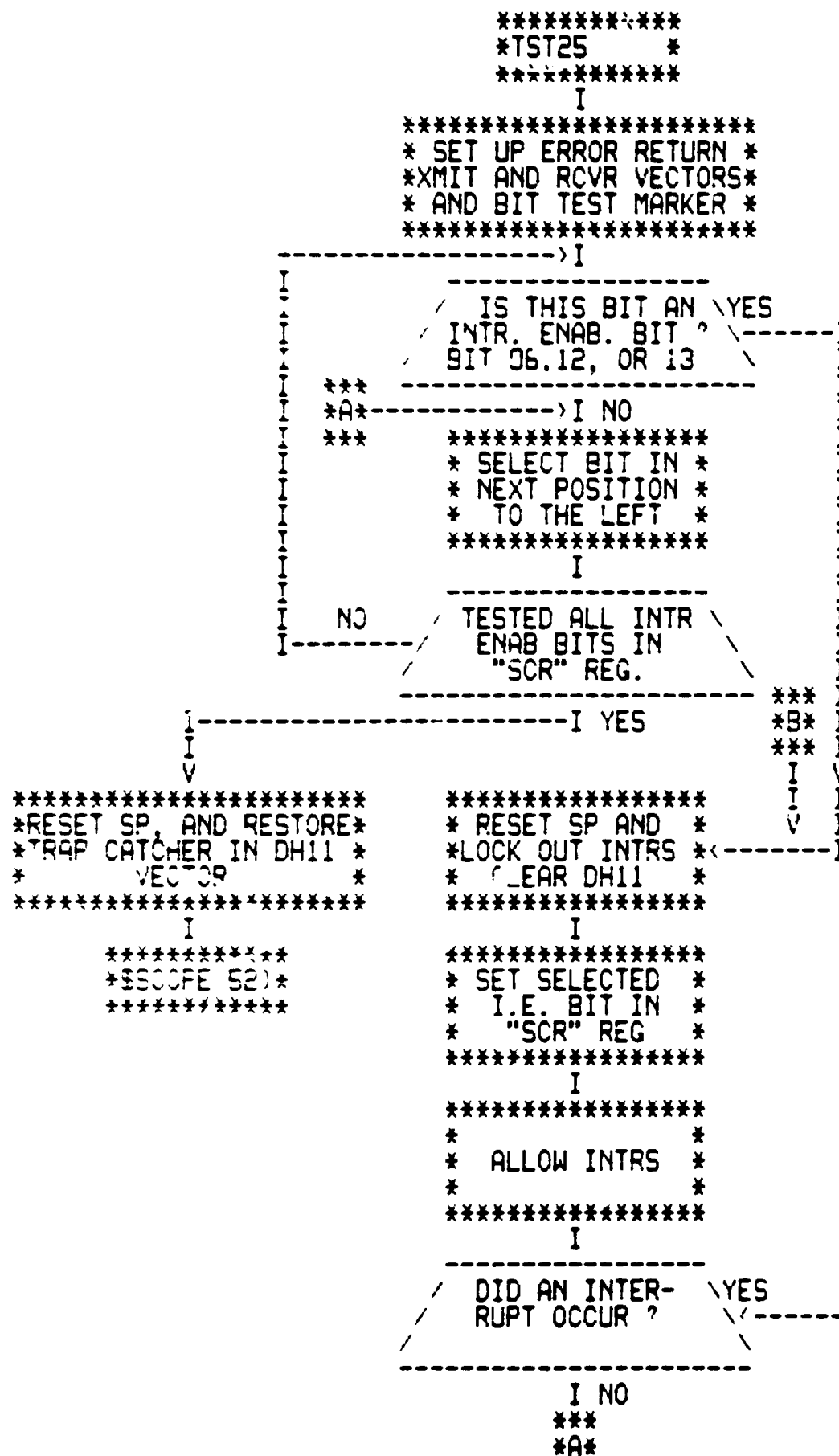
```

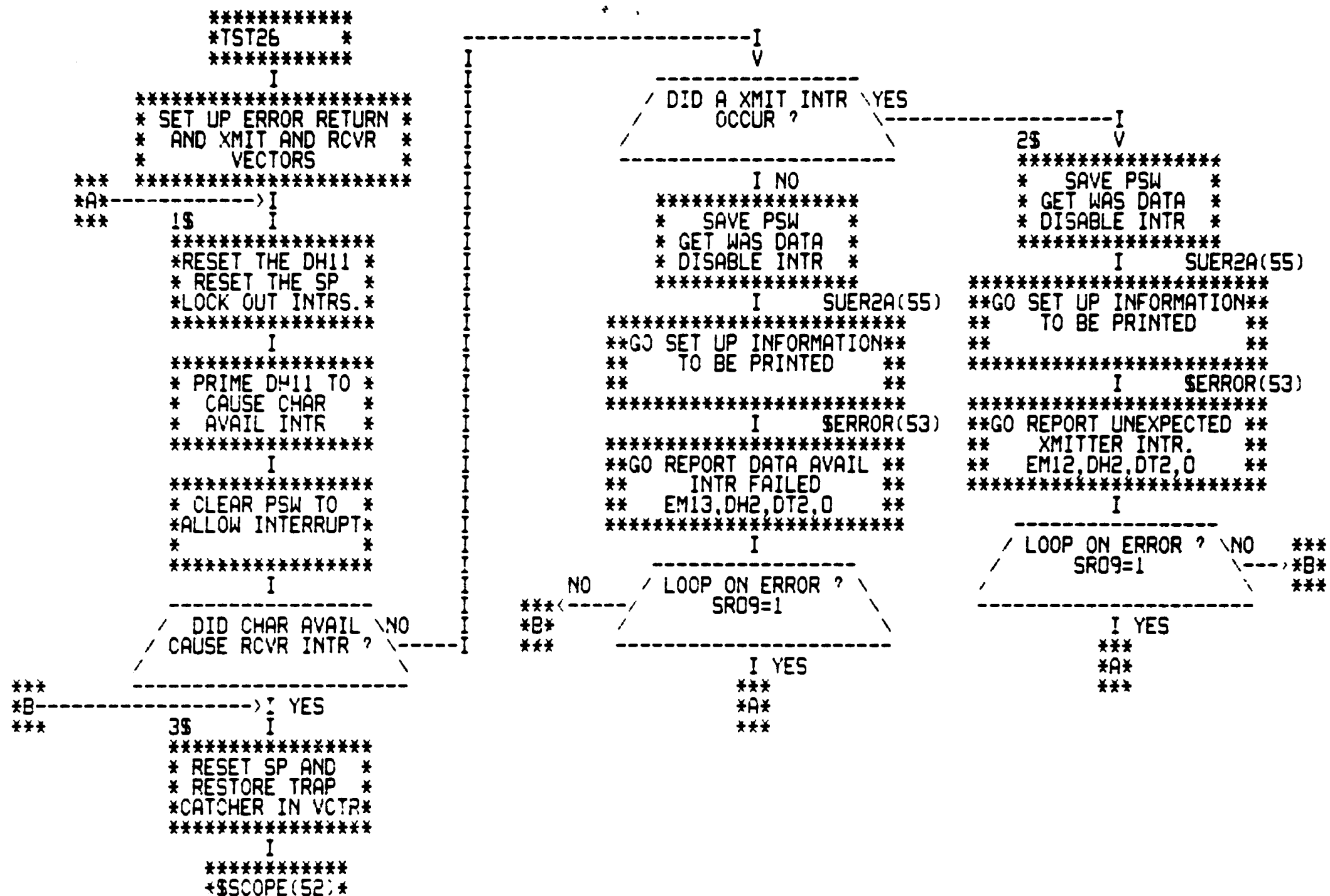
```

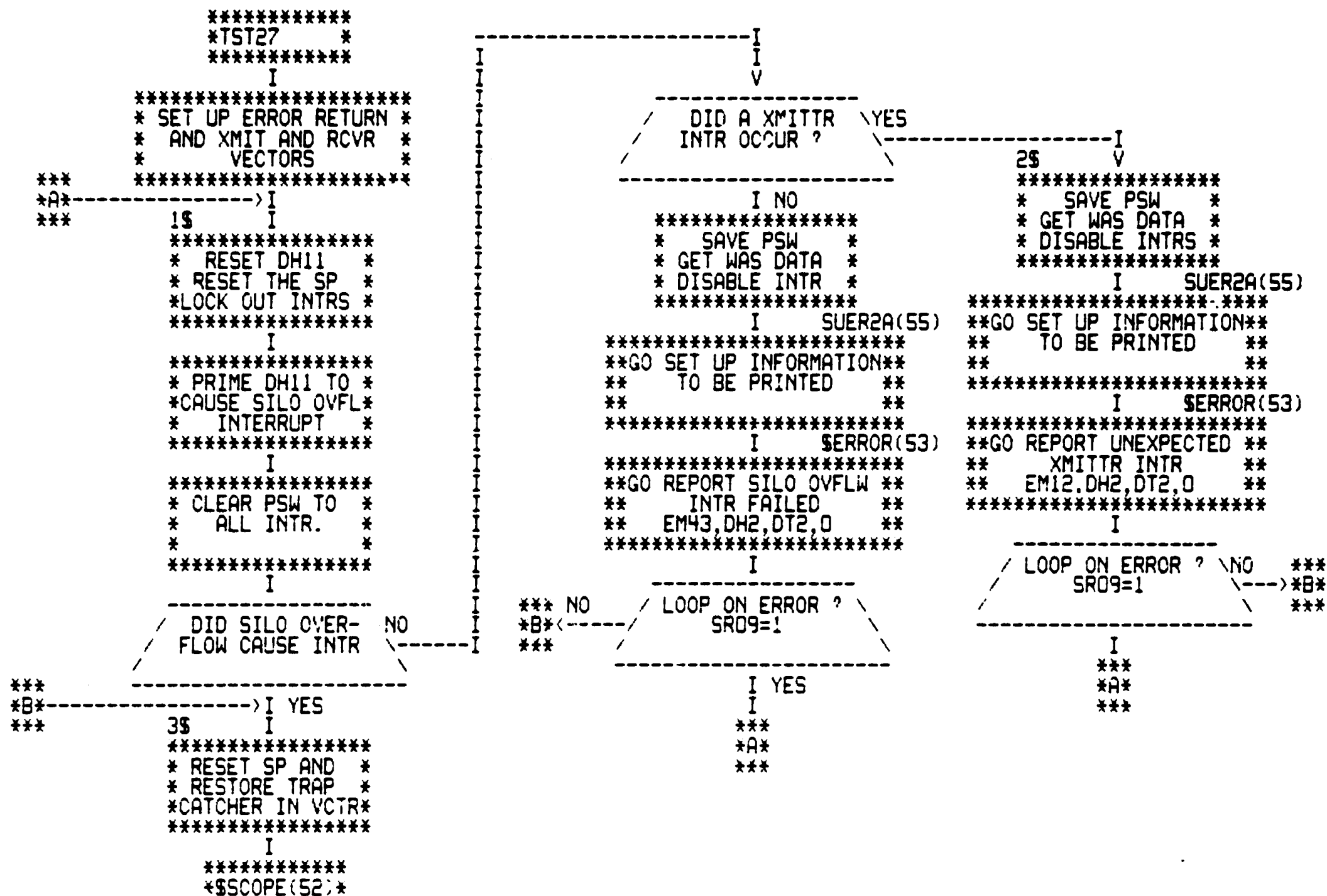
*****
*TST23 *
*****
I
*****
* SET UP REGADR *
*AND 177777 TEST*
* PATTERN IN R5 *
*****
I<-----
1$ I SELINE(60) I
*****
**GO SELECT A LINE NO. **
** TO TEST **
** **
*****
I
-----
YES / TESTED ALL \
I / SELECTED LINES ? \
I
*****
2$ I NO
*****
**WRITE ALL ONES *
**INTO ALL "BCR" *C*
**MEMORY LOCATION*
*****
I
*****
*ZERO THE SELECT*
* ED "BCR" *
* LOCATION *
*****
I
*****
* INIT $TMP2 TO *
*START CHECKING *
* AT LOC 00 *
*****
4$ I<-----
*****
* READ BACK A *
*"BCR" LOCATION *
*
*****
5$ I
-----
/ DID IT CONTAIN \ NO
/ THE CORRECT DATA \
-----> *B*
*****
I YES
***
*A*

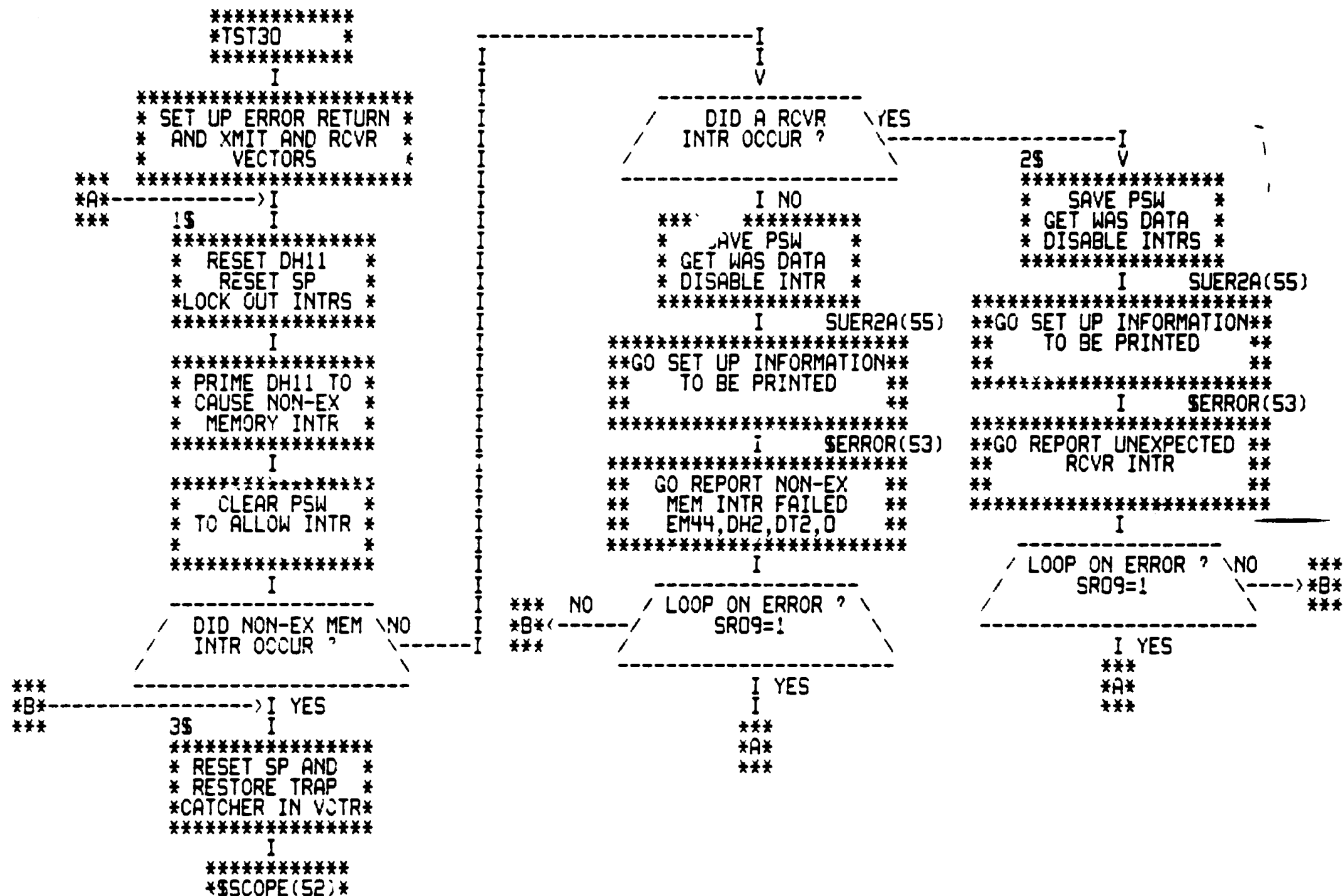
```

[illegible]







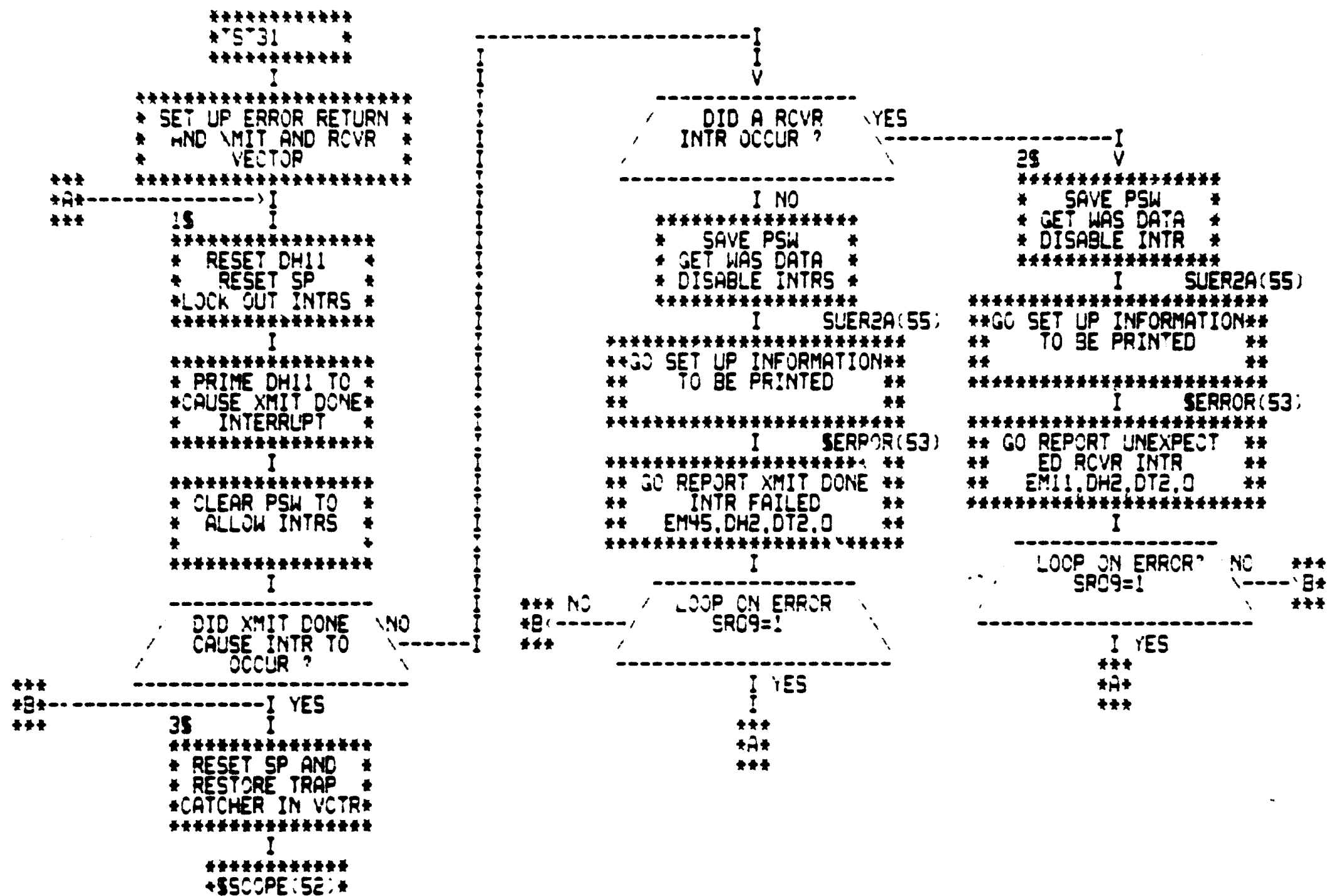


B07

NO-11-DIAGN-8 DH11 DIAGNOSTIC
TEST 31 TEST TRANSMIT DONE I.E. WITH INTR. CONDITION ACTIVE

DECFL0 VER 00.12 23-SEP-76 14:43 PAGE 27

SEQ 0079



1


```

*****
* TST32 *
*****
I
*****
* SET UP ERROR RETURN *
* AND XMIT VECTOR *
*****
***
*****
I
3S I TIMEIT(59)
*****
** CALL TIMER **
*****
I
DID TIMEOUT
OCCUR BEFORE INTR
ON SELECTED LINE
I YES
*****
* SAVE ERROR PSW *
* GET WAS DATA *
*****
I SUER2A(55)
*****
** GO SET UP INFORMATION **
** TO BE PRINTED **
*****
I SUNUM(56)
*****
** GO SET UP LINE NO. **
** IN ERROR MESSAGE **
*****
I $ERROR(53)
*****
** GO REPORT TIMEOUT **
** WAITING FOR XMIT DONE **
** EM15,DH2,DT2,0 **
*****
I
LOOP ON ERROR ? YES ***
SRC9=1 ***
I NO
***
I 9S I
***
** CLEAR DH11, RESTORE *
** TRAP CATCHER IN VEC *
** TOR AND CLR PSW *
*****
I
*****
* $SCOPE 52 *

```

```

NOTE: ***
COME HERE *D*
ON XMIT ***
INTERUPT I
-----
DID XMIT DONE \ NO
GET SET OK -----
I
V
I YES
5S I
*****
* NPR1A(29) *
*****
BUFFER ACTIVE \ NO
BIT CLEARED FOR -----
SELECTED LINE ? I
V
I YES
6S I
*****
* NPR2A(29) *
*****
"CAR" REG FOR \ NO
SELECTED LINE -----
GOT INCREMENTED? I
V
I YES
7S I
*****
* NPR3A(29) *
*****
"BCR" REG FOR \ NO
SELECTED LINE -----
GOT DECREMENTED ? I
V
I YES
***
**
***

```

```

*****
* NPR5A,29 *
*****
I
***
**
***

```

```

*****
* NPR5A,29 *
*****
I
***
**
***

```

D07

MC-11-DZDMM-B DH11 DIAGNOSTIC
TEST 32 BASIC "NPR" LOGIC TEST 1 (CONT.)

DECFL0 VER 00.12 23-SEP-76 14:43 PAGE 29

SEQ 0081

*NPR1A(28) *

I

*SAVE ERROR PSW *
* GET WAS DATA *
*

I SUER2A(55)

GO SET UP INFORMATION
** TO BE PRINTED **
**

I 95

** CALL SUBR TO SET UP **
** LINE NO. **
**

I \$ERROR(53)

** GO REPORT XMIT DONE **
** FAILED TO SET **
** EM14,DH2,DT2.0 **

I

LOOP ON ERROR ? \YES ***
SRC9=1 \--->*E*

I NO

*NPP5A(28) *

E

I

*NPP6A(28) *

*NPR2A(28) *

I

*SAVE ERROR PSW *
* GET WAS DATA *
*

I SUER2A(55)

GO SET UP INFORMATION
** TO BE PRINTED **
**

I 95

** CALL SUBRT TO SET UP **
** LINE NO. **
**

I \$ERROR(53)

** GO REPORT "BAR" **
** REG ERROR **
** EM14,DH2,DT2.0 **

I

LOOP ON ERROR ? \YES ***
SRC9=1 \--->*E*

I NO

*NPR3A(29) *

I

*SAVE ERROR PSW *
* GET WAS DATA *
*

I SUER2A(55)

GO SET UP INFORMATION
** TO BE PRINTED **
**

I 95

** CALL SUBR TO SET UP **
** LINE NO. **
**

I \$ERROR(53)

** GO REPORT "CAR" **
** REG ERROR **
** EM14,DH2,DT2.0 **

I

LOOP ON ERROR ? \YES ***
SRC9=1 \--->*E*

I NO

*NPP4A(28) *

I

*SAVE ERROR PSW *
* GET WAS DATA *
*

I SUER2A(55)

GO SET UP INFORMATION
** TO BE PRINTED **
**

I 95

** CALL SUBR TO SET UP **
** LINE NO. **
**

I \$ERROR(53)

** GO REPORT "BCR" **
** REG ERROR **
** EM14,DH2,DT2.0 **

I

LOOP ON ERROR ? \YES ***
SRC9=1 \--->*E*

I NO

95 SUNUM(56)

** GO SET UP LINE NO. **
** IN ERROR MESSAGE **
**

```

*****
*TST33
*****
I
*****
* SET UP ERROR *
* LOOP RETURN *
*****
***
+2+-----I
*** 15 I SELINE(60)
*****
**GO SELECT A LINE NO. **
** TO TEST **
*****
I
-----
***YES / TESTED ALL \
**D* / SELECTED LINES \
***
***
+2+-----I NO
***
*****
* CLEAR OUT THE *
* DH11 *
*****
I CLC90(56)
*****
** GO CLEAR "BCR" **
** AND "CAR" MEMORIES **
*****
I LDBCR(56)
*****
** LOAD BYTE COUNT **
** MEMORY WITH ALL **
** ONES **
*****
I
*****
*ACTIVATE DH11 TO XMIT*
* ONE BYTE ON THE *
* SELECTED LINE *
*****
I
*****
* INIT TIMERS *
* "A" AND "B" *
*****
I
***
+3+
***

```

```

***
*B*
***
35 I
-----
/ IS XMIT DONE \ ***
/ SET YET ? \ ---E*
-----
I NO
I TIMEIT(59)
*****
** CALL TIMER **
*****
I
-----
/ TIMEOUT WHILE \ NO
/ WAITING FOR XMIT \
/ DONE ? \
-----
I YES
*****
* SAVE ERROR PSW *
* SET UP WAS DATA *
* SET UP S/B DATA *
*****
I SUER2A(55)
*****
**GO SET UP INFORMATION**
** TO BE PRINTED **
*****
I SUNUM(56)
*****
** GO PUT LINE NO. **
** IN ERROR MESSAGE **
*****
I $ERROR(53)
*****
** GO REPORT TIMEOUT **
** WAITING FOR XMIT DONE**
** EMS0,DH12,DT2,0 **
*****
I
-----
/ LOOP ON ERROR ? \ YES ***
/ SR09=1 \ ---C*
-----
I NO
***
+A*
***

```

```

***
+E*
***
I
-----
/ "BCR" FOR SEL- \ NO
/ ECTED LINE GOT \
/ INCREMENTED ? \
-----
I YES
-----
/ "BCR" FOR UN- \ NO
/ SELECTED LINES \
/ NOT INCREMENTED \
-----
I YES
-----
/ "CAR" FOR SEL- \ NO
/ ECTED LINE GOT \
/ INCREMENTED \
-----
I YES
-----
/ "CAR" FOR ALL \ NO
/ OTHER LINES NOT \
/ INCREMENTED \
-----
I YES
I
-----
/ LOOP ON ERROR ? \ NO ***
/ SR09=1 \ ---A*
-----
I YES
***
* SET UP REGADR *
***
*****
I SUER4(55)
*****
**GO SET UP INFORMATION**
** TO BE PRINTED **
** EMS1,DH11,DT5,0 **
*****
I $ERROR(53)
*****
** GO REPORT "CAR" **
** REG ERROR **
*****
I
-----
/ LOOP ON ERROR ? \ NO ***
/ SR09=1 \ ---A*
-----
I YES
***
+C*
***

```

```

*****
* SET UP REGADR *
*****
I SUER4(55)
*****
**GO SET UP INFORMATION**
** TO BE PRINTED **
*****
I $ERROR(53)
*****
** GO REPORT "BCR" **
** REG ERROR **
** EMS1,DH11,DT5,0 **
*****
I
-----
/ LOOP ON ERROR ? \ NO ***
/ SR09=1 \ ---A*
-----
I YES
***
* SET UP REGADR *
***
*****
I SUER4(55)
*****
**GO SET UP INFORMATION**
** TO BE PRINTED **
** EMS1,DH11,DT5,0 **
*****
I $ERROR(53)
*****
** GO REPORT "CAR" **
** REG ERROR **
*****
I
-----
/ LOOP ON ERROR ? \ NO ***
/ SR09=1 \ ---A*
-----
I YES
***
+C*
***

```

```

***
*D*
***
I
*****
*$SCOPE(52)*
*****

```

```

*****
TST34
*****
I
*****
* SET UP ERROR *
* RETURN AND RCVR *
* VECTOR *
*****
***
***
I
*****
* LOOK OUT INTR *
* RESET SP AND *
* CLEAR DH11 *
*****
I
*****
* SET SILO MAINT TO *
* LOAD SILO AND ENAB *
* CHAR AVAIL INTR *
*****
I
*****
* CLEAR THE PSW *
* TO ALLOW INTR *
*
*****
2$ I
-----
/ DID TIMEOUT \ YES
/ OCCUR WHILE WAIT- \
/ ING FOR INTR. \
-----
3$ I NO
-----
/ DID SILO CONTAIN \ NO
/ TEST PATTERN \
/ 125252 ? \
-----
4$ I YES
-----
/ DID SILO FILL \ NO
/ COUNT = 1? \
-----
5$ I YES
*****
***
* RESET SP, RESTORE *
* TRAP CATCHER IN VECT *
***
* OR, CLR PSW *
*****
I
*****
* $SCOPE(52)*
*****

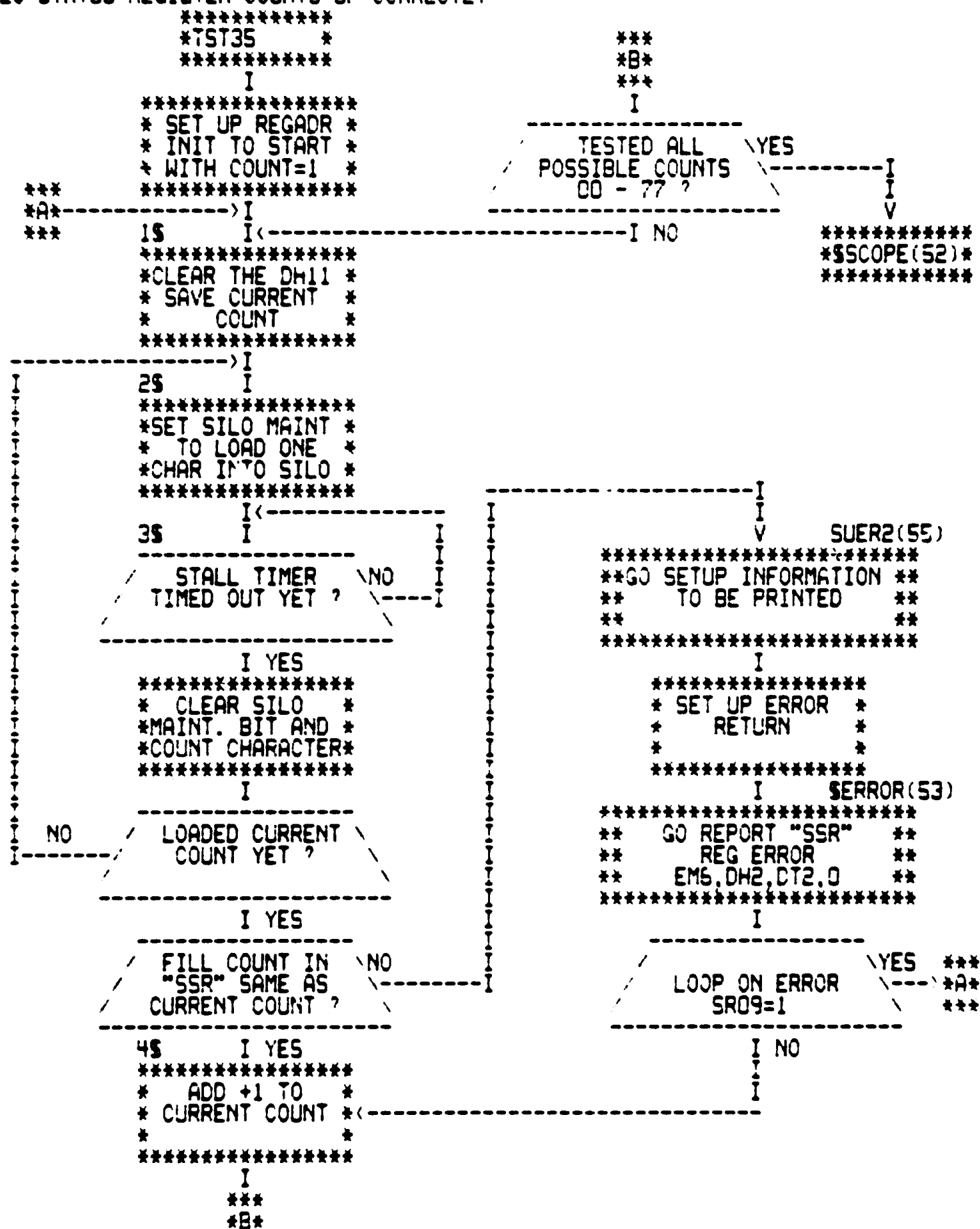
```

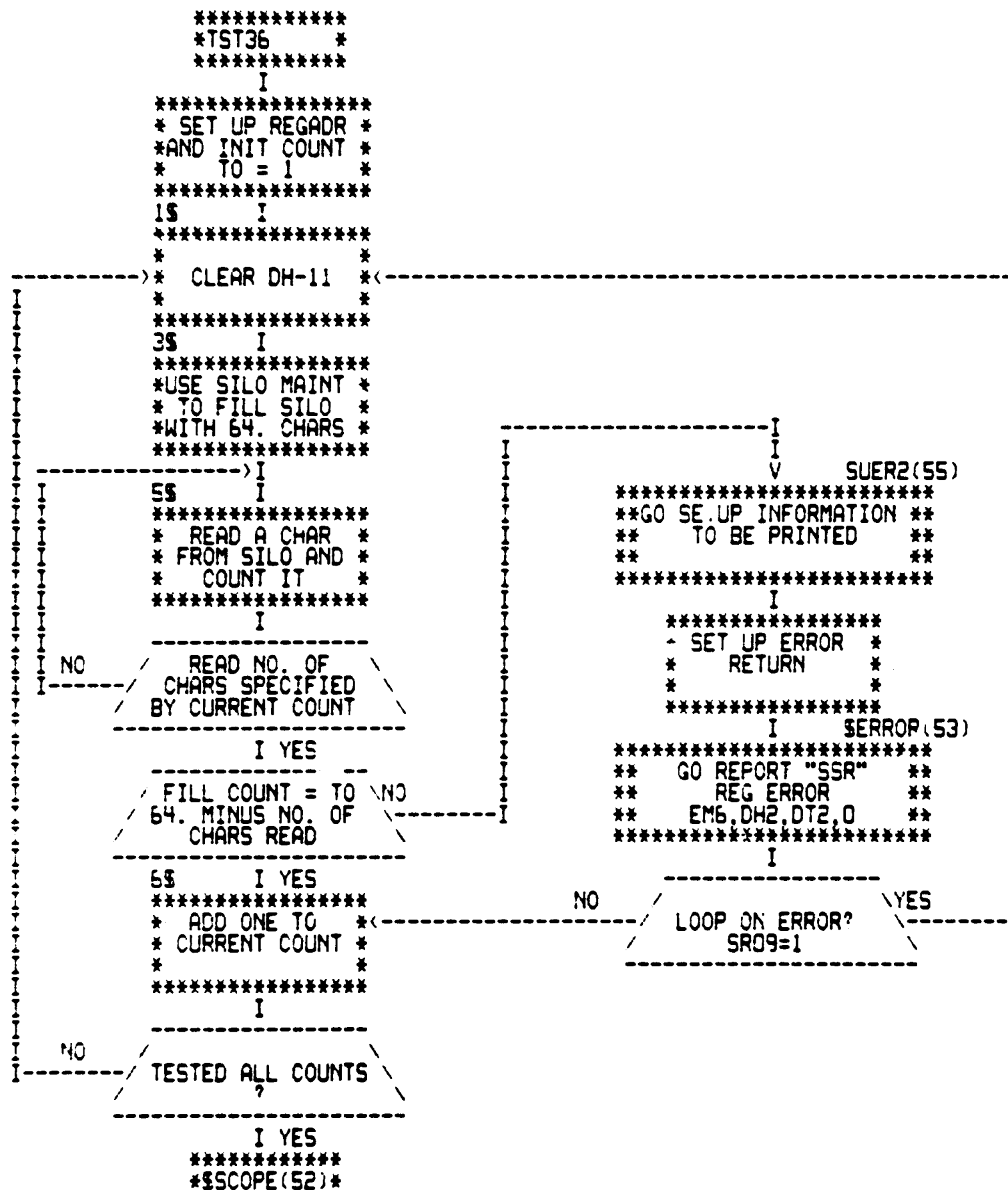
G07

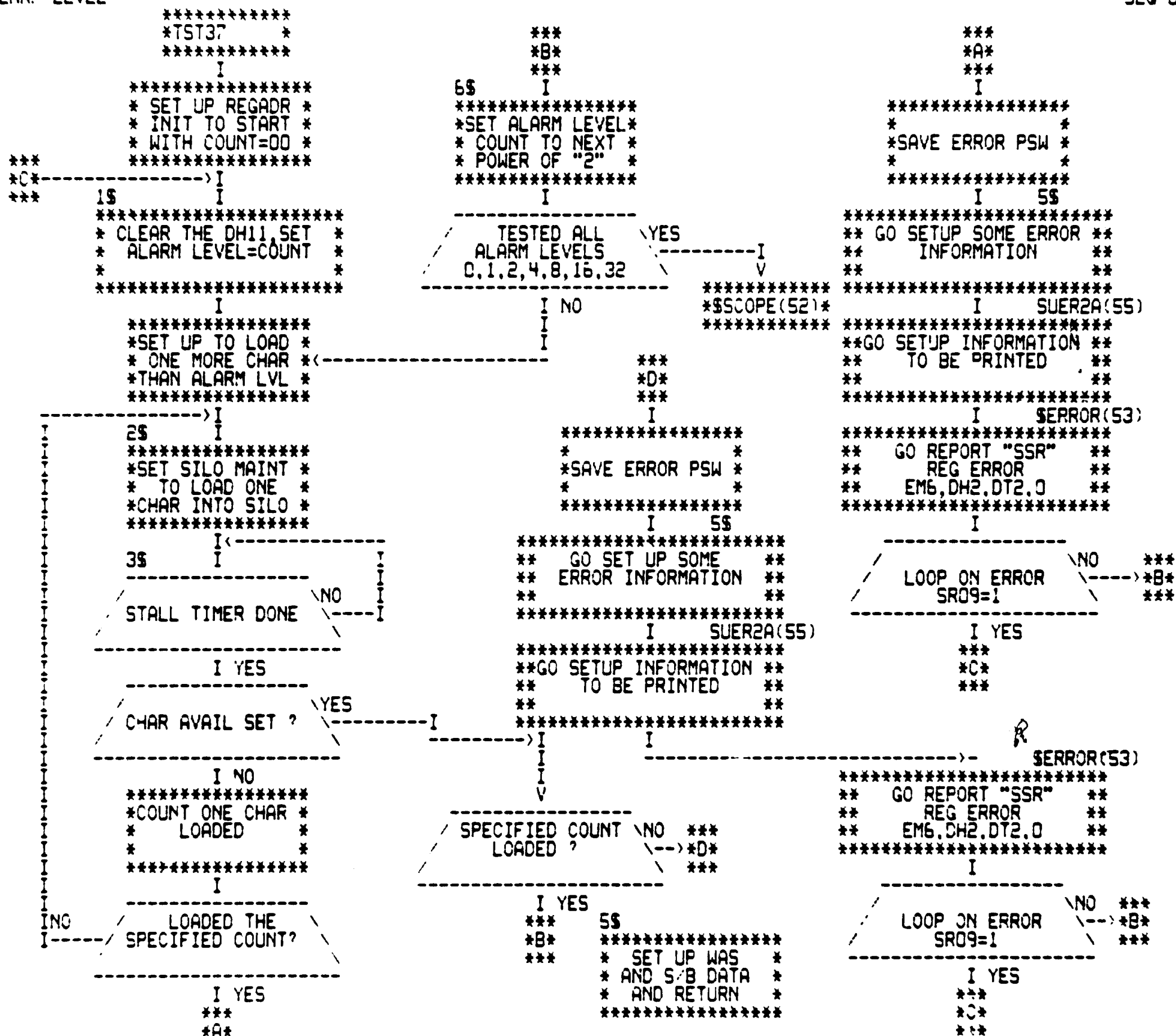
MD-11-DZDMM-B DH11 DIAGNOSTIC
TEST 35 TEST THAT SILO STATUS REGISTER COUNTS UP CORRECTLY

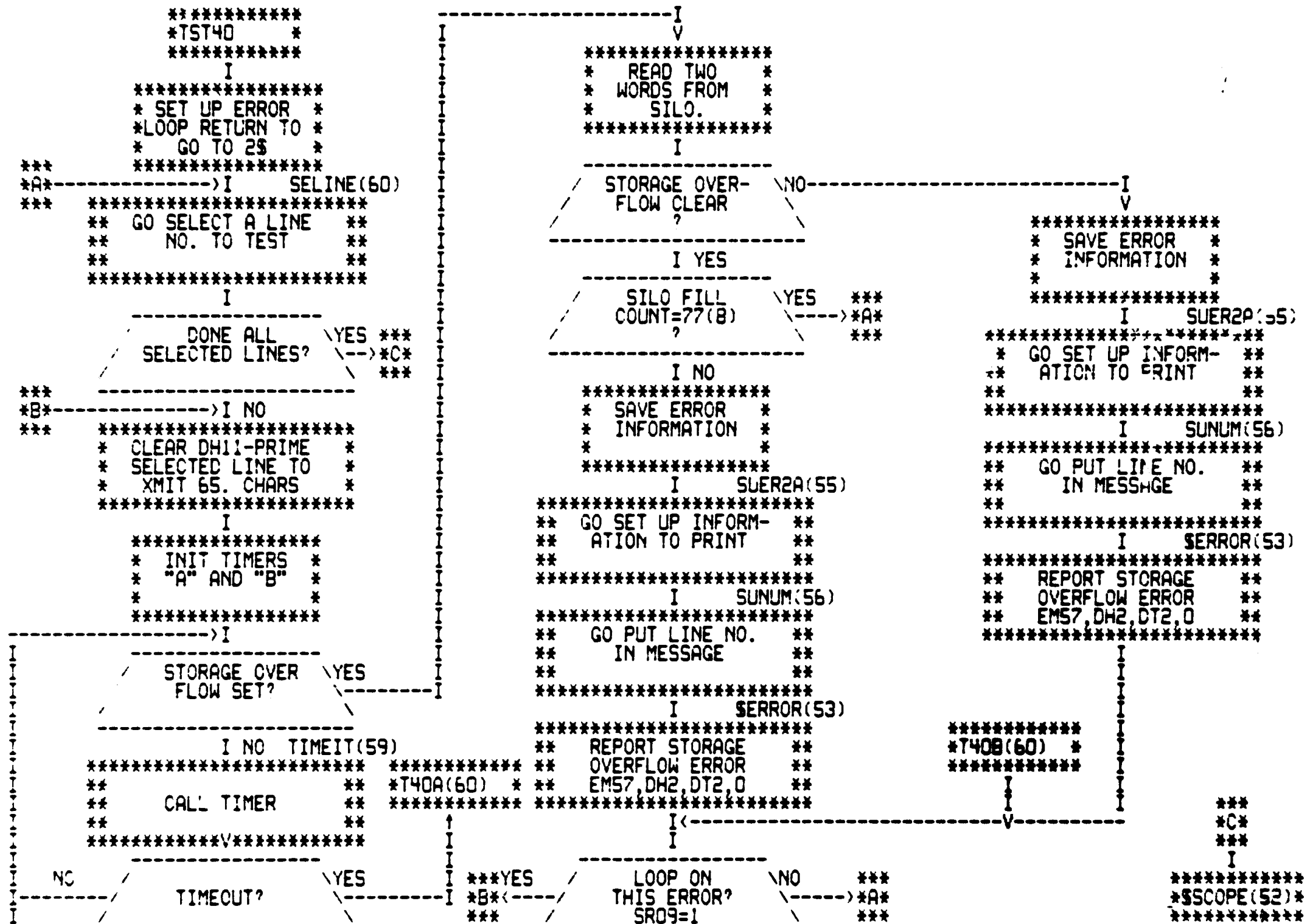
DECFLO VER 00.12 23-SEP-76 14:43 PAGE 32

SEQ 0084









[illegible]

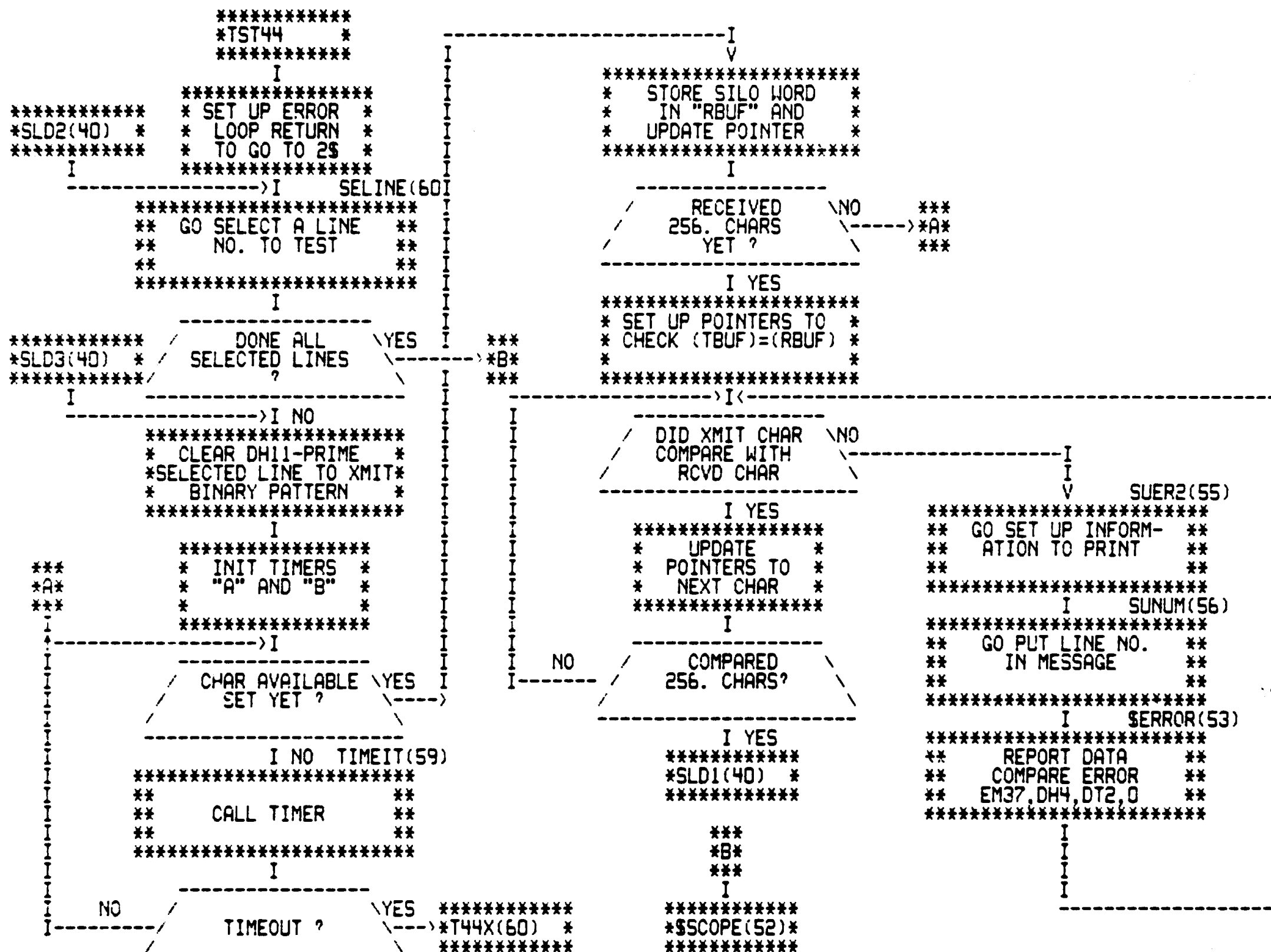
```
*****
*TST42 *
*****

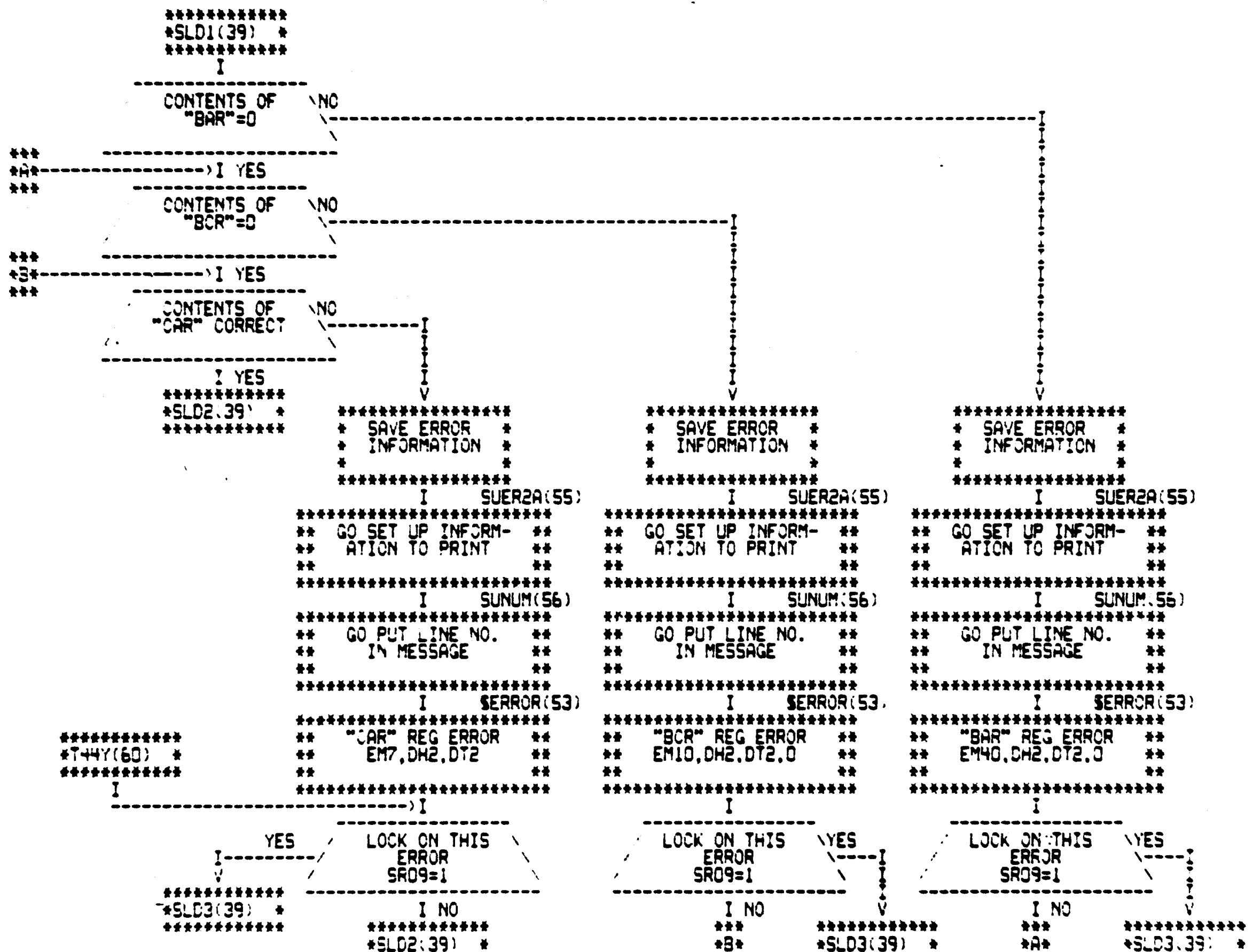
I
*****
* SET UP ERROR *
* LOOP RETURN *
*
***
*****
*A*-----I
*** 1$ I SELINE(60)
*****
** GO SELECT A LINE **
** NO TO TEST **
**
*****
I
-----
/ DONE TESTING \ YES
/ ALL SELECTED \
/ LINES ? \
-----
I NO
*****
** INIT TO START AT LOW- **
** EST SPEED **
** INIT "TIMEC" = -1 **
***
*****
*B*-----I
*** 2$ I
*****
* CLEAR THE DH11 *
*
*****
I
*****
* ACTIVATE DH11 TO *
* XMIT AND RECEIVE *
* ON CHAR *
*****
I
*****
* INIT TIMERS *
* "A" AND "B" *
*
*****
4$ I
-----
/ CHAR AVAILABLE \ YES ***
/ SET YET ? \ ---->*D*
-----
I NO
***
*C*

***
*C*
***
I TIMEIT(59)
*****
** CALL TIMER **
**
*****
I
-----
/ DID TIMEOUT \ YES
/ OCCUR BEFORE \
/ CHAR AVAIL SET \
-----
I NO
*****
* SAVE ERROR PSW, SET *
* UP SPEED CODE, WAS *
* AND S/B DATA *
*****
I SUER2A(55)
*****
** GO SETUP INFORMATION **
** TO BE PRINTED **
*****
I SUNUM(56)
*****
** GO SET UP LINE NO. **
** IN ERROR MESSAGE **
*****
I $ERROR(53)
*****
** GO REPORT TIMEOUT **
** WAITING FOR DONE **
** EM22,DH2,DT2,0 **
*****
I
-----
/ LOOP ON ERROR ? \ NO
/ SR09=1 \
-----
I YES
*****
I YES
*****
I NO
***
*E*
***

*****
I
-----
/ TESTED ALL \ YES ***
/ SPEEDS ON \ ---->*A*
/ SELECTED LINES ? \
-----
***
```

[illegible]





```

*****
* TST45 *
*****
I
*****
* SET UP ERROR *
* LOOP RETURN *
*****
***
* A-----I
*** 15 I SELINE(60)
*****
**GO SELECT A LINE NO. **
** TO TEST **
*****
I
-----
/ TESTED ALL SEL- YES
/ ECTED LINES ? \-----
***
* D-----I NO
*** 25 I *****
***** $SCOPE(52) *****
* ACTIVATE DM11 TO XFER *
* ONE CHAR WITH "ODD" *
* PARITY SELECTED *
*****
I
*****
* INIT TIMERS *
* "A" AND "B" *
*****
I
-----
35 I
-----
CHAR AVAIL SET YES ***
YES ? \-----C*
***
I NO TIMEIT(59)
*****
** CALL TIMER **
*****
I
-----
/ CHAR AVAILABLE
/ TIMEOUT ? \
I YES
***
* B*

```

```

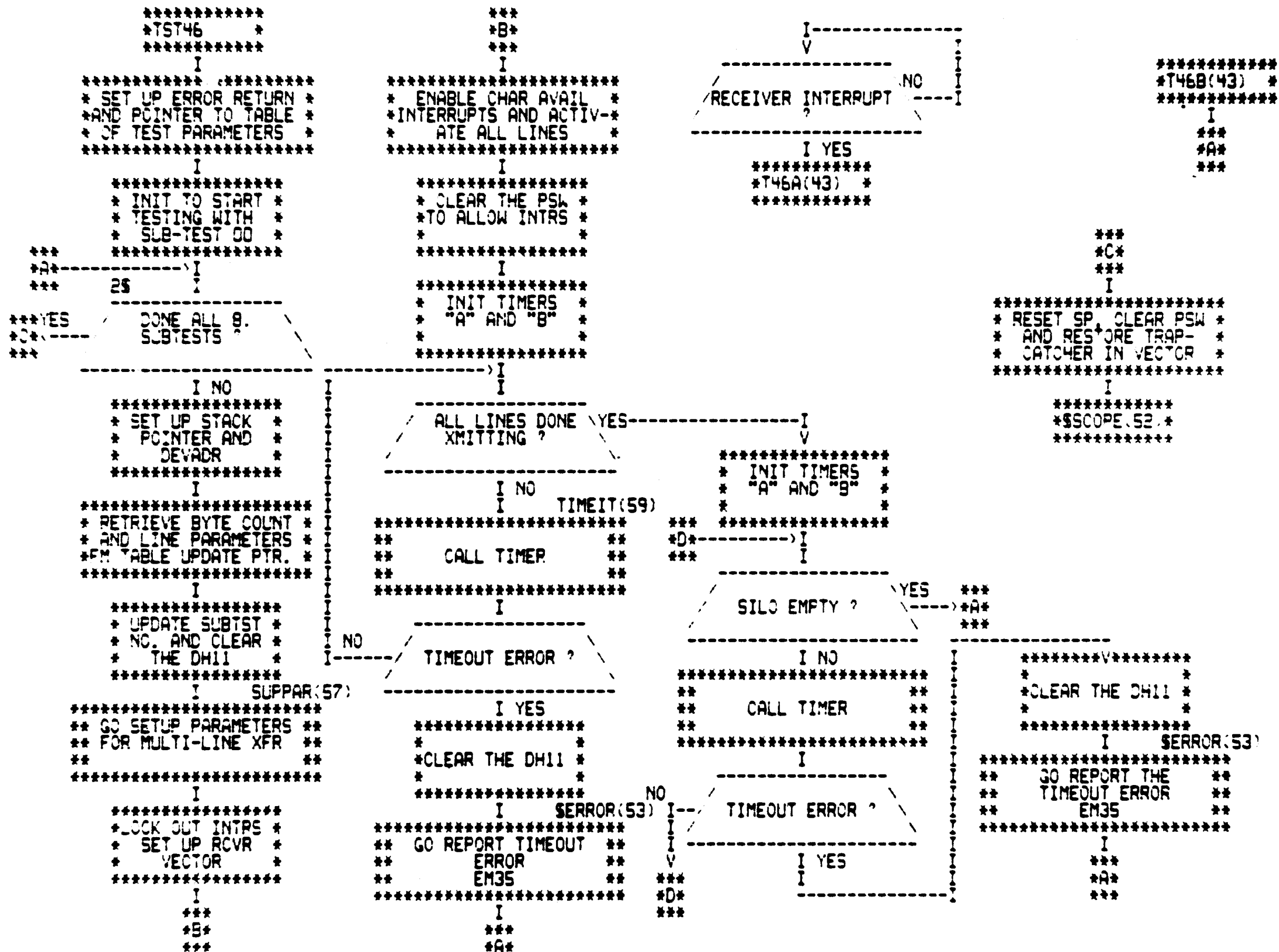
***
* B*
***
I
*****
* SAVE ERROR PSW SET *
* UP WAS, S B DATA AND *
* REGADR *
*****
I SUER2A(55)
*****
**GO SETUP INFORMATION **
** TO BE PRINTED **
*****
I SUNUM(56)
*****
** GO SET UP LINE NO. **
** IN ERROR MESSAGE **
*****
I SERROR(53)
*****
** GO REPORT DATA **
** AVAILABLE TIMEOUT **
** EM22,DM2,DT2,0 **
*****
I
-----
*** YES
* D<-----/ LOOP ON ERROR \
*** / SRO9=1 \
I NO
***
* A*
***

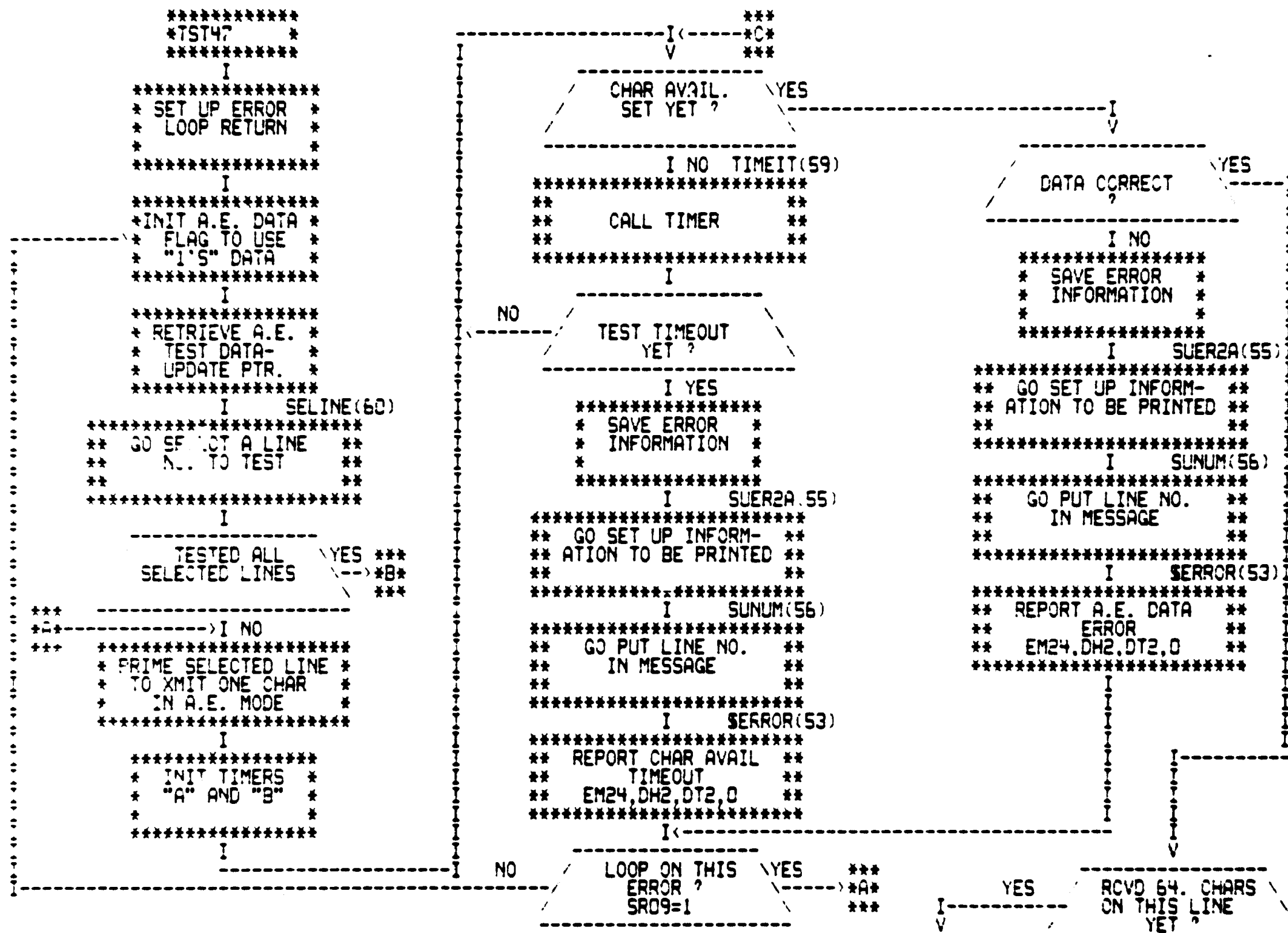
```

```

***
* C*
***
45 I
-----
/ CORRECT DATA YES ***
/ RECEIVED ? \-----A*
***
I NO
*****
* SAVE ERROR PSW *
* SET UP REGADR *
*****
I SUER2A(55)
*****
**GO SETUP INFORMATION **
** TO BE PRINTED **
*****
I SUNUM(56)
*****
** GO SET UP LINE NO. **
** IN ERROR MESSAGE **
** EM33,DM2,DT2,0 **
*****
I SERROR(53)
*****
** GO REPORT PARITY **
** DATA ERROR **
*****
I
-----
*** YES
* D<-----/ LOOP ON ERROR ? \
*** / SRO9=1 \
I NO
***
* A*
***

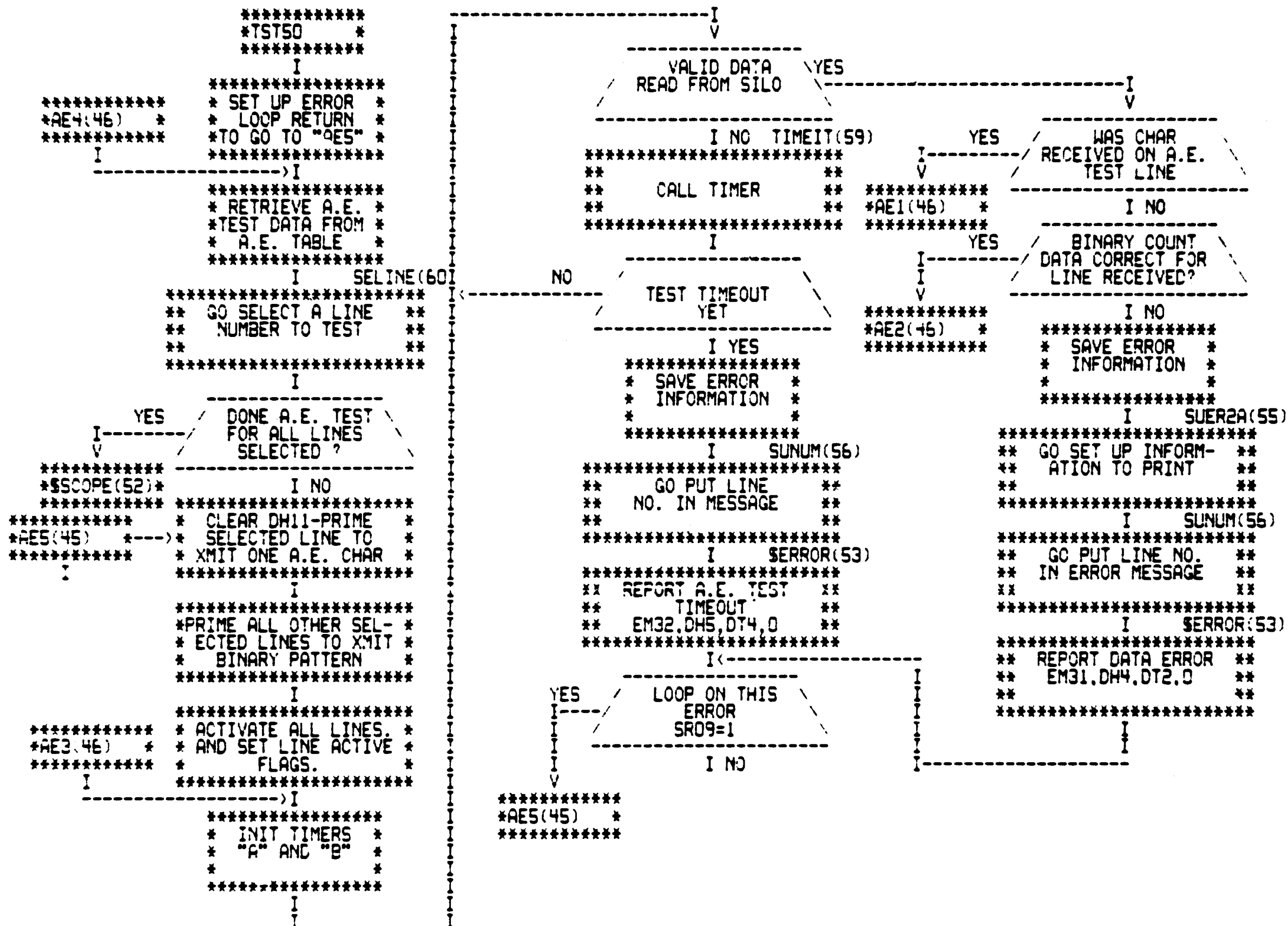
```

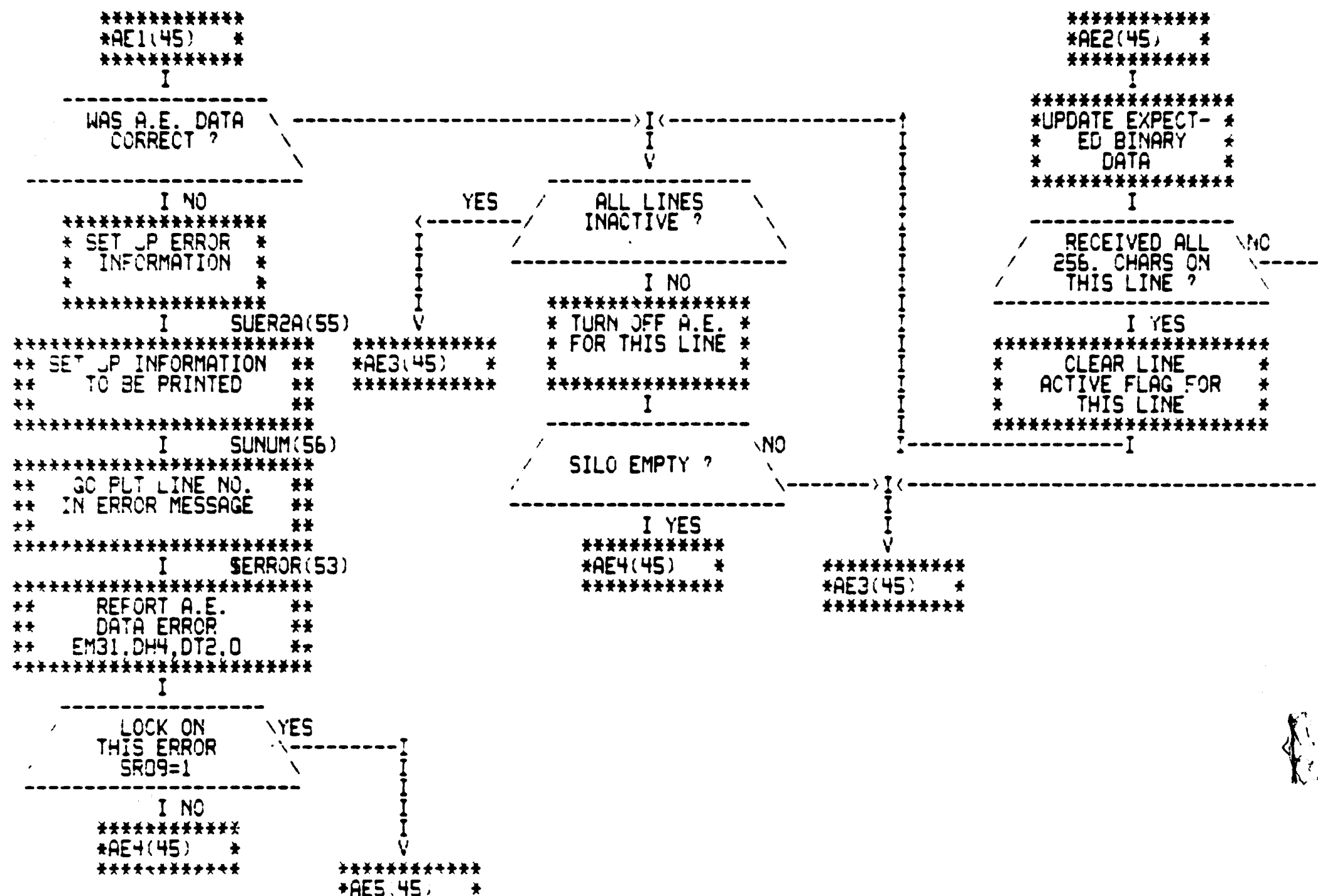





B
*** NOTE: THIS TEST REPEATED FOR
I BOTH ALL (1'S) AND ALL (0'S) TEST
***** DATA - THEN EXITS TO NEXT TEST
SCOPE(52) VIA "SCOPE"

* DISABLE AUTO * I NO
* ECHO MODE * ***
* ----->*C*





```

*****
*TST51 *
*****
I
*****
* SET UP ERROR *
* LOOP RETURN *
* *****
***>I
***>I
*****
** CLEAR THE SEL- **
** ECTED DH11 **
** *****
I
*****
* SET UP XMIT ONE *
* CHAR IN A.E. MODE ON *
* ALL SELECTED LINES *
* *****
I
*****
* ACTIVATE ALL SELECT- *
* ED LINES-SET ACT- *
* IVE FLAGS *
* *****
I
*****
* INIT TIMERS *
* "A" AND "B" *
* *****
>I
----->I
/ \
CHAR AVAILABLE / YES
SET YET ? \
-----
I NO TIMEIT(59)
*****
** CALL TIMER **
** *****
I
-----
NO / \ YES
TIMEOUT ? \
-----

```

I
V

```

*****
* TURN OFF A.E. *
* BIT FOR THIS *
* LINE *
* *****
I
*****
* CLEAR LINE *
* ACTIVE FLAG *
* FOR THIS LINE *
* *****
I
ALL LINES / YES
INACTIVE ? \
-----
I NO
***
*A*
***
I
V
*****
* SAVE ERROR *
* INFORMATION *
* *****
I $ERROR(53)
** REPORT DATA AVAIL- **
** ABLE TIMEOUT **
** EM36,DH5,DT4,0 **
*****

```

```

CORRECT DATA / NO
FOR THIS LINE? \
-----
I YES
AUTO ECHOED /
64. CHARS ON \
THIS LINE -----
I YES
*****
* TURN OFF A.E. *
* BIT FOR THIS *
* LINE *
* *****
I
*****
* CLEAR LINE *
* ACTIVE FLAG *
* FOR THIS LINE *
* *****
I
ALL LINES / YES
INACTIVE ? \
-----
I NO
***
*A*
***
I
V
*****
* SAVE ERROR *
* INFORMATION *
* *****
I $ERROR(53)
** REPORT DATA AVAIL- **
** ABLE TIMEOUT **
** EM36,DH5,DT4,0 **
*****

```

```

*****
* SAVE EPROR *
* INFORMATION *
* *****
I SUER2A(55)
** GO SET UP INFORMATION **
** TO BE PRINTED **
** *****
I SUNUM(56)
** GO PUT LINE NO. **
** IN MESSAGE **
** *****
I $ERROR(53)
** REPORT AUTO ECHO **
** DATA ERROR **
** EM31,DH4,DT2,0 **
*****
I<-----
/ \
LOOP ON THIS /
ERROR ? \
-----
I YES
***
*B*
***

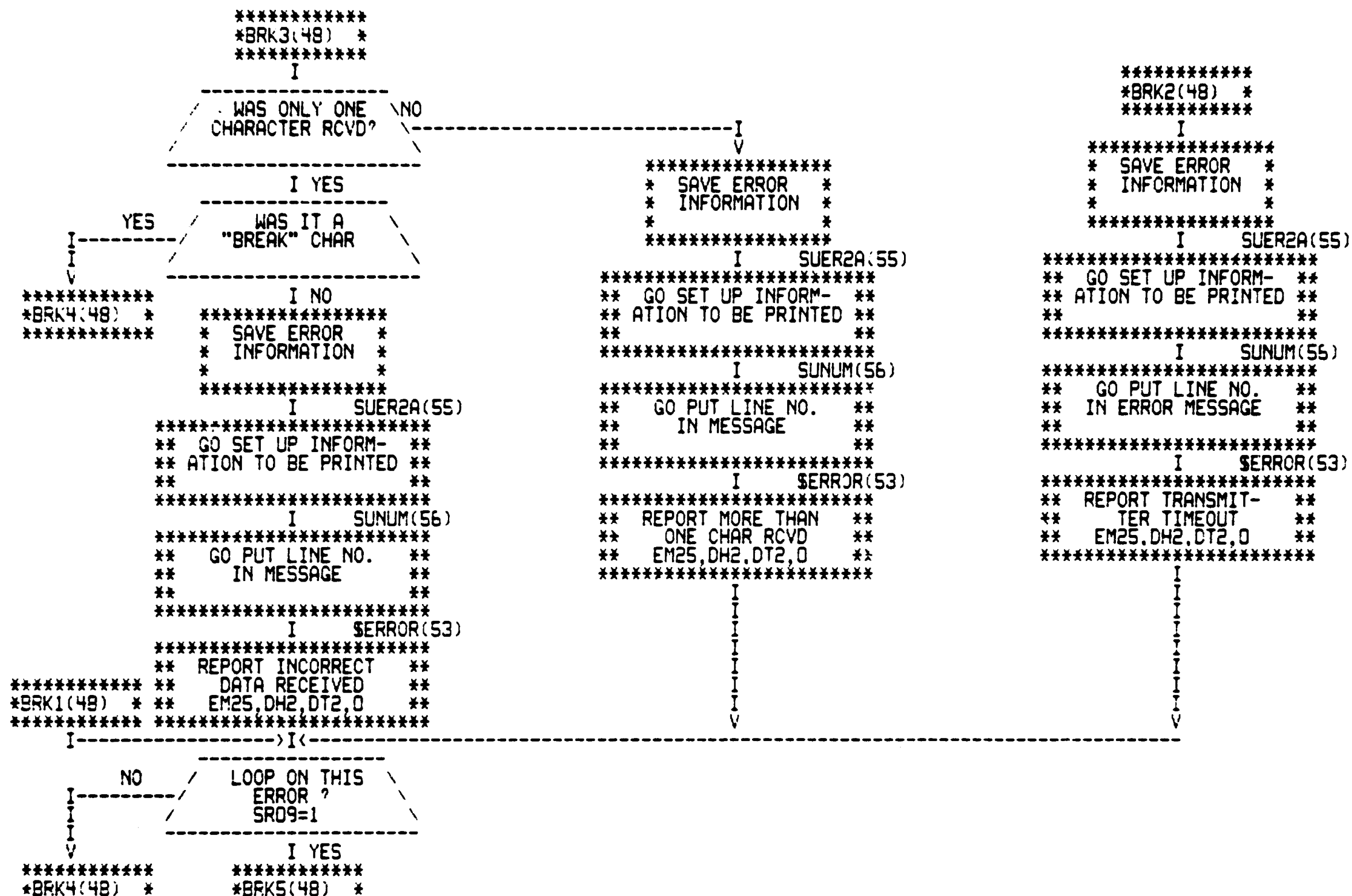
```

I
V

```

-----I-----
V
*****
* PRIME SELECTED LINE *
* TO XMIT 256. CHARS *
* SET THE "BREAK" BIT *
*****
I
*****
* ACTIVATE THE *
* SELECTED LINE *
* *
*****
I
*****
* INIT TIMERS *
* "A" AND "B" *
* *
*****
----->I-----
-----
/ BUFFER ACTIVE \ YES
/ BIT CLEARED \
/ YET ? \
-----
I NO TIMEIT(59)
*****
** **
** CALL THE TIMER **
** **
*****
I
-----
NO / TIMEOUT YET? \
-----
I YES
*****
*BRK2(49) *
*****
-----
I
V
*****
*BRK3(49) *
*****

```



[illegible]

```

*****
*TST54 *
*****
I
*****
* SET UP ERROR *
* LOOP RETURN TO *
* GO TO "B" *
*****
***
*A*----->I SELINE(60)
***
*****
** GO SET UP A LINE **
** NO. TO TEST **
**
*****
I
-----
/ DONE ALL SEL- \ YES
/ ECTED LINES ? \
-----
***
*B*----->I NO
***
*****
* PRIME SELECTED LINE *
* TO XMIT 68. CHARC. *
* (FORCE OVERRUN) *
*****
I
*****
* INIT TIMEFS *
* "A" AND "E" *
*****
----->I
-----
/ XMIT DONE YET? \ YES
/
-----
I NO TIMEIT(59)
*****
** CALL TIMER **
**
*****
I
-----
/ XMIT DONE \ YES ***
/ TIMEOUT ERROR? \ *C*
/
-----

```

```

-----I
V
*****
* READ 65. CHARS *
* FROM THE SILO *
*
*****
I
*****
* CHECK THAT OVERRUN *
* BIT SET PROPERLY ON *
* 65TH CHAR. *
*****
I
-----
***YES / OVERRUN BIT \
*A*-----/ SET PROPERLY AND \
*** / DATA CORRECT? \
-----
I NO
*****
* SAVE ERROR *
* INFORMATION *
*
*****
I SUER2A(55)
*****
**GO SET UP INFORMATION**
** TO BE PRINTED **
**
*****
I SUNUM(56)
*****
** GO SET UP LINE # **
** IN ERROR MESSAGE **
**
*****
I $ERROR(53)
*****
** REPORT OVERRUN **
** BIT FAILED **
** EMS6,DH2,DT2,0 **
*****
I<-----
*** NO / LOOP ON THIS \
*A*-----/ ERROR SRO9=1? \
*** /
-----
I YES
***
*B*

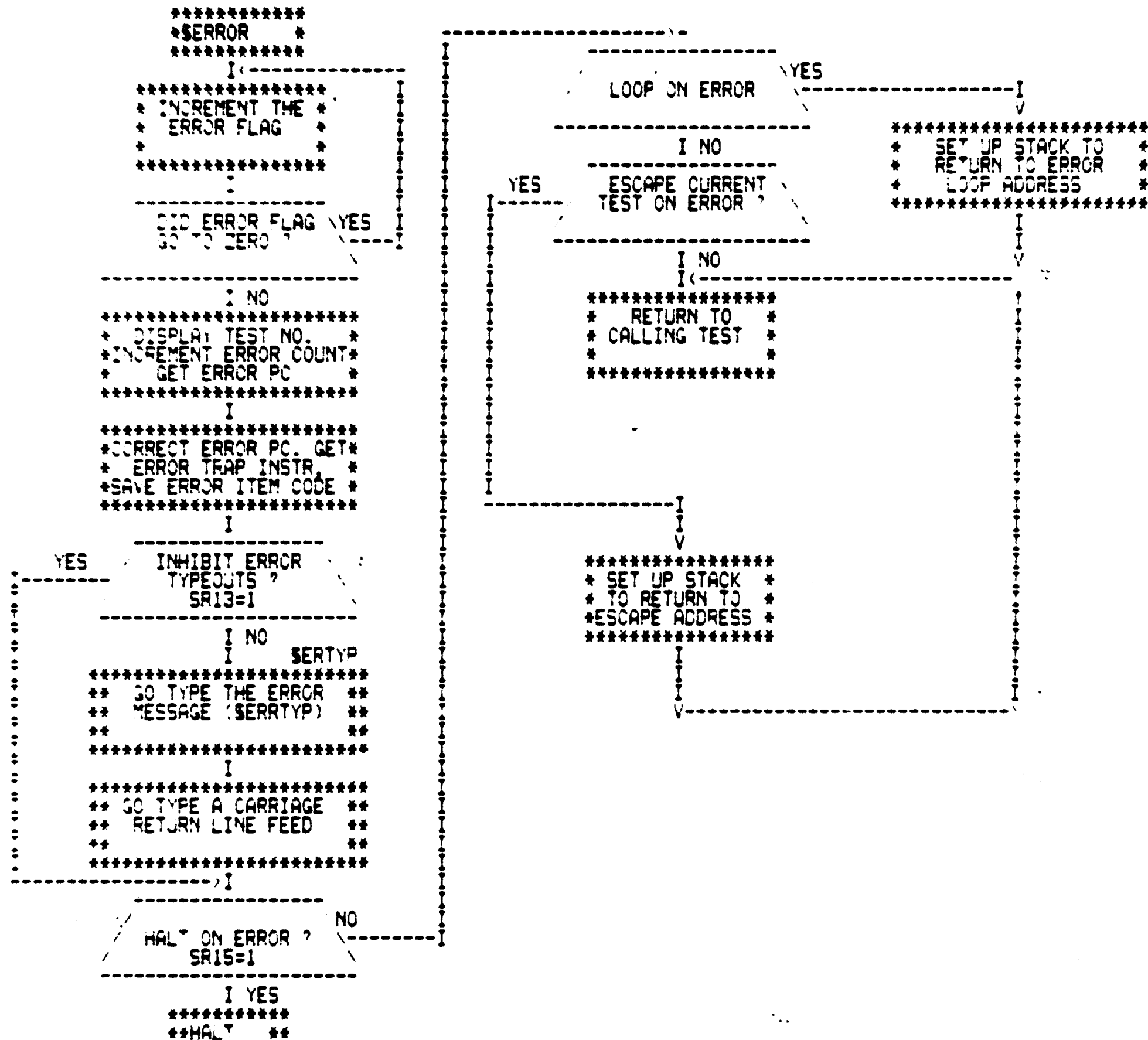
```

```

***
*C*
***
*****
* SAVE ERROR *
* INFORMATION *
*
*****
I SUER2A(55)
*****
** GO SET UP THE ERROR **
** INFORMATION **
**
*****
I SUNUM(56)
*****
** GO SET UP LINE # **
** IN ERROR MESSAGE **
**
*****
I $ERROR(53)
*****
** REPORT XMIT DONE **
** TIMEOUT ERROR **
** EMS0,DH2,DT2,0 **
*****
I

```


1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525



```

*****
*SET L*
*****
I
*****
* SET UP STMP6 *
* TO A 20(8) TO *
* COUNT 15. LINES*
*****
I
*****
* INIT R2 TO INDEX *
* READ BUFFER TEST *
* DATA TABLE *
*****
I
*****
* INIT R3 TO SET UP *
* VALID DATA BIT *
* LINE NO. IN TABLE *
*****
I
*****
* INIT R4 TO SET UP *
* FOR INDEXING HIGH *
* BYTE IN TABLE *
*****
I
*****
* INIT "SCR" TO *
* START WITH *
* LINE 00 *
*****
I
*****
* LOAD "CAR" "BCR". *
* AND "LPR" REGS *
* FOR SELECTED LINE *
*****
I
*****
* CLEAR LC BYTE TABLE *
* ENTRY AND LOAD [R3] *
* INTO HI BYTE *
*****
I
***
**A*
***

```

```

***
**A*
***
I
*****
* INCREMENT (SCR) *
* UPDATE R3,R2, AND *
* R4 FOR NEXT ENTRY *
*****
I
*****
* COUNT STMP6 *
* TO INDICATE THE *
* NEXT LINE TO DO *
*****
I
-----
NO HAVE WE SET
I----- UP ALL 15. LINES
-----
I YES
*****
**RTS **
*****
I
*****
*CHPS1*
*****
I
*****
* PUSH NEW PSW=000 *
* AND NEW PC=15 *
* ON TO STACK *
*****
I
*****
* DO AN RTI TO *
* LOAD NEW PSW *
* GO TO 15 *
*****
I
*****
* DO RTS TO *
* RETURN TO *
* CALLER *
*****
I
*****
**RTS **

```

```

*****
*CHPS2*
*****
I
*****
* PUSH PSW=340 *
* AND PC=15 *
* ON THE STACK *
*****
I
*****
* DO AN RTI TO *
* LOAD NEW PSW *
* GO TO 15 *
*****
I
*****
* DO AN RTS *
* TO RETURN TO *
* CALLER *
*****
I
*****
**RTS **
*****

```

```

*****
*SAPS*
*****
I
*****
* PUSH STACK TO *
* SAVE SPACE *
* FOR SAVPSW *
*****
I
*****
* PUSH THE CURRENT *
* CONTENTS OF THE *
* TRAP VECTOR *
*****
I
*****
* SET UP TRAP VECT- *
* OR TO GO TO 15 *
*****
I
*****
* DO A "TRAP" *
* INSTRUCTION *
*****
I
*****
* SAVE PSW, PUSH *
* 25 ON STACK *
* DO RTI *
*****
I
*****
* RESTORE TRAP VECTOR *
* POP STACK (SAVPS) *
* INTO STMP0 *
*****
I
*****
**RTS **
*****

```

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

```
*****
*LDTBFI *
*****
I
*****
* SET R1 TO *
* PCINT TO AMIT *
* BUFFER *
*****
```

```
*****
* INIT R2=0 *
* AS A CHAR *
* GENERATOR *
*****
```

```
*****
* LOAD [R2] INTO *
* XMI BUFFER *
* UPDATE R1 *
*****
```

```
*****
* GENERATE *
* NEXT CHAR *
*****
```

```
*****
* LOADED ALL *
* 256.. COMBINATIONS *
*****
```

```
I YES
*****
**RTS **
```

```
*****
*SUER1 *
*****
I SAPS(54)
*****
** GO SAVE PSW **
** IN STMPO **
*****
```

```
*****
* SAVE TEST NO. *
* IN RO *
*****
```

```
*****
* SAVE RO,R1,R2,R6 *
* IN SREG<0,1,2,6> *
*****
```

```
*****
* CORRECT SREG6 *
*****
```

```
*****
**RTS **
*****
```

```
*****
*SUER2 *
*****
I SAPS(54)
*****
** GO SAVE PSW **
** IN STMPO **
*****
```

```
*****
* SAVE TEST NO. *
* IN RO *
*****
```

```
*****
* SAVE RO,R1,R2,R3,R4, *
* AND R6 IN *
* SREG<0,1,2,3,4,6> *
*****
```

```
*****
* CORRECT SREG6 *
*****
```

```
*****
**RTS **
*****
```

```
*****
*SUER2A *
*****
I
*****
```

```
*****
* SUER4 *
*****
```

```
*****
* SAVE LINE NO. *
* IN STMPO *
*****
```

```
*****
* SAVE RO,R1,R2,R3,R4 *
* IN SREG<0,1,2,3,4> *
*****
```

```
*****
**RTS **
*****
```

```
*****
*SUER3 *
*****
I
*****
* SAVE RO,R1,R2 *
* IN SREG<0,1,2> *
*****
```

```
*****
**RTS **
*****
```

```
*****
* SUER4 *
*****
```

```
*****
* SAVE RO,R1,R2,R3,R4 *
* IN SREG<0,1,2,3,4> *
*****
```

```
*****
**RTS **
*****
```

```

*****
*SUNUM      *
*****
I
*****
* SAVE REGS RETRIEVE *
* ADDRESSES OF NUM  *
* AND MSG BUFFER    *
*****
I
*****
* GET NUMBER *
* TO CONVERT *
*****
I
*****
* CONVERT AND *
* STORE MSD  *
*****
I
*****
* CONVERT AND *
* STORE LSD  *
*****
I
*****
* RESTORE REGS *
*****
I
*****
**RTS **
*****

```

```

*****
*CLCABC    *
*****
I
*****
* INIT A LINE *
* COUNTER TO  *
* START AT 00 *
*****
I
*****
* SELECT A *
* LINE IN THE *
* "SCR"    *
*****
I
*****
* CLEAR "CAR" *
* AND "BCR"  *
* FOR SEL LINE *
*****
I
*****
* UPDATE LINE *
* COUNTER    *
*****
I
-----
CLEARED ALL \NO
16. LINES YET -----
I
I YES
*****
* INIT "SCR" *
* TO SELECT  *
* LINE 00    *
*****
I
*****
**RTS **
*****

```

```

*****
*LDBCR     *
*****
I
*****
* INIT A LINE *
* COUNTER TO  *
* STRT AT 00  *
*****
I
*****
* SELECT A *
* LINE IN THE *
* "SCR"    *
*****
I
*****
* LOAD THAT *
* LINE'S "BCR" *
* WITH A 177777 *
*****
I
*****
* GENERATE NEXT *
* LINE NO.      *
*****
I
-----
DONE 16. LINES ? \NO
-----
I
I YES
*****
**RTS **
*****

```

```

*****
*ENDA(5:)*
*****
I
*****
*$SCOPE(52)*
*****
*****
* "NOP" SCOPE *
* CALL IN SEOP *
* ROUTINE *
*****
I
*****
* GENERATE NEXT *
* DEVICE NO. *
* *
*****
I
*****
* UPDATE DEVICE PARA- *
* METER TABLE POINTERS *
* SHIFT SELECT MASK *
*****
I
-----
/ TESTED ALL \ YES
/ POSSIBLE DM11'S ----- I
-----
I NO
-----
NO / IS THIS DM11
/ SELECTED FOR TEST \
-----
I YES
*****
* CLEAR $STNM *
* TO START AT *
* TST 1 AGAIN *
*****
I
*****
P" TO * *RSTRTA(02)*
ND OF *
*****
*****

```



```

*****
*CHKADR *
*****
I
-----
/ WAS DEVICE ADDR \ NO
/ TYPED GREATER \
/ THAN 160020(8) ? \
-----
15 I YES
-----
/ WAS IT LESS \ NO
/ THAN 160420(8) ? \
-----
25 I YES
-----
/ WAS IT A MULT- \ NO
/ IPLE OF 20(8) ? \
-----
45 I YES
*****
/ PRINT INVALID
/ ADDRESS MESSAGE
/
*****
35 I
*****
*GENERATE AND LOAD 16.*
* ENTRY DH11 DEVICE *
* ADDRESS TABLE *
*****
I<-----
*****
**RTS **
*****

```

```

*****
*CHKVCT *
*****
I
-----
/ WAS VECTOR ADDR \ NO
/ TYPED GREATER \
/ THAN 300(8) \
-----
15 I YES
-----
/ WAS IT LESS \ NO
/ THAN 1000(8) \
-----
25 I YES
-----
/ WAS IT A MULT- \ NO
/ IPLE OF 10(8) \
-----
45 I YES
*****
/ PRINT INVALID
/ VECTOR MESSAGE
/
*****
35 I
*****
*GENERATE AND LOAD *
* 16.ENTRY DH11 VECT. *
* ADDRESS TABLE *
*****
I<-----
*****
**RTS **
*****

```

```

*****
*BUSER*
*****
I
*****
* SAVE THE PSW *
* AND STACK *
* POINTER *
*****
I
*****
* RETRIEVE AND SAVE *
* "TRAPPC" AND "TRAPPS" *
* FROM STACK *
*****
I
*****
* INITIALIZE THE *
* STACK POINTER *
*****
I $ERROR(53)
*****
** REPORT THE BUS **
** ERROR **
** EM14,DH4,DT5,DF2 **
*****
I
*****
* RESET THE *
* WORLD AND *
* CLEAR PSW *
*****
I
*****
*REST1*
*****

```

```

*****
*RESERR*
*****
I
*****
* SAVE THE PSW *
* AND STACK *
* POINTER *
*****
I
*****
* RETRIEVE AND SAVE *
* "TRAPPC" AND "TRAPPS" *
* FROM STACK *
*****
I
*****
* INITIALIZE *
* STACK POINTER *
*****
I $ERROR(53)
*****
** REPORT THE RSVD **
** INSTR ERROR **
** EM15,DH4,DT5,DF2 **
*****
I
*****
* RESET THE *
* WORLD AND *
* CLEAR PSW *
*****
I
*****
*REST1*
*****

```

```

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

```



```

*****  

*SELINE *  

*****  

I  

-----  

NO / 1ST TIME THRU \  

|----FOR CURRENT TEST ? |  

|  

|  

|  

| I YES  

| *****  

| * SET ENTRY FLAG INIT *  

| * MASK AND "LINE" TO *  

| * BEGIN WITH LINE 00 *  

| *****  

| I  

| *****  

| * GENERATE LINE *  

|-----> * NO. IN "LINE" * <-----  

| * *  

| *****  

| I  

| *****  

| * SHIFT SELECT *  

| * TEST MASK *  

| * *  

| *****  

| I  

|-----  

YES / TESTED FOR ALL \  

|-----16. LINES? |  

|  

| I NO  

|-----  

| / IS THIS LINE \ NO  

| / SELECTED FOR |  

| / TESTING ? |  

|-----  

V  

***** I YES  

LINE # * *****  

ENTRY * * MOVE RETURN *  

LAG * * PC AROUND *  

***** * EXIT BRANCH *  

*****  

I  

I  

I *****  

|-----> * INIT LINE SEL. *  

|-----> * BITS IN "SOR" *  

| * *  

| *****  

| I  

| *****  

| **RTS **

```

AE1	45	45																
AE2	45	45																
AE3	45	45																
AE4	45	45	45															
AE5	45	45	45															
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														
ENDR	51	57																
HALT	02	53																
INPAR	01	02																
INPAR3	01	02	02															
INPARA	01	57																
LOBCR	30	56																
LOTBF1	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
SAFE	54	55	55	60	60													
SELNE	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	05	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							

K09

SEQ 0114

[illegible]

L09

SEQ 0115

[illegible]

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

SEQ 0116

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 SRO8 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.

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.

	OPERATIONAL SWITCH SETTINGS
	TRAP CATCHER
	STARTING ADDRESS(ES)
	ACT11 HOOKS
	APT PARAMETER BLOCK
	BASIC DEFINITIONS
	COMMON TAGS
	APT MAILBOX-ETABLE
	ERROR POINTER TABLE
	INITIALIZE THE COMMON TAGS
	GET VALUE FOR SOFTWARE SWITCH REGISTER
T1	CHECK SSYN RESPONSE FROM ALL DM11 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	TRANSMITTER 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 MODEM CONTROL DIAGNOSTIC

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33
 DZDMMB.P11 TABLE OF CONTENTS

SEQ 0119

5320	156	DM11-BB DIAGNOSTIC CONTINUED
5320	157	DM11-BB DIAGNOSTIC CONTINUED
5320	158	DM11-BB DIAGNOSTIC CONTINUED
5320		END OF PASS ROUTINE
5320		SCOPE HANDLER ROUTINE
5320		ERROR HANDLER ROUTINE
5320		ERROR MESSAGE TIMEOUT ROUTINE
5320		BINARY TO OCTAL (ASCII) AND TYPE
5320		CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
5320		TYPE ROUTINE
5320		APT COMMUNICATIONS ROUTINE
5320		TTY INPUT ROUTINE
5320		READ AN OCTAL NUMBER FROM THE TTY
5320		TRAP DECODER
5320		TRAP TABLE
5320		POWER DOWN AND UP ROUTINES

D10

MAINDEX-11-DZDMM-B
DHMMAC.P11

MACY11 27(732) 23-SEP-76 14:33 PAGE 1

SEQ 0120

ih.ewrur-

6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

165400

```
.NLIST CND.MD.MC
.LIST TOC.ME.SEO,BIN
$SWR=165400
.MCALL .HEADER..$CMTAG..SETUP..$ERROR
.MCALL .SWRHI..SWRLO..$CATCH..$EQUATE.SETUP..$SEOP.GETSWR
.MCALL .$$SCOPE..$ERRTYP..$STYPOCT..$STYPDEC..$TYPE..$POWER..$TRAP
.MCALL .$READ..$RDOCT
.MCALL .$ACT11..$APTHDR..$APTCLS..$APTTYPE
.ENABLE ABS
.TITLE MAINDEC-11-DZDMM-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 000174
000176 000000
000200 000137 026114
000204 000137 002146
000210 000137 026104
```

```
.=174
DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
SWREG:   .WORD 0      ;;SOFTWARE SWITCH REGISTER
.SBTTL STARTING ADDRESS(ES)
      JMP      @INPARX      ;;JUMP TO STARTING ADDRESS OF PROGRAM
      JMP      @BEGIN       ;;BEGIN EXECUTION WITH DEFAULT PARAMETERS
      JMP      @INPARC      ;;INPUT PARAMETERS - DEVICE SELECTION ONLY
```

.SBTTL ACT11: HOOKS

```
;;*****
;;HOOKS REQUIRED BY ACT11
```

```
000214 000214      $SVPC=.      ;SAVE PC
000046 000046      .=46
000046 000046      $ENDAD      ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
000052 000052      .=52
000052 120000      .WORD 120000      ;;2)SET LOC.52 TO 120000
000214 000214      .=$SVPC      ;;RESTORE PC
.SBTTL APT PARAMETER BLOCK
```

```

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

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

.SBTTL BASIC DEFINITIONS

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

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

;*GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0      ;;GENERAL REGISTER
R1= %1      ;;GENERAL REGISTER
R2= %2      ;;GENERAL REGISTER
R3= %3      ;;GENERAL REGISTER
R4= %4      ;;GENERAL REGISTER
R5= %5      ;;GENERAL REGISTER
R6= %6      ;;GENERAL REGISTER
R7= %7      ;;GENERAL REGISTER
SP= %6      ;;STACK POINTER
PC= %7      ;;PROGRAM COUNTER

;*PRIORITY LEVEL DEFINITIONS
PRO= 0      ;;PRIORITY LEVEL 0
PRI= 40     ;;PRIORITY LEVEL 1

```

118	000100	PR2=	100	::PRIORITY LEVEL	2
119	000140	PR3=	140	::PRIORITY LEVEL	3
120	000200	PR4=	200	::PRIORITY LEVEL	4
121	000240	PR5=	240	::PRIORITY LEVEL	5
122	000300	PR6=	300	::PRIORITY LEVEL	6
123	000340	PR7=	340	::PRIORITY LEVEL	7

:*"SWITCH REGISTER" SWITCH DEFINITIONS

124	100000	SW15=	100000
125	040000	SW14=	40000
126	020000	SW13=	20000
127	010000	SW12=	10000
128	004000	SW11=	4000
129	002000	SW10=	2000
130	001000	SW09=	1000
131	000400	SW08=	400
132	000200	SW07=	200
133	000100	SW06=	100
134	000040	SW05=	40
135	000020	SW04=	20
136	000010	SW03=	10
137	000004	SW02=	4
138	000002	SW01=	2
139	000001	SW00=	1
140		.EQUIV	SW09,SW9
141		.EQUIV	SW08,SW8
142		.EQUIV	SW07,SW7
143		.EQUIV	SW06,SW6
144		.EQUIV	SW05,SW5
145		.EQUIV	SW04,SW4
146		.EQUIV	SW03,SW3
147		.EQUIV	SW02,SW2
148		.EQUIV	SW01,SW1
149		.EQUIV	SW00,SW0

:*DATA BIT DEFINITIONS (BIT00 TO BIT15)

150	100000	BIT15=	100000
151	040000	BIT14=	40000
152	020000	BIT13=	20000
153	010000	BIT12=	10000
154	004000	BIT11=	4000
155	002000	BIT10=	2000
156	001000	BIT09=	1000
157	000400	BIT08=	400
158	000200	BIT07=	200
159	000100	BIT06=	100
160	000040	BIT05=	40
161	000020	BIT04=	20
162	000010	BIT03=	10
163	000004	BIT02=	4
164	000002	BIT01=	2
165	000001	BIT00=	1
166		.EQUIV	BIT09,BIT9
167		.EQUIV	BIT08,BIT8
168		.EQUIV	BIT07,BIT7
169		.EQUIV	BIT06,BIT6

```

174 .EQUIV BIT05,BIT5
175 .EQUIV BIT04,BIT4
176 .EQUIV BIT03,BIT3
177 .EQUIV BIT02,BIT2
178 .EQUIV BIT01,BIT1
179 .EQUIV BIT00,BIT0
180
181
182
183
184
185
186
187
188
189
190
191
192
193
000004
000010
000014
000014
000014
000014
000020
000024
000030
000034
000060
000064
000240

;*BASIC "CPU" TRAP VECTOR ADDRESSES
ERRVEC= 4 ;; TIME OUT AND OTHER ERRORS
RESVEC= 10 ;; RESERVED AND ILLEGAL INSTRUCTIONS
TBITVEC= 14 ;; "T" BIT
TRTVEC= 14 ;; TRACE TRAP
BPTVEC= 14 ;; BREAKPOINT TRAP (3PT)
IOTVEC= 20 ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
PWRVEC= 24 ;; POWER FAIL
EMTVEC= 30 ;; EMULATOR TRAP (EMT) **ERROR**
TRAPVEC= 34 ;; "TRAP" TRAP
TKVEC= 60 ;; TTY KEYBOARD VECTOR
TPVEC= 64 ;; TTY PRINTER VECTOR
PIRQVEC= 240 ;; PROGRAM INTERRUPT REQUEST VECTOR

```

.SBTTL COMMON TAGS

```

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

```

194									
195									
196									
197									
198									
199									
200		001100							
201	001100		SCMTAG:	.=1100					::START OF COMMON TAGS
202	001100	000000							
203	001102	000							
204	001103	000							
205	001104	000000							
206	001106	000000							
207	001110	000000							
208	001112	000000							
209	001114	000							
210	001115	001							
211	001116	000000							
212	001120	000000							
213	001122	000000							
214	001124	000000							
215	001126	000000							
216	001130	000000							
217	001132	000000							
218	001134	000							
219	001135	000							
220	001136	000000							
221	001140	177570							
222	001142	177570							
223	001144	177560							
224	001146	177562							
225	001150	177564							
226	001152	177566							
227	001154	000							
228	001155	002							
229	001156	012							
230	001157	000							
231	001160	000000							
232									
233	001162	000000							
234	001164	000000							
235	001166	000000							
236	001170	000000							
237	001172	000000							
238	001174	000000							
239	001176	000000							
240	001200	000000							
241	001202	000000							
242	001204	000000							
243	001206	000000							
244	001210	000000							
245	001212	000000							
246	001214	000000							
247	001216	000000							
248	001220	000000							
249	001222	000000							

250	001224	000000	\$ESCAPE:0	::ESCAPE CN ERROR ADDRESS
251	001226	077	\$QUES: .ASCII /?/	::QUESTION MARK
252	001227	015	\$CRLF: .ASCII <15>	::CARRIAGE RETURN
253	001230	000012	\$LF: .ASCII <12>	::LINE FEED
254			:*****	
255			.SBTTL APT MAILBOX-ETABLE	
256			:*****	
257			.EVEN	
258			\$MAIL:	::APT MAILBOX
259	001232		\$MSGTY: .WORD	::MESSAGE TYPE CODE
260	001232	000000	\$FATAL: .WORD	::FATAL ERROR NUMBER
261	001234	000000	\$TESTN: .WORD	::TEST NUMBER
262	001236	000000	\$PASS: .WORD	::PASS COUNT
263	001240	000000	\$DEVCT: .WORD	::DEVICE COUNT
264	001242	000000	\$UNIT: .WORD	::I/O UNIT NUMBER
265	001244	000000	\$MSGAD: .WORD	::MESSAGE ADDRESS
266	001246	000000	\$MSGLG: .WORD	::MESSAGE LENGTH
267	001250	000000	\$ETABLE:	::APT ENVIRONMENT TABLE
268	001252		\$ENV: .BYTE	::ENVIRONMENT BYTE
269	001252	000	\$ENVM: .BYTE	::ENVIRONMENT MODE BITS
270	001253	000	\$SWREG: .WORD	::APT SWITCH REGISTER
271	001254	000000	\$USWR: .WORD	::USER SWITCHES
272	001256	000000	\$CPUOP: .WORD	::CPU TYPE, OPTIONS
273	001260	000000	BITS 15-11=CPU TYPE	
274			11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05	
275			11/70=06, PDQ=07, Q=10	
276			BIT 10=REAL TIME CLOCK	
277			BIT 9=FLOATING POINT PROCESSOR	
278			BIT 8=MEMORY MANAGEMENT	
279			::HIGH ADDRESS, M.S. BYTE	
280	001262	000	\$MAMS1: .BYTE	::MEM. TYPE, BLK#1
281	001263	000	\$MTYP1: .BYTE	::MEM. TYPE, BLK#1
282			MEM. TYPE BYTE -- (HIGH BYTE)	
283			900 NSEC CORE=001	
284			300 NSEC BIPOLAR=002	
285			500 NSEC MOS=003	
286	001264	000000	\$MADR1: .WORD	::HIGH ADDRESS, BLK#1
287			MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE	
288	001266	000	\$MAMS2: .BYTE	::HIGH ADDRESS, M.S. BYTE
289	001267	000	\$MTYP2: .BYTE	::MEM. TYPE, BLK#2
290	001270	000000	\$MADR2: .WORD	::MEM. LAST ADDRESS, BLK#2
291	001272	000	\$MAMS3: .BYTE	::HIGH ADDRESS, M.S. BYTE
292	001273	000	\$MTYP3: .BYTE	::MEM. TYPE, BLK#3
293	001274	000000	\$MADR3: .WORD	::MEM. LAST ADDRESS, BLK#3
294	001276	000	\$MAMS4: .BYTE	::HIGH ADDRESS, M.S. BYTE
295	001277	000	\$MTYP4: .BYTE	::MEM. TYPE, BLK#4
296	001300	000000	\$MADR4: .WORD	::MEM. LAST ADDRESS, BLK#4
297	001302	000000	\$VECT1: .WORD	::INTERRUPT VECTOR#1 BUS PRIORITY#1
298	001304	000000	\$VECT2: .WORD	::INTERRUPT VECTOR#2 BUS PRIORITY#2
299	001306	000000	\$BASE: .WORD	::BASE ADDRESS OF EQUIPMENT UNDER TEST
300	001310	000000	\$DEVN: .WORD	::DEVICE MAP
301	001312	000000	\$CDW1: .WORD	::CONTROLLER DESCRIPTION WORD#1
302	001314	000000	\$CDW2: .WORD	::CONTROLLER DESCRIPTION WORD#2
303	001316	000000	\$DDW0: .WORD	::DEVICE DESCRIPTOR WORD#0
304	001320	000000	\$DDW1: .WORD	::DEVICE DESCRIPTOR WORD#1
305	001322	000000	\$DDW2: .WORD	::DEVICE DESCRIPTOR WORD#2

K10

MAXDEC-11-DZDHM-B MAC 11 27:7321 23-SEP-76 14:33 PAGE 9
DZDHMB.P11 APT MAIL BOX-STABLE

SEQ 0127

ADDRESS	DATA	DESCRIPTION
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		


```

323 .SBTTL ERROR POINTER TABLE
324
325 :*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
326 :*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
327 :*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
328 :*NOTE1: IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
329 :*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
330
331 :*      EM      ;;POINTS TO THE ERROR MESSAGE
332 :*      DH      ;;POINTS TO THE DATA HEADER
333 :*      DT      ;;POINTS TO THE DATA
334 :*      DF      ;;POINTS TO THE DATA FORMAT
335
336
337 001356 $ERRTB:
338 ;ERROR TABLE ITEM FOR ERROR MESSAGE 1
339
340 001356 030250 EM1 ;"DH11 REGISTER REFERENCE CAUSED TIMEOUT"
341 001360 030317 DH1 ;" (PC) (PS) (SP) TEST DEVADR REGADR "
342 001362 030376 DT1 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2
343 001364 000000 0 ;PRINT ALL OCTAL
344
345 ;ERROR TABLE ITEM FOR ERROR MESSAGE 2
346
347 001366 030414 EN2 ;"SYSTEM CONTROL REGISTER ERROR"
348 001370 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
349 001372 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
350 001374 000000 0 ;PRINT ALL OCTAL
351
352 ;ERROR TABLE ITEM FOR ERROR MESSAGE 3
353
354 001376 030572 EM3 ;"DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG"
355 001400 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
356 001402 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
357 001404 000000 0 ;PRINT ALL OCTAL
358
359 ;ERROR TABLE ITEM FOR ERROR MESSAGE 4
360
361 001406 030650 EM4 ;"LINE PARAMETER REGISTER ERROR"
362 001410 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
363 001412 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
364 001414 000000 0 ;PRINT ALL OCTAL
365
366 ;ERROR TABLE ITEM FOR ERROR MESSAGE 5
367
368 001416 030706 EM5 ;"BREAK CONTROL REGISTER ERROR"
369 001420 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
370 001422 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
371 001424 000000 0 ;PRINT ALL OCTAL
372
373 ;ERROR TABLE ITEM FOR ERROR MESSAGE 6
374
375 001426 030743 EM6 ;"SILO STATUS REGISTER ERROR"
376 001430 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
377 001432 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
378 001434 000000 0 ;PRINT ALL OCTAL
  
```

M10

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 11
 DZDMMB.P11 ERROR POINTER TABLE

SEQ 0129

```

379
380 ;ERROR TABLE ITEM FOR ERROR MESSAGE 7
381
382 001436 030776 EM7 ;"CURRENT ADDRESS REGISTER ERROR - LINE #XX"
383 001440 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
384 001442 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
385 001444 000000 0 ;PRINT ALL OCTAL
386
387 ;ERROR TABLE ITEM FOR ERROR MESSAGE 10
388
389 001446 031050 EM10 ;"BYTE COUNTER REGISTER ERROR - LINE #XX"
390 001450 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
391 001452 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
392 001454 000000 0 ;PRINT ALL OCTAL
393
394 ;ERROR TABLE ITEM FOR ERROR MESSAGE 11
395
396 001456 031117 EM11 ;"UNEXPECTED DH11 RCVR INTERRUPT"
397 001460 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
398 001462 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
399 001464 000000 0 ;PRINT ALL OCTAL
400
401 ;ERROR TABLE ITEM FOR ERROR MESSAGE 12
402
403 001466 031156 EM12 ;"UNEXPECTED DH11 XMITTR INTERRUPT"
404 001470 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
405 001472 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
406 001474 000000 0 ;PRINT ALL OCTAL
407
408 ;ERROR TABLE ITEM FOR ERROR MESSAGE 13
409
410 001476 031217 EM13 ;"CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT"
411 001500 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
412 001502 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
413 001504 000000 0 ;PRINT ALL OCTAL
414
415 ;ERROR TABLE ITEM FOR ERROR MESSAGE 14
416
417 001506 031300 EM14 ;"TRANSMITTER NPR LOGIC ERROR - LINE # "
418 001510 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
419 001512 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
420 001514 000000 0 ;PRINT ALL OCTAL
421
422 ;ERROR TABLE ITEM FOR ERROR MESSAGE 15
423
424 001516 031347 EM15 ;"XMITTR FAILED TO INTERRUPT - LINE # "
425 001520 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
426 001522 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
427 001524 000000 0 ;PRINT ALL OCTAL
428
429 ;ERROR TABLE ITEM FOR ERROR MESSAGE 16
430
431 001526 031415 EM16 ;"RCVR FAILED TO INTERRUPT"
432 001530 030452 DH2 ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
433 001532 030550 DT2 ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
434 001534 000000 0 ;PRINT ALL OCTAL

```

N10

MAINDEC-11-DZDHM-8 MACY11 27(732) 23-SEP-76 14:33 PAGE 12
 DZDHMB.P11 ERROR POINTER TABLE

SEQ 0130

```

435
436
437
438
439 001536 031446      EM17      ;"TRANSMITTER TIMING ERROR - LINE # "
440 001540 031512      DH6        ;" (PC) (PS) (SP) TEST DEVADR SPEED TIMEB TIMEC"
441 001542 030550      DT2        ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
442 001544 000000      0          ;PRINT ALL OCTAL
443
444
445
446 001546 031610      EM20      ;RECEIVER TIMING ERROR - LINE # "
447 001550 031512      DH6        ;" (PC) (PS) (SP) TEST DEVADR SPEED TIMEB TIMEC"
448 001552 030550      DT2        ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
449 001554 000000      0          ;PRINT ALL OCTAL
450
451
452
453 001556 031651      EM21      ;"RCVR FAILED TO INTERRUPT - LINE # "
454 001560 030452      DH2        ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
455 001562 030550      DT2        ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
456 001564 000000      0          ;PRINT ALL OCTAL
457
458
459
460 001566 031715      EM22      ;"CHAR AVAIL FAILED TO SET ON TIME - LINE # "
461 001570 030452      DH2        ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
462 001572 030550      DT2        ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
463 001574 000000      0          ;PRINT ALL OCTAL
464
465
466
467 001576 031771      EM23      ;"BASIC DATA TEST ERROR - LINE # "
468 001600 032032      DH7        ;" (PC) (PS) (SP) TEST DEVADR CHRLNG WAS S/B "
469 001602 030550      DT2        ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
470 001604 000000      0          ;PRINT ALL OCTAL
471
472
473
474 001606 032127      EM24      ;"AUTO ECHO TEST ERROR - LINE # "
475 001610 030452      DH2        ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
476 001612 030550      DT2        ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
477 001614 000000      0          ;PRINT ALL OCTAL
478
479
480
481 001616 032167      EM25      ;"BREAK BIT TEST ERROR - LINE # "
482 001620 030452      DH2        ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
483 001622 030550      DT2        ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
484 001624 000000      0          ;PRINT ALL OCTAL
485
486
487
488 001626 032227      EM26      ;"HALF-DUPLEX TEST ERROR - LINE # "
489 001630 030452      DH2        ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
490 001632 030550      DT2        ;$ERRPC,$TMPD,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4

```

ERROR POINTER TABLE

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	: SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2
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	: SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2
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
001662	030550	DT2	: SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
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	: SERRPC, STMP0, STMP2
001674	000000	0	:PRINT ALL OCTAL
:ERROR TABLE ITEM FOR ERROR MESSAGE 33			
001676	032720	EM33	: "PARITY LOGIC TEST ERROR - LINE # "
001700	030453	DH2	: (PC) (PS) (SP) TEST DEVADR REGADR WAS
001702	030550	DT2	: SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
001704	000000	0	:PRINT ALL OCTAL
:ERROR TABLE ITEM FOR ERROR MESSAGE 34			
001706	032763	EM34	: "MULTI-LINE PARITY DATA TEST ERROR - LINE # - SUBTEST # "
001710	032523	DH4	: (PC) (PS) (SP) TEST WASADR SBADR WAS
001712	030550	DT2	: SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
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	: SERRPC, STMP0, STMP3
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	:SERRPC,STMPD,STMP2
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 B "
001742	030550	DT2	:SERRPC,STMPD,\$REG6,\$REG0,\$REG1,\$REG2,\$REG3,\$REG4
001744	000000	0	:PRINT ALL OCTAL
:ERROR TABLE ITEM FOR ERROR 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	:SERRPC,STMPD,\$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	:SERRPC,STMPD,STMP2
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	:SERRPC,STMPD,STMP2
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	:SERRPC,STMPD,\$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	:SERRPC,STMPD,\$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	:SERRPC,STMPD,\$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) LINEAR PATTN TEST DEVADR REGADR WAS S B "

603	002032	033766	D15	:SERRPC,STMP0,STMP1,\$REG0,\$REG1,\$REG2,\$REG3,\$REG4	
604	002034	000000	0	:PRINT ALL OCTAL	
605					
606				:ERROR TABLE ITEM FOR ERROR MESSAGE 47	
608	002036	034010	EM47	: "BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE # "	
609	002040	033671	DH10	: (PC) LINEAR PATTN TEST DEVADR REGADR WAS	S/B"
610	002042	033766	DT5	:SERRPC,STMP0,STMP1,\$REG0,\$REG1,\$REG2,\$REG3,\$REG4	
611	002044	000000	0	:PRINT ALL OCTAL	
612					
613				:ERROR TABLE ITEM FOR ERROR MESSAGE 50	
614					
615	002046	034071	EM50	: "TEST TIMEOUT WAITING FOR XMIT DONE - LINE # "	
616	002050	030452	DH2	: (PC) (PS) (SP) TEST DEVADR REGADR WAS	S/B"
617	002052	030550	DT2	:SERRPC,STMP0,\$REG6,\$REG0,\$REG1,\$REG2,\$REG3,\$REG4	
618	002054	000000	0	:PRINT ALL OCTAL	
619					
620				:ERROR TABLE ITEM FOR ERROR MESSAGE 51	
621					
622	002056	034147	EM51	: "NPR LOGIC TEST 2 ERROR"	
623	002060	034176	DH11	: (PC) LINACT LINCHK TEST DEVADR REGADR WAS	S B"
624	002062	033766	DT5	:SERRPC,STMP0,STMP1,\$REG0,\$REG1,\$REG2,\$REG3,\$REG4	
625	002064	000000	0	:PRINT ALL OCTAL	
626					
627				:ERROR TABLE ITEM FOR ERROR MESSAGE 52	
628					
629	002066	034273	EM52	: "BASIC DATA COMPARE ERROR"	
630	002070	030452	DH2	: (PC) (PS) (SP) TEST DEVADR REGADR WAS	S B"
631	002072	030550	DT2	:SERRPC,STMP0,\$REG6,\$REG0,\$REG1,\$REG2,\$REG3,\$REG4	
632	002074	000000	0	:PRINT ALL OCTAL	
633					
634				:ERROR TABLE ITEM FOR ERROR MESSAGE 53	
635					
636	002076	034071	EM50	: "TEST TIMEOUT WAITING FOR XMIT DONE - LINE # "	
637	002100	034324	DH12	: (PC) SPEED (SP) TEST DEVADR REGADR WAS	S B"
638	002102	030550	DT2	:SERRPC,STMP0,\$REG6,\$REG0,\$REG1,\$REG2,\$REG3,\$REG4	
639	002104	000000	0	:PRINT ALL OCTAL	
640					
641				:ERROR TABLE ITEM FOR ERROR MESSAGE 54	
642					
643	002106	031715	EM22	: "CHAR AVAIL FAILED TO SET ON TIME - LINE # "	
644	002110	034324	DH12	: (PC) SPEED (SP) TEST DEVADR REGADR WAS	S/B"
645	002112	030550	DT2	:SERRPC,STMP0,\$REG6,\$REG0,\$REG1,\$REG2,\$REG3,\$REG4	
646	002114	000000	0	:PRINT ALL OCTAL	
647					
648				:ERROR TABLE ITEM FOR ERROR MESSAGE 55	
649					
650	002116	031715	EM22	: "CHAR AVAIL FAILED TO SET ON TIME - LINE # "	
651	002120	034421	DH13	: (PC) (PS) (SP) TEST DEVADR CHRLNG SCRNAS SCRS B	
652	002122	030550	DT2	:SERRPC,STMP0,\$REG6,\$REG0,\$REG1,\$REG2,\$REG3,\$REG4	
653	002124	000000	0	:PRINT ALL OCTAL	
654					
655				:ERROR TABLE ITEM FOR ERROR MESSAGE 56	
656					
657	002126	034520	EM56	: "OVERRUN BIT FAILED TO SET - LINE # "	
658	002130	030452	DH2	: (PC) (PS) (SP) TEST DEVADR REGADR WAS	S B"

E11

MAINDEC-11-DZDMM-B MF 11 27(732) 23-SEP-76 14:33 PAGE 16
DZDMMB.P11 ERROR POINTER TABLE

SEQ C134

659	002132	030550	DI2	:SERRPC,STMPD,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4"
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	DI2	: " (PC) (PS) (SP) TEST DEJADR REGADR WAS
666	002142	030550	DI2	:SERRPC,STMPD,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4"
667	002144	000000	0	:PRINT ALL OCTAL
668				

S/B"

669	002146	005000			BEGIN: CLR RO	;;INIT RO TO INDICATE DEFAULT PARAMETERS
670	002150	005067	026062		BEGIN: CLR TITFLG	;;INIT TITLE MESSAGE FLAG
671					.SBTTL INITIALIZE THE COMMON TAGS	
672					;;CLEAR THE COMMON TAGS (\$CMTAG) AREA	
673	002154	012706	001100		MOV \$CMTAG,R6	;;FIRST LOCATION TO BE CLEARED
674	002160	005026			CLR (R6)+	;;CLEAR MEMORY LOCATION
675	002162	022706	001140		CMP \$SWR,R6 ; ; DONE?	
676	002166	001374			BNE -6	;;LOOP BACK IF NO
677	002170	012706	001100		MOV \$STACK,SP	;;SETUP THE STACK POINTER
678					;;INITIALIZE A FEW VECTORS	
679	002174	012737	020746	000020	MOV \$SCOPE,\$IOTVEC	;;IOT VECTOR FOR SCOPE ROUTINE
680	002202	012737	000340	000022	MOV \$340,\$IOTVEC+2	;;LEVEL 7
681	002210	012737	021236	000030	MOV \$ERROR,\$EMTVEC	;;EMT VECTOR FOR ERROR ROUTINE
682	002216	012737	000340	000032	MOV \$340,\$EMTVEC+2	;;LEVEL 7
683	002224	012737	024022	000034	MOV \$TRAP,\$TRAPVEC	;;TRAP VECTOR FOR TRAP CALLS
684	002232	012737	000340	000036	MOV \$340,\$TRAPVEC+2	;;LEVEL 7
685	002240	012737	024104	000024	MOV \$SPWRDN,\$PWAVEC	;;POWER FAILURE VECTOR
686	002246	012737	000340	000026	MOV \$340,\$PWAVEC+2	;;LEVEL 7
687	002254	016767	016400	016370	MOV \$ENDCT,\$EOPCT	;;SETUP END-OF-PROGRAM COUNTER
688	002262	005067	176734		CLR \$TIMES	;;INITIALIZE NUMBER OF ITERATIONS
689	002266	005067	176732		CLR \$ESCAPE	;;CLEAR THE ESCAPE ON ERROR ADDRESS
690	002272	112767	000001	176615	MOVB #1,\$SEMAX	;;ALLOW ONE ERROR PER TEST
691	002300	012767	002300	176600	MOV \$SLPADR	;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
692	002306	012767	002306	176574	MOV \$SLPERR	;;SETUP THE ERROR LOOP ADDRESS
693					;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS	
694					;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.	
695	002314	013746	000004		MOV \$ERRVEC,-(SP)	;;SAVE ERROR VECTOR
696	002320	012737	002354	000004	MOV \$64,\$ERRVEC	;;SET UP ERROR VECTOR
697	002326	012767	177570	176604	MOV \$DSWR,\$SWR	;;SETUP FOR A HARDWARE SWICH REGISTER
698	002334	012767	177570	176600	MOV \$DDISP,\$DISPLAY	;;AND A HARDWARE DISPLAY REGISTER
699	002342	022777	177777	176570	CMP #-1,\$DSWR	;;TRY TO REFERENCE HARDWARE SWR
700	002350	001012			BNE 66\$	;;BRANCH IF NO TIMEOUT TRAP OCCURRED
701						;;AND THE HARDWARE SWR IS NOT = -1
702	002352	000403			BR 65\$	;;BRANCH IF NO TIMEOUT
703	002354	012716	002362		64\$: MOV \$65\$,(SP)	;;SET UP FOR TRAP RETURN
704	002360	000002			RTI	
705	002362	012767	000176	176550	65\$: MOV \$SWREG,\$SWR	;;POINT TO SOFTWARE SWR
706	002370	012767	000174	176544	MOV \$DISPREG,\$DISPLAY	
707	002376	012637	000004		66\$: MOV (SP)+,\$ERRVEC	;;RESTORE ERROR VECTOR
708						
709	002402	005067	176632		CLR \$PASS	;;CLEAR PASS COUNT
710	002406	132767	000200	176637	BITB \$APTSIZE,\$ENVM	;;TEST USER SIZE UNDER APT
711	002414	001403			BEQ 67\$	;;YES,USE NON-APT SWITCH
712	002416	012767	001254	176514	MOV \$SSWREG,\$SWR	;;NO,USE APT SWITCH REGISTER
713	002424				67\$:	
714					.SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER	
715	002424	005737	000042		TST \$42	;;ARE WE RUNNING UNDER XXDP ACT?
716	002430	001012			BNE 68\$	;;BRANCH IF YES
717	002432	126727	176614	000001	CMPB \$ENV,#1	;;ARE WE RUNNING UNDER APT?
718	002440	001406			BEQ 68\$	;;BRANCH IF YES
719	002442	026727	176472	000176	CMP \$SWR,\$SWREG	;;SOFTWARE SWITCH REG SELECTED?
720	002450	001005			BNE 69\$	;;BRANCH IF NO
721	002452	104406			GTSWR	;;GET SOFT-SWR SETTINGS
722	002454	000403			BR 69\$	
723	002456	112767	000001	176450	68\$: MOVB #1,\$AUTOB	;;SET AUTO-MODE INDICATOR
724	002464				69\$:	

G11

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 19
 DZDMMB.P11 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0136

```

725 002464 012767 026566 175312 START1: MOV #BUSER,ERRVEC ;SET UP THE BUS ERROR VECTOR
726 002472 012767 000340 175306 MOV #340,ERRVEC+2 ;
727 002500 012767 026642 175302 MOV #RESERR,RESVEC ;SET UP THE RSVD INSTR VECTOR
728 002506 012767 000340 175275 MOV #340,RESVEC+2 ;
729 002514 005767 025516 TST TITFLG ;HAVE WE TYPED TITLE ONCE ?
730 002520 001012 BNE 1$ ;BR IF YES
731 002522 104401 TYPE ;GO TYPE PROGRAM TITLE
732 002524 034634 TITLE
733 002526 005167 025504 COM TITFLG ;SET FLAG - TYPE TITLE ONLY ONCE PER LOAD
734 002532 032777 000001 176400 BIT #BIT0,DSWR ;DO WE WANT TO AUTOSIZE?
735 002540 001002 BNE 1$ ;BRANCH IF NOT.
736 002542 004767 022330 JSR PC,AUTOSZ ;GO AUTOSIZE.
737 002546 005767 025116 1$: TST VCFLG ;START AT 200 ??
738 002552 001413 BEQ 11$ ;BR IF NOT
739 002554 032777 000001 176356 BIT #BIT0,DSWR ;ARE PARAMETERS TO BE INPUT MANUALLY?
740 002562 001003 BNE 9$ ;BRANCH IF YES.
741 002564 016700 025406 MOV ADRVEC,RO ;OTHERWISE, GET ADDRESSES BETWEEN VECTORS FROM AUTOSIZER
742 002570 000402 BR 10$
743 002572 004767 023250 9$: JSR PC,INPARA ;GO ASK FOR PARAMETERS
744 002576 005067 025066 10$: CLR VCFLG ;RE INIT VECTOR FLAG
745 002602 005700 11$: TST RO ;USE DEFAULT PARAMETERS ?
746 002604 001407 BEQ START2 ;BR IF YES
747 002606 022700 177777 CMP #-1,RO ;CHANGE DH SELECT PARAM ONLY ?
748 002612 001002 BNE 2$ ;BR IF NOT
749 002614 000167 023376 JMP INPAR3 ;GO ASK FOR SELECT PARAM.
750 002620 000167 023302 2$: JMP INPAR ;GO ASK FOR ALL PARAMETERS
751
752 002624 012767 027566 025364 START2: MOV #DHADTB-2,ADPTR ;GET POINTER TO ADDRESS TABLE
753 002632 012767 027626 025360 MOV #DHVCTB-2,VCPTR ;GET POINTER TO VECTOR TABLE
754 002640 012767 027670 025354 MOV #BRLVL-2,BRPTR ;GET POINTER TO BR LEVEL TABLE
755 002646 012767 177777 025334 MOV #-1,DHNUM ;START WITH DH #00
756 002654 012767 000001 024314 MOV #1,SELMSK ;SET UP DH11 BIT TEST MARKER
757
758 002662 005267 025322 RESTAT: INC DHNUM ;GENERATE DH11 DEV NUMBER
759 002666 062767 000002 025322 ADD #2,ADPTR ;UPDATE TABLE POINTERS
760 002674 062767 000002 025316 ADD #2,VCPTR
761 002702 062767 000002 025312 ADD #2,BRPTR
762 002710 036767 024262 024262 BIT SELMSK,DHSEL ;TEST FOR SELECTED DH11
763 002716 001004 BNE RSTRTA ;BR IF SELECTED FOR TEST
764 002720 006367 024252 REST1: ASL SELMSK ;SHIFT MARKER TO TEST NEXT DH11
765 002724 001737 BEQ START2 ;BR IF 16 TESTED - START OVER
766 002726 000755 BR RESTAT ;GO TEST IF THIS ONE SELECTED
767 002730 017767 025262 024234 RSTRTA: MOV @ADPTR,DHADR ;SET UP DH11 ADDRESS
768 002736 017767 025256 024230 MOV @VCPTR,DHVCT ;SET UP THE DH11 VECTOR ENTRY
769 002744 017767 025252 025234 MOV @BRPTR,DHRLVL ;GET BR LEVEL VALUES
770 002752 004567 021616 JSR R5,SUNUM ;GO SET DH NUMBER IN THE MESSAGE BUFFER
771 002756 030210 DHNUM
772 002760 034724 TITLE2+20
773 002762 104401 TYPE ;GO PRINT "TESTING DH11 #XX"
774 002764 034704 TITLE2
775 002766 004767 021274 JSR PC,LDTBF1 ;GO LOAD XMITTR OUTPUT BUFFER WITH
776 ;BINARY COUNT PATTERN
777 002772 012767 002772 176106 MOV #.,$LPADR ;INIT SCOPE LOOP RETURN

```

H11

MAINDE?-11-DZDMM-B
DZDMMB.P11 TI

MACY11 27(732) 23-SEP-76 14:33 PAGE 19
CHECK SSYN RESPONSE FROM ALL DH11 REGISTERS

SEQ 0137

003000 000004

```

*****
:TEST 1 CHECK SSYN RESPONSE FROM ALL DH11 REGISTERS
*****
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 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
		SH4	DEVICE RESPONDING L	E72-6
			DEVICE SELECTED H	E09-11
%				
	MOV	R1,R2		;COPY IT INTO R2
	MOV	R2,R5		;ALSO R5
	ADD	#20,R5		;R5 WILL TELL US WHEN WE'VE TESTED ALL 8
	MOV	ERRVEC, -(SP)		;SAVE BUS ERROR VECTOR
	MOV	#3\$,ERRVEC		;GO TO 3\$ IF REG FAILS TO RESPOND
	SUB	#2,R2		;SO WE START WITH FIRST REG
1\$:	ADD	#2,R2		;POINT TO A DH11 REGISTER
	CMP	R2,R5		;TESTED ALL EIGHT ?
	BEG	4\$		;BR IF YES
2\$:	TST	(R2)		;ACCESS DH11 REG ADDR
	BR	1\$		;BR WHEN ALL 8 ARE DONE
3\$:	JSR	PC,SUER1		;GO SET UP ERROR INFO
	CMP	(SP)+,(SP)+		;FIX SP BECAUSE OF TRAP

819	003002	010102		
820	003004	010205		
821	003006	062705	000020	
822	003012	016746	174766	
823	003016	012767	003044	174760
824	003024	162702	000002	
825				
826	003030	062702	000002	
827	003034	020205		
828	003036	001412		
829	003040	005712		
830	003042	000772		
831				
832	003044	004767	021240	
833	003050	022626		

I11

MAINDEC-11-DZDMM-B
DZDMMB.P11 T1

MACY11 27(732) 23-SEP-76 14:33 PAGE 20
CHECK SSYN RESPONSE FROM ALL DH11 REGISTERS

SEQ 0138

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

J11

MAINDEC-11-DZDMM-B
DZDMMB.P11 T2

MACY11 27(732) 23-SEP-76 14:33 PAGE 21
TEST THAT "MASTER CLR" CAN CLEAR THE "SCR", "LPR", "BKR", AND "SSR" REGS

SEQ 0139

839
840
841
842 003070 000004
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894

::*****
:TEST 2 TEST THAT "MASTER CLR" CAN CLEAR THE "SCR", "LPR", "BKR", AND "SSR" REGS
:*****

TEST2: 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

%

MOV #MSTCLR, R5 ;GET POINTER TO ADDRESS TABLE
MOV R1, (R5)+ ;SET UP THE TEST ADDRESS TABLE SO THAT
MOV R1, (R5)+ ;IT CONTAINS THE ADDRESSES OF THE
MOV R1, (R5)+ ;SCR, LPR, BKR, AND SSR REGISTERS
MOV R1, (R5)+
ADD #SSR, -(R5) ;GENERATE SSR ADDRESS
ADD #BKR, -(R5) ;GENERATE BKR ADDRESS



K11

MAINDEC-11-DZDMM-B
DZDMMB.P11 T2MACY11 27(732) 23-SEP-76 14:33 PAGE 22
TEST THAT "MASTER CLR" CAN CLEAR THE "SCR", "LPR", "BKR", AND "SSR" REGS

SEQ 0140

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			SEQ	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

911
912
913
914 003170 000004
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966

:TEST 3 TEST "SCR" REG R/W BITS CAN SET/CLR (NORMAL MODE)

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	DUI.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	

%

M11

 MAINDEC-11-DZDMM-B
 DZDMMB.P11 T3

 MACY11 27(732) 23-SEP-76 14:33 PAGE 24
 TEST "SCR" REG R/W BITS CAN SET/CLR (NORMAL MODE)

SEQ 0142

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

N11

MAINDEC-11-DZOHM-B
DZOHMB.P11 T4MACY11 27(732) 23-SEP-76 14:33 PAGE 25
TEST "SCR" REG. READ ONLY BITS (NORMAL MODE)

SEQ 0143

998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053

003300 000004

```

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

```

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

ERRORS:

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

SYNC:

M7277 SH4 LOAD SSR LOW BYTE H CR1

DEBUG:

SAME AS FOR TEST 03

KEY LOGIC:

SAME AS FOR TEST 03

```

%
      MOV      R1,R2          ;MAKE IT THE REG. ADDR ALSO
      MOV      #1,R5          ;INIT BIT TEST MARKER
1$:   BIT      R5,RGMSK2      ;IS IT A READ ONLY BIT ?
      BNE      3$             ;BR IF IT IS - GO TEST IT
2$:   ASL      R5             ;SHIFT BIT MARKER
      BEQ      TST5           ;BR IF DONE ALL BITS
      BR       1$            ;GO TEST THIS BIT

3$:   CLR      R4             ;RESULT S/B = 000000
      CLR      (R2)           ;INIT REG BEING TESTED
      MOVB     #0,SSR(R1)     ;SCOPE SYNC
      MOV      R5,(R2)        ;ATTEMPT TO SET A READ ONLY BIT
      MOV      (R2),R3        ;GET THE WAS DATA
      BEQ      2$            ;BR IF THE BIT DIDN'T SET

      JSR      PC,SUER2       ;GO SET UP ERROR INFO
      MOV      #3$,$LPERR     ;SET UP ERROR LOOP RETURN ADDR
      ERROR    2              ;READ ONLY BIT SET IN "SCR"
      BR       2$            ;CONTINUE WITH NEXT BIT

```

000001
024240

000000 000016

021000
003324 175532


```

*****
:TEST 5          TEST "SCR" REG. BITS THAT CAN BE SET/CLR IN MAINT. MODE
:*****
TSTS:  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 (SCRC9=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 21 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:

- ### 1.2 STEP 3 FAILURE TO SET WITH MAINT. MODE SET

M7277 SH4 LOAD SSR LOW BYTE H CRI

- ## 2.0 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:

[illegible]

				SAME AS TEST. 03 WITH THE FOLLOWING ADDITION		
				M7289	SH4	74121 ONE-SHOTS E35-6, E23-6
11110						
11111						
11112						
11113						
11114						
11115	003364	012767	003414 175516	MOV	#3\$, \$LPERR	:SET UP THE ERROR LOOP RETURN
11116	003372	010102		MOV	R1, R2	:MAKE IT REG ADDR TOC
11117	003374	012705	000001	MOV	#1, R5	:INIT BIT TEST MARKER
11118	003400	030567	024160	1\$: BIT	R5, RGMSK6	:IS IT A READ ONLY BIT ??
11119	003404	001003		BNE	3\$	:BR IF YES - TEST IT
11120	003406	006305		2\$: ASL	R5	:SHIFT THE BIT MARKER
11121	003410	001457		BEQ	TST6	:BR IF DONE ALL SELECTED BITS
11122	003412	000772		3\$: SP	1\$	:GO TEST FOR THIS BIT
11123						
11124	003414	010504		3\$: MOV	R5, R4	:SET UP S/B DATA
11125	003416	052704	001000	BIS	#BIT09, R4	:PUT IN THE MAINT. BIT
11126	003422	005012		CLR	(R2)	:INIT REG BEING TESTED
11127	003424	052712	001000	BIS	#BIT09, (R2)	:TURN ON MAINT. MODE
11128	003430	112761	000000 000016	MOVB	#0, SSR(R1)	:SCOPE SYNC
11129	003436	050512		BIS	R5, (R2)	:SET THE SELECTED BIT
11130	003440	011203		MOV	(R2), R3	:GET THE WAS DATA
11131	003442	020304		CMP	R3, R4	:DID SELECTED BIT GET SET ??
11132	003444	001404		BEQ	4\$	:BR IF IT DID
11133						
11134	003446	004767	020676	JSR	PC, SUER2	:GO SET UP ERROR INFO
11135	003452	104002		ERROR	2	:SELECTED BIT FAILED TO SET IN MAINT MODE
11136	003454	000754		BR	2\$	:GO TEST NEXT BIT
11137						
11138	003456	042712	001000	4\$: BIC	#BIT09, (R2)	:TURN OFF MAINT. MODE
11139	003462	042704	001000	BIC	#BIT09, R4	:CLR MAINT BIT IN S/B DATA
11140	003466	112761	000000 000017	MOVB	#0, SSR+1(R1)	:SCOPE SYNC
11141	003474	040512		BIC	R5, (R2)	:ATTEMPT TO CLR SELECTED BIT
11142	003476	011203		MOV	(R2), R3	:GET THE WAS DATA
11143	003500	020304		CMP	R3, R4	:DID BIT GET CLEARED ??
11144	003502	001404		BEQ	5\$	:BR IF IT DIDN'T
11145						
11146	003504	004767	020640	JSR	PC, SUER2	:GO SET UP ERROR INFO
11147	003510	104002		ERROR	2	:SELECTED BIT GOT CLEARED WITH MAINT MODE OFF
11148	003512	000735		BR	2\$	:GO TEST NEXT BIT
11149						
11150	003514	012704	001000	5\$: MOV	#BIT09, R4	:SET UP S/B DATA
11151	003520	050412		BIS	R4, (R2)	:SET MAINT. MODE
11152	003522	012761	000000 000004	MOV	#0, LPR(R1)	:SCOPE SYNC
11153	003530	040512		BIC	R5, (R2)	:NOW CLR SELECTED BIT
11154	003532	011203		MOV	(R2), R3	:GET THE WAS DATA
11155	003534	020304		CMP	R3, R4	:DID BIT GET CLEARED OK ??
11156	003536	001723		BEQ	2\$	:BR IF YES
11157						
11158	003540	004767	020604	JSR	PC, SUER2	:GO SET UP ERROR INFO
11159	003544	104002		ERROR	2	:FAILED TO CLR SELECTED BIT IN MAINT MODE
11160	003546	000717		BR	2\$	:GO SELECT NEXT BIT FOR TEST

003550 000004

```

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

```

THIS TEST VERIFIES THAT ALL R/W BITS IN THE "LPR" REGISTER 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)
        SH5.6.7.9  OUTPUT MUX CHIPS (74151'S PIN 2)

```

```

003552 012767 003606 175330
003560 010102
003562 062702 000004

```

```

%
MOV    #35,$LPERR      ;SET UP THE ERROR LOOP RETURN
MOV    R1,R2           ;COPY IT IN R2
ADD     #LPR,R2         ;GENERATE REGADR IN R2

```

E12

 MAINDEC-11-DZDMM-6
 DZDMMB.P11 T6

 MACY11 27(732) 23-SEP-76 14:33 PAGE 29
 TEST THAT ALL R W BITS IN "LPR" CAN BE SET CLR

SEG 0147

1231	003566	012705	000001		MOV	#1,R5	:INIT BIT TEST MARKER
1232	003572	030567	023760		BIT	R5,RGMSK3	:TEST THIS BIT ??
1233	003576	001003		1\$:	BNE	3\$	:BR IF YES
1234	003600	006305		2\$:	ASL	R5	:SHIFT THE MARKER
1235	003602	001430			BEQ	TST7	:BR IF DONE ALL BITS
1236	003604	000772			BR	1\$	:GO TEST NXT BIT
1237							
1238	003606	010504		3\$:	MOV	R5,R4	:SET UP S/B DATA
1239	003610	005012			CLR	(R2)	:INIT REG BEING TESTED
1240	003612	112761	000000 000016		MOVB	#0,SSR,R1	:SCOPE SYNC
1241	003620	010512			MOV	R5,(R2)	:SET LPR BIT
1242	003622	011133			MOV	(R2),R3	:GET THE WAS DATA
1243	003624	020304			CMP	R3,R4	:DID IT SET
1244	003626	001403			BEQ	4\$	:BR IF IT SET PROPERLY
1245							
1246	003630	004767	020514		JSR	PC,SUER2	:GO SET UP ERROR INFO
1247	003634	104004			ERROR	4	:LPR BIT FAILED TO SET PROPERLY
1248							
1249	003636	005004		4\$:	CLR	R4	:GET READY TO CLEAR SELECTED BIT
1250	003640	112761	000000 000017		MOVB	#0,SSR+1(R1)	:SCOPE SYNC
1251	003646	040512			BIC	R5,(R2)	:CLEAR THE BIT
1252	003650	011203			MOV	(R2),R3	:GET THE WAS DATA
1253	003652	001752			BEQ	2\$	:BR IF BIT CLEARED PROPERLY
1254							
1255	003654	004767	020470		JSR	PC,SUER2	:GO SET UP ERROR INFO
1256	003660	104004			ERROR	4	:LPR BIT FAILED TO CLEAR PROPERLY
1257	003662	000746			BR	2\$	:GO SELECT NEXT BIT

1244
1245
1246
1247 003664 000004
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298 003666 012767 003722 175214
1299 003674 010102

```

*****
: *TEST 7      TEST THAT ALL R/W BITS IN "BKR" CAN BE SET/CLR
*****

```

```

TEST:  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 51 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 FOR THE PREVIOUS TESTS ARE THE ACTUAL REGISTER CHIPS AND THE INPUT SELECTED ON THE OUTPUT DATA MULTIPLEXORS.

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

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

KEY LOGIC:

```

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

```

```

M7278 SH5 - SH8 74175 REGISTER CHIPS (E51,E38,E67,E60)
      SH5 - SH8 74151'S MUX CHIPS INPUT PIN 13

```

%

```

MOV  #3$,SLPERR      :SET UP THE ERROR LOOP RETURN
MOV  R1,R2           :GENERATE "BKR" ADDRESS IN R2

```

G12

MAINDEC-11-DZDMM-B
DZDMMB.P11 T7MAY11 27(732) 23-SEP-76 14:33 PAGE 31
TEST THAT ALL R/W BITS IN "BKR" CAN BE SET/CLR

SEQ 0149

1300	003576	052702	000014		ADD	#BKR,R2	
1301	003702	012705	000001		MOV	#1,R5	:INIT BIT TEST MARKER
1302	003706	030567	023646		BIT	R5,RGMSK4	:TEST THIS BIT ??
1303	003712	001003		1\$:	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)	:SCOPE SYNC
1321	003752	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

H12

MAINDEC-11-DZDMM-B
DZDMMB.P11 T10

MACY11 27(732) 23-SEP-76 14:33 PAGE 32
TEST THAT ALL R/W BITS IN "SSR" CAN BE SET/CLR

SEQ 0150

1328
1329
1330
1331 004000 000004
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383

*TEST 10 TEST THAT ALL P/W BITS IN "SSR" CAN BE SET/CLR

*ST10: 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 REPOCRT 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 FUI

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	DUI,DU2,DT2
M7279	SH2	SSR <13:08> H	(E20 AND E24)
M7278	SH5 - SH8	REGISTER CHIPS E53,E68, OR E69 (74175'S)	
		OUTPUT MUX CHIPS - (74151'S	PIN 12)

%

MAINDEC-11-DZDMM-B
DZDMMB.P11 T10

MACY11 27(732) 23-SEP-76 14:33 PAGE 33
TEST THAT ALL R/W BITS IN "SSR" CAN BE SET/CLR

SEQ 0151

1384	004002	012767	004036	175100	MOV	#3\$,SLPERR	;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	E	;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

1417
1418
1419
1420 004126 000004
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461 004130 012767 004164 174752
1462 004136 010102
1463 004140 062702 000004
1464 004144 012705 000001
1465 004150 030567 023402
1466 004154 001003
1467 004156 006305
1468 004160 001436
1469 004162 000772
1470
1471 004164 016704 023366
1472 004170 005012

:TEST 11 TEST THAT CLR/SET OF BIT "N" IN "LPR" DOES NOT CLEAR ANY OTHER BITS

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)

%

	MOV	#3\$, \$LPERR	:SET UP THE ERROR LOOP RETURN
	MOV	R1, R2	:SET UP THE REG ADDR
	ADD	#LPR, R2	
	MOV	#1, R5	:INIT BIT TEST MASK
1\$:	BIT	R5, RGMSK3	:TEST THIS BIT ??
	BNE	3\$	:BR IF YES
2\$:	ASL	R5	:SHIFT THE BIT TEST MASK
	BEQ	TST12	:BR IF TESTED ALL BITS
	BR	1\$	:GO TEST THIS BIT
3\$:	MOV	RGMSK3, R4	:SET UP S/B DATA
	CLR	(R2)	:INIT REG BEING TESTED

K12

MAINDEC-11-DZDMM-B
DZDMMB.P11 T11

MACY11 27(732) 23-SEP-76 14:33 PAGE 35
TEST THAT CLR/SET OF BIT "N" IN "LPR" DOES NOT CLEAR ANY OTHER BITS

SEQ 0153

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			SEQ	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			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			SEQ	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

L12

MAINDEC-11-DZDMM-B
DZDMMB.P11 T12

MACY11 27(732) 23-SEP-76 14:33 PAGE 36
TEST THAT CLR/SET OF BIT "N" IN "BKR" DOES NOT CLEAR ANY OTHER BITS

SEQ 0154

1495
1496
1497
1498 004256 000004
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550

:TEST 12 TEST THAT CLR/SET OF BIT "N" IN "BKR" DOES NOT CLEAR ANY OTHER BITS

TEST12: 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)

%

MOV	#3\$, \$LPERR	:SET UP THE ERROR LOOP RETURN
MOV	R1, R2	:SET UP THE REG ADDR
ADD	#BKR, R2	
MOV	#1, R5	:INIT BIT TEST MASK
1\$: BIT	R5, RGMSK4	:TEST THIS BIT "
BNE	3\$	:BR IF YES
2\$: ASL	R5	:SHIFT THE BIT TEST MASK
BEQ	TST13	:BR IF TESTED ALL BITS
BR	1\$	:GO TEST THIS BIT
3\$: MOV	RGMSK4, R4	:SET UP S/B DATA
BIC	R5, R4	:CLR BIT "N"

004260	012767	004314	174622
004266	010102		
004270	062702	000014	
004274	012705	000001	
004300	030567	023254	
004304	001003		
004306	006305		
004310	001436		
004312	000772		
004314	016704	023240	
004320	040504		

M12

MAINDEC-11-DZDMM-B
DZDHMB.P11 T12

MACY11 27(732) 23-SEP-76 14:33 PAGE 37
TEST THAT CLR SET OF BIT "N" IN "BKR" DOES NOT CLEAR ANY OTHER BITS

SEQ 0155

1551	004322	005012		CLR	(R2)	; INIT REG BEING TESTED
1552	004324	016712	023230	MOV	RGMSK4, (R2)	; SET ALL R/W BITS IN PKR
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		BIS	R5, R4	; SET BIT "N" IN S/B DATA
1564	004360	112761	000000 000017 4\$:	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

N12

MAINDEC-11-DZDMM-B
DZDMMB.P11 T13

MACY11 27(732) 23-SEP-76 14:33 PAGE 39
TEST THAT CLR/SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER BITS

SEQ 0156

```

1573      ;*****
1574      ;*TEST 13      TEST THAT CLR/SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER BITS
1575      ;*****
1576      004406  000004  TST13: SCOPE
1577      .REM      %
1578      TEST ABSTRACT:
1579      *****
1580

```

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

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

ANY ERROR DETECTED IS REORTED 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 FUI

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)

%

```

1617      004410  012767  004444  174472  MOV      #3$, $LPERR      ;SET UP THE ERROR LOOP RETURN
1618      004416  010102                MOV      R1, R2          ;SET UP THE PEG ADDR
1619      004420  062702  000016                ADD      #SSR, R2
1620      004424  012705  000001                MOV      #1, R5          ;INIT BIT TEST MASK
1621      004430  030567  023126  1$: BIT      R5, RGMSK5        ;TEST THIS BIT ??
1622      004434  001003                BNE      3$             ;BR IF YES
1623      004436  006305  2$: ASL      R5             ;SHIFT THE BIT TEST MASK
1624      004440  001442                BEQ      TST14          ;BR IF TESTED ALL BITS
1625      004442  000772                BR       1$             ;GO TEST THIS BIT
1626
1627      004444  016704  023112  3$: MOV      RGMSK5, R4        ;SET UP S/B DATA
1628      004450  040504                BIC      R5, R4          ;CLR BIT "N"

```

B13

MAINDEC-11-D2DMM-B
D2DMMB.P11 T13

MACY11 27(732) 23-SEP-76 14:33 PAGE 39
TEST THAT CLR SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER BITS

SEG 0157

1629	004452	005012		CLR	(R2)	:INIT REG BEING TESTED
1630	004454	016712	023102	MOV	RGMSYS, (R2)	:SET ALL R/W BITS IN SSR
1631	004460	012761	000000 000004	MOV	#0, LPR(R1)	:SCOPE SYNC
1632	004466	040513		BIC	R5, (R2)	:CLEAR BIT "N" IN SSR
1633	004470	011203		MOV	(R2), R3	:GET THE WAS DATA
1634	004472	042703	077700	BIC	#77700, R3	:CLEAR JUNK BITS
1635	004476	020304		COMP	R3, R4	:DID IT CLEAR OK ?
1636	004500	001404		BEG	45	:BR IF YES
1637	004502	004767	017642	JSR	PC, SJER2	:GO SET UP ERROR INFO
1638	004506	104006		ERROR	6	:BIT "N" FAILED TO CLR PROPERLY
1639	004510	000752		BR	25	:GO TEST NEXT BIT
1640	004512	050504		BIS	R5, R4	:SET BIT "N" IN S/B DATA
1641	004514	012761	000000 000014 45:	MOV	#0, BKR(R1)	:SCOPE SYNC
1642	004516	050512		BIS	R5, (R2)	:SET BIT "N" IN SSR
1643	004518	011203		MOV	(R2), R3	:GET THE WAS DATA
1644	004519	042703	077700	BIC	#77700, R3	:CLEAR JUNK BITS
1645	004520	020304		COMP	R3, R4	:DID BIT "N" SET PROPERLY ?
1646	004524	001740		BEG	25	:BR IF YES
1647	004526	004767	017606	JSR	PC, SUEF2	:GO SET UP ERROR INFO
1648	004528	104006		ERROR	6	:BIT "N" FAILED TO SET PROPERLY
1649	004530	000752		BR	25	:GO SELECT NEXT BIT

004546 000004

```

*****
:TEST 14 "CAR" MEMORY ADDRESSING TEST
*****
:ST14: SCOPE
:REM
:TEST ABSTRACT:
*****

```

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

MEMORY PATTERN:	LOCATION	CONTENTS	(BOTH OCTAL)
	00	000000	
	01	010421	
	02	021042	
	03	031463	
	04	042104	
	05	052525	
	06	063146	
	07	073567	
	10	104210	
	11	114631	
	12	125252	
	13	135673	
	14	146314	
	15	156735	
	16	167356	
	17	177777	

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

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

ERRORS:

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

SYNC:

1.) WRITE SYNC: M7277 SH4 LOAD SSR LOW BYTE H CR1
2.) READ SYNC: M7277 SH4 LOAD SSR HIGH BYTE H CR2

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 INTER 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) DUI,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,E05 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)

```

%
MOV R1,R2      ;COPY IT IN R2
ADD #CAR,R2    ;SET UP REGADR IN R2
MOV LINSSEL,STMP7 ;SAVE LINE SELECT PARAMETER
MOV #-1,LINSSEL ;DO ALL LINES FOR THIS TEST
JSR PC,SELIN   ;GO SELECT A LINE NO.
BR 3$          ;BR IF DONE ALL SELECTED LINES
TSTB LINE     ;DOING LINE 00 ?
BNE 2$        ;BR IF NOT
CLR R4        ;INIT TEST DATA

2$: BISB LINE,(R1) ;SELECT A LINE
    MOVB #0,SSR(R1) ;SCOPE SYNC
    MOV R4,(R2)     ;LOAD THE CAR REG.

```

```

1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752 004550 010102
1753 004552 062702 000006
1754 004556 016767 022420 174434
1755 004564 012767 177777 022410
1756 004572 004767 017704 1$:
1757 004576 000415
1758 004600 105767 023406
1759 004604 001001
1760 004606 005004
1761
1762 004610 156711 023376 2$:
1763 004614 112761 000000 000016
1764 004622 010412

```


E10

MAINDEC-11-DZOHM-B
DZOHMB.P11

T14

MACY11 27(732) 23-SEP-76 14:33 PAGE 42
"CAR" MEMORY ADDRESSING TEST

SEQ 0160

1765	004624	062704	010421			ADD	#10421,R4	:GENERATE NEW DATA
1766	004630	000760				BR	1\$	:GO DO NEXT LINE
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				SNE	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	004652	011203				MOV	(R2),R3	:GET CONTENTS OF CAR
1777	004654	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	RS,SUNUM	:SET UP LINE NO. IN MSG BUFFER
1782	004700	030212				LINE		
1783	004702	031045				EM7+47		
1784	004704	012767	004722	174176		MOV	#6\$,SLPERR	: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

1794
1795
1796
1797 004736 000004
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849

:TEST 15 "BCR" MEMORY ADDRESSING TEST

:ST15: 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 EACH LOCATION
(00 - 17(8)) IN EACH CHIP. THAT IS: LOC00 = 00, LOC 01 = 01,
.....LOC 17 = 17.

MEMORY PATTERN: LOCATION	CONTENTS	(BCTH 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 M7279.

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,E28
          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)

%      MOV R1,R2          ;COPY IT IN R2
      ADD #BCR,R2         ;SET UP REGADR IN R2
      MOV LINSSEL,$TMP7   ;SAVE LINE SELECT PARAMETER
      MOV #-1,LINSSEL     ;DO ALL LINES FOR THIS TEST
      JSR PC,SELINE       ;GO SELECT A LINE NO.
      BR 3$               ;BR IF DONE ALL SELECTED LINES
      TSTB LINE           ;DOING LINE 00 ?
      BNE 2$              ;BR IF NOT
      CLR R4              ;INIT TEST DATA

      BISB LINE,(R1)       ;SELECT A LINE
      MOVB #0,$SR(R1)     ;SCOPE SYNC
  
```

```

1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894 004740 010102
1895 004742 062702 000010
1896 004746 016767 022230 174244
1897 004754 012767 177777 022220
1898 004762 004767 017514
1899 004766 000415
1900 004770 105767 023216
1901 004774 001001
1902 004776 005004
1903
1904 005000 156711 023206
1905 005004 112761 000000 000016
  
```

H13

MAINDEC-11-DZDMM-B
DZDMMB.P11 T15MACY11 27(732) 23-SEP-76 14:33 PAGE 45
"BCR" MEMORY ADDRESSING TEST

SEQ 0163

1906	005012	010412			MOV	R4, (R2)	: LOAD THE BCR REG.
1907	005014	062704	010421		ADD	#10421, R4	: GENERATE NEW DATA
1909	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

MAINDEC-11-DZDMM-B
DZDMMB.P11 T16

MACY11 27(732) 23-SEP-76 14:33 PAGE 46
"CAR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES

SEQ 0164

1936
1937
1938
1939 005126 000004
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991

::*****
:TEST 16 "CAR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
:*****

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)

```
%
MOV R1,R2 ;COPY IT INTO R2
ADD #CAR,R2 ;R2 GETS CAR ADDRESS
1$: JSR PC,SELINE ;GO SELECT A LINE NO.
BR TST17 ;BR IF DONE ALL SELECTED LINES
MOV #-1,R4 ;RESULT IN CAR S/B = 177777
BISB LINE,(R1) ;SELECT A LINE NO.
JSR R5,SUNUM ;GO SET UP LINE NO. IN MSG BUFFER
LINE
EM7+47

MOV B #0,SSR(R1) ;SCOPE SYNC
MOV R4,(R2) ;LOAD A CAR WITH ALL ONES
MOV (R2),R3 ;GET THE WAS DATA FROM THE CAR
CMP R4,R3 ;DID IT CONTAIN ALL ONES "
```

005130 010102
005132 062702 000006
005136 004767 017340
005142 000443
005144 C:2704 177777
005150 156711 023036
005154 004567 017414
005160 030212
005162 031045
005164 112761 000000 000016 2\$:
005172 010412
005174 011203
005176 020403

J13

MAINDEC-11-DZDMM-8
DZDHMB.P11 T16MACY11 27(732) 23-SEP-76 14:33 PAGE 47
"CAR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES

SEQ 0165

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		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

K13

MAINDEC-11-DZDMM-B
DZDMMB.P11 T17MACY11 27(732) 23-SEP-76 14:33 PAGE 48
"BCR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES

SEQ 0166

2008
2009
2010
2011 005252 000004
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063

```

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

```

```

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)
*****

```

```

%
      MOV      R1,R2      ;COPY IT INTO R2
      ADD      #BCR,R2    ;R2 GETS BCR ADDRESS
1$:   JSR      PC,SELINE  ;GO SELECT A LINE NO.
      BR       TST20      ;BR IF DONE ALL SELECTED LINES
      MOV      #-1,R4     ;RESULT IN BCR S/B = 177777
      BISB     LINE,(R1)  ;SELECT A LINE NO.
      JSR      R5,SUNUM   ;GO SET UP LINE NO. IN MSG BUFFER
      LINE     EM10+44
%
2$:   MOVB     #0,SSR(R1) ;SCOPE SYNC
      MOV      R4,(R2)    ;LOAD A BCR WITH ALL ONES
      MOV      (R2),R3    ;GET THE WAS DATA FROM THE BCR
      CMP      R4,R3      ;DID IT CONTAIN ALL ONES

```

000000 000016

L13

MAINDEC-11-DZDMM-B
DZDMMB.P11 T17MACY11 27(732) 23-SEP-76 14:33 PAGE 49
"BCR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES

SEQ 0167

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		CLR	R4	:RESULT IN BCR S/B = 000000
2071	005344	112761	000000 000017 3\$:	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

2080
2081
2082
2083 005376 000004
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135

```

::*****
;TEST 20      "CAR" MEMORY PATTERNS TEST / O'S DISTURB
::*****
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 PARAMETER "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.) IERROR 461 IS CALLED TO REPORT ANY ERROR DETECTED. 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.) 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	#PATRMA,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,SELIN	;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

N13

MAINDEC-11-DZDHM-B
DZDHMB.P11 T20MACY11 27(732) 23-SEP-76 14:53 PAGE 51
"CAR" MEMORY PATTERNS TEST / O'S DISTURB

SEQ 0169

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, SJER4	; GO SET UP ERROR IN FO
2158	005546	004567	017022		JSR	PS, SUNUM	; GO SET UP LINE NO. IN MSG BUFFER
2159	005552	001206			\$TMP2		
2160	005554	033566			EM46+63		
2161	005556	012767	005434 173324		MOV	#2\$, \$LPERR	; SET UP ERROR LOOP RETURN
2162	005564	104046			EKROP	4\$	; 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

005634 000034

```

*****
:TEST 21 "BCR" MEMORY PATTERNS TEST / O'S DISTURB
*****
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 PARAMETER "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 471 IS CALLED TO REPORT ANY ERROR DETECTED. 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.) 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	#PATANA,R5	;SET UP POINTER TO DATA PATTERNS
005620	012567	173360	15: MOV	(R5)+,S1MP1	;GET A DATA TEST PATTERN
005624	001472		BEQ	TST22	;BR IF DONE THREE PATTERNS
005626	004767	016650	115: JSR	PC,SELINE	;GO SELECT A LINE TO TEST
005632	000772		BR	15	;BR IF SELECTED ALL LINES
005634	116767	022352 022352	MOV	B,LINEA	;SAVE THE LINE NO. FOR ERROR LOOP
005642	105067	173342	25: CLRB	S1MP3	;INIT LINE COUNTER

MAINDEC-11-DZDMM-B
 DZDMMB.P11 T21

 MACY11 27(732) 23-SEP-76 14:33 PAGE 53
 "BCR" MEMORY PATTERNS TEST O'S DISTURB

SEQ 0171

22324	005646	116711	173336	35:	MOVB	\$TMP3, (R1)	: SELECT A LINE TO CLEAR
22325	005652	005012			CLR	(R2)	: CLR BCR FOR THAT LINE
22326	005654	105267	173330		INCB	\$TMP3	: GENERATE NEW LINE NO.
22327	005660	126727	173324	000020	CMPB	\$TMP3, #20	: DONE CLEARING ALL LINES ?
22328	005666	001357			BNE	35	: BR IF NOT
22329							
22330	005670	116711	022320		MOVB	LINEA, (R1)	: SET LINE SELECT BITS
22331	005674	112761	000000	000016	MOVB	#0, SSR(R1)	: SCOPE SYNC
22332	005702	016712	173276		MOV	\$TMP1, (R2)	: LOAD BCR WITH TEST PATTERN
22333							
22334	005706	105067	173274		CLRB	\$TMP2	: INIT A LINE COUNTER
22335	005712	016704	173266	45:	MOV	\$TMP1, R4	: SET UP S/B DATA
22336	005716	116711	173264		MOVB	\$TMP2, (R1)	: SELECT A LINE TO CHECK
22337	005722	112761	000000	000017	MOVB	#0, SSR+1(R1)	: SCOPE SYNC
22338	005730	011203			MOV	(R2), R3	: GET WAS DATA
22339	005732	126767	173250	022252	CMPB	\$TMP2, LINE	: IS THIS THE LINE WITH THE TEST PATTERN
22340	005740	001401			BEQ	55	: BR IF IT IS
22341	005742	005004			CLR	R4	: MAKE S/B DATA = 000000
22342	005744	020304		55:	CMP	R3, R4	: CORRECT DATA IN BCR ?
22343	005746	001412			BEQ	65	: BR IF YES
22344							
22345	005750	004767	016462		JSR	PC, SUER4	: GO SET UP ERROR IN FO
22346	005754	004567	016614		JSR	R5, SUNUM	: GO SET UP LINE NO. IN MSG BUFFER
22347	005760	001206			\$TMP2		
22348	005762	034066			EM47+56		
22349	005764	012767	005642	173116	MOV	#25, \$LPERP	: SET UP ERROR LOOP RETURN
22350	005772	104047			ERROR	47	: INCORRECT DATA READ FROM BCR
22351							
22352	005774	105267	173206	65:	INCB	\$TMP2	: GENERATE NEXT LINE NO.
22353	006000	122767	000020	173200	CMPB	#20, \$TMP2	: DONE ALL LINES ?
22354	006006	001707			BEQ	115	: BR IF YES
22355	006010	000740			BR	45	: GO CHECK NEXT LINE

107

006012 000004

```

*****
:TEST 22      "CAR" MEMORY PATTERNS TEST / 1'S DISTURB
*****
:ST22: 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)

2299	006014	010102			%	MOV	R1,R2	;SET UP REGADR
2300	006016	062702	000006			ADD	#CAR,R2	
2301	006022	C12705	177777			MOV	#-1,R5	;TEST PATERN IN R5 = 177777
2302	006026	010567	173152			MOV	R5,\$TMP1	;SAVE FOR ERROR REPORTING
2303	006032	004767	016444		15:	JSR	PC,SELINE	;GO SELECT A LINE TO TEST
2304	006036	000465				BR	TS123	;BR IF DONE ALL LINES
2305	006040	116767	022146	022146		MOVB	LINE,LINEA	;SAVE THE LINE NO. FOR ERROR LOOP
2306								
2307	006046	105067	173136		25:	CLRB	\$TMP3	;INIT LINE COUNTER
2308	006052	116711	173132		35:	MOVB	\$TMP3,(R1)	;SELECT A LINE TO CLEAR
2309	006056	010512				MOV	R5,(R2)	;LOAD CAR WITH 177777
2310	006060	105267	173124			INCB	\$TMP3	;GENERATE NEW LINE NO.
2311	006064	126727	173120	000020		CMPB	\$TMP3,#20	;DONE SETTING ALL LINES TO 177777

E14

MAINDEC-11-DZDMM-B
DZDMMB.P11

T22

MACY11 27(732) 23-SEP-76 14:33 PAGE 55
"CAR" MEMORY PATTERNS TEST / 1'S DISTURB

SEQ 0173

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			CLR	R4	4\$:	;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), F3		;GET WAS DATA
2323	006132	126767	173050	022052	CMPB	\$TMP2, LINE		;IS THIS THE LINE WITH THE TEST PATTERN
2324	006140	001401			SEQ	5\$		;BR IF IT IS
2325	006142	010504			MOV	R5, R4		;MAKE S/B DATA = 177777
2326	006144	020304			CMP	R3, R4	5\$:	;CORRECT DATA IN CAR ?
2327	006146	001412			BEQ	6\$		;BR IF YES
2328								
2329	006150	004767	016262		JSR	PC, SJERR4		;GO SET UP ERROR IN FO
2330	006154	004567	016414		JSF	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		INCB	\$TMP2	6\$:	;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	020741			BP	4\$		;GO CHECK NEXT LINE

2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395

006212 000004

```

*****
:TEST 23      "BCR" MEMORY PATTERNS TEST / 1'S DISTURB
*****
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 "LINSSEL" 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)

%

2383	006214	010102			MOV	R1,R2	;SET UP REGADR
2384	006216	062702	000010		ADD	#BCR,R2	
2385	006222	012705	177777		MOV	#-1,R5	;TEST PATERN IN R5 = 177777
2386	006226	010567	172752		MOV	R5,\$TMP1	;SAVE IT FOR ERROR REPORTING
2387	006232	004767	016244	1\$:	JSR	PC,SELIN	;GO SELECT A LINE TO TEST
2388	006236	000465			BR	TST24	;BR IF DONE ALL LINES
2389	006240	116767	021746 021746		MOVB	LINE,LINEA	;SAVE THE LINE NO.
2391	006246	105067	172736	2\$:	CLRB	\$TMP3	;INIT LINE COUNTER
2392	006252	116711	172732	3\$:	MOVB	\$TMP3,(R1)	;SELECT A LINE TO INIT
2393	006256	010512			MOV	R5,(R2)	;LOAD BCR WITH 177777
2394	006260	105267	172724		INCB	\$TMP3	;GENERATE NEW LINE NO.
2395	006264	126727	172720 000020		CPB	\$TMP3,#20	;DONE SETTING ALL LINES TO 177777

G14

MAINDEC-11-DZDHM-8
DZDHMB.P11 T23MACY11 27(722) 23-SEP-76 14:33 PAGE 57
"BCR" MEMORY PATTERNS TEST / 1'S DISTURB

SEQ 0175

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			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			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 FO
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		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

H14

MAINDEC-11-DZDMM-B
DZDMMB.P11 T24

MACY11 27(732) 23-SEP-76 14:33 PAGE 58
TEST THAT "CAR" MEMORY EXT BITS SET/CLR PROPERLY

SEQ 0176

006412 000004

:TEST 24 TEST THAT "CAR" MEMORY EXT BITS SET/CLR PROPERLY

+ST24: 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

424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479

SSR06 H

CH1

M7278 SH7 74151 MUX CHIPS E66 AND E59 (INPUT PIN 12)

NOTE: THER MAY BE A PRINT ERROR ON SH7 OF THE M7278. THE
SIGNALS INTO THE MUX CHIPS E66 AND E59 COME FROM THE
M7277 SH5 RATHER FROM M7279 SH3.

2480							
2481							
2482							
2483							
2484							
2485							
2486							
2487							
2488							
2489	006414	010102			MOV	R1,R2	;SET UP REGADR
2490	006416	062702	000016		ADD	#SSR,R2	
2491	006422	012705	027230		MOV	#PATRNB,R5	;SET UP POINTER TO DATA PATTERNS
2492	006426	012567	172552	1\$:	MOV	(R5)+,\$+MP1	;GET THE PATTERNS
2493	006432	012567	172550		MOV	(R5)+,\$+MP2	
2494	006436	001505			BEQ	TST25	;BR IF DONE ALL PATTERNS
2495							
2496	006440	105067	172544	2\$:	CLRB	\$TMP3	;INIT A LINE COUNTER
2497	006444	142711	000017	3\$:	BICB	#17,(R1)	;INIT LINE SELECT BITS IN "SCR"
2498	006450	156711	172534		BISB	\$TMP3,(R1)	;SELECT A LINE IN SCR
2499	006454	142711	000060		BICB	#60,(R1)	;SO WE CLEAR ALL THE MEM EXT BITS
2500	006460	005061	000006		CLR	CAR(R1);CLEAR A	CAR
2501	006464	105267	172520		INCB	\$TMP3	;GENERATE NXT LINE NO.
2502	006470	122767	000020	172512	CMPB	#20,\$TMP3	;CLEARED THE WHOLE THING ?
2503	006476	001362			BNE	3\$	;BR IF NOT
2504							
2505	006500	004767	015776	4\$:	JSR	PC,SELINE	;GO SELECT A LINE NO.
2506	006504	000750			BR	1\$	;BR IF DONE ALL LINES
2507	006506	156711	021500		BISB	LINE,(R1)	;SET UP LINE SELECT BITS
2508	006512	156711	172466		BISB	\$TMP1,(R1)	;SET UP MEM EXT BIT PATTERN
2509	006516	012761	000000	000004	MOV	#0,LPR(R1)	;SCOPE SYNC
2510	006524	012761	000000	000006	MOV	#0,CAR(R1)	;WRITE EXT BITS IN THIS LOCATION
2511							
2512	006532	105067	172454		CLRB	\$TMP4	;INIT A LINE COUNTER
2513	006536	016704	172444	5\$:	MOV	\$TMP2,R4	;SET UP S/B DATA
2514	006542	142711	000017		BICB	#17,(R1)	;INIT SELECT BITS IN "SCR"
2515	006546	156711	172440		BISB	\$TMP4,(R1)	;SET SELECT BITS IN SCR
2516	006552	012761	000000	000014	MOV	#0,BKR(R1)	;SCOPE SYNC
2517	006560	016103	000006		MOV	CAR(R1),R3	;READ THE SELECTED "CAR"
2518	006564	011203			MOV	(R2),R3	;GET THE WAS DATA
2519	006566	042703	177477		BIC	#177477,R3	;CLEAR JUNK BITS
2520	006572	126767	021414	172412	CMPB	LINE,\$TMP4	;LINE UNDER TEST ??
2521	006600	001401			BEQ	6\$	;BR IF YES
2522	006602	005004			CLR	R4	;MAKE S/B DATA = 000000
2523	006604	020304		6\$:	CMP	R3,R4	;WERE MEM EXT BITS CORRECT ?
2524	006606	001412			BEQ	7\$	;BR IF YES
2525							
2526	006610	004767	015534		JSR	PC,SUER2	;GO SET UP ERROR INFO
2527	006614	004567	015754		JSR	R5,SUNJM	;GO SET LINE NO. IN MSG BUFFER
2528	006620	001212			\$TMP4		
2529	006622	031045			EM7+47		
2530	006624	012767	006440	172256	MOV	#2\$,\$LPERR	;SET UP ERROR LOOP RETURN
2531	006632	104007			ERROR	7	;MEM EXT BITS READ INCORRECTLY
2532							
2533	006634	105267	172352	7\$:	INCB	\$TMP4	;GENERATE NXT LINE NO.
2534	006640	122767	000020	172344	CMPB	#20,\$TMP4	;DONE ALL LINES
2535	006646	001674			BEQ	2\$	;BR IF YES

J14

MAINDEC-11-DZDMM-B
DZDHMB.P11 T24

MACY11 27(732) 23-SEP-76 14:33 PAGE 60
TEST THAT "CAR" MEMORY EXT BITS SET/CLR PROPERLY

SEQ 0178

2536 006650 000732

BR

SS

;GO CHECK NEXT LINE

2537
2538
2539
2540 006652 000004
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592

:TEST 25 TEST INTR. ENAB. BITS - INTR. CONDITION DISABLED

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 USED 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
%				
MOV	#3\$, \$LPERR			;SET UP THE ERROR LOOP RETURN
MOV	R1, R2			;MAKE IT REGADR TOO
MOV	DHVCT, R3			;GET FIRST VECTOR ADDRESS
MOV	#4\$, (R3)+			;GO TO 3\$ IF RCVR INTRs
MOVB	DHRLVL, (R3)+			
TSTB	(R3)+			;UPDATE POINTER
MOV	#5\$, (R3)+			;GO TO 5\$ IF XMITTR INTRs

006654 012767 006732 172226
006662 010102
006664 016703 020304
006670 012723 006762
006674 116723 021306
006700 105723
006702 012723 007004

L14

MAINDEC-11-DZDMM-B
DZDMMB.P11 T25MACY11 27(732) 23-SEP-76 14:33 PAGE 62
TEST INTR. ENAB. BITS - INTR. CONDITION DISABLED

SEQ 0180

2593	005706	116713	021275		MOV	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 INTR
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, SUE2A	: 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, SUE2A	: 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+AP	: GO RESTORE TRAP CATCHER IN VECTOR

M14

MAINDEC-11-DZDMM-B
DZDMMB.P11 T26

MACY11 27(732) 23-SEP-76 14:33 PAGE 63
TEST CHAR. AVAIL. I.E. WITH INTR. CONDITION ACTIVE

SEQ 0181

2626
2627
2628
2629 007036 000004
2630
2631
2632
2633

*TEST 26 TEST CHAR. AVAIL. I.E. WITH INTR. CONDITION ACTIVE

TST26: SCOPE
REM %
TEST ABSTRACT:

THIS TEST USES MAIN1. MODE (SCRO9=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. PRINT 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 02

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.

2674
2675
2676
2677
2678
2679
2680 007040 012767 007076 172042
2681 007046 010102

%
MOV #1\$, \$LPERR ;SET UP THE ERROR LOOP RETURN
MOV R1, R2 ;MAKE IT REGADR TOO

N14

MAINDEC-11-DZDMM-B
DZDMMB.P11 T26MACY11 27(732) 23-SEP-76 14:33 PAGE 64
TEST CHAR. AVAIL. I.E. WITH INTR. CONDITION ACTIVE

SEQ 0182

2682	007050	016703	020120	MOV	DHVC, R3	;GET FIRST VECTOR ADDR
2683	007054	012723	007206	MOV	#3\$, (R3)+	;GO TO 3\$ IF RCVR INTR
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 INTR
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 INTR
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

MACY11-11-DZCHM-B
[2D4MB.P11] 727

MACY11 27(732) 23-SEP-76 14:33 PAGE 65
TEST SILO OVFLW. I.E. WITH INTR. CONDITION ACTIVE

SEQ 0193

007216 000004

```

*****
*TEST 27      TEST SILO OVFLW. I.E. WITH INTR. CONDITION ACTIVE
*****
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 INTR.
3. PRIME DH11 TO GENERATE SILO FULL INTR. - ALLOW INTR.
4. REPORT ERROR IF NO RCVR. INTR OCCURS OR A FALSE XMIT INTR. DOES OCCUR
5. AFTER REPORTING ANY ERRORS DETECTED RESET THE SP AND VECTORS THEN GO TO TEST 30

ERRORS:

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

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

%

007223 012767 007253 171662
007226 010102
007233 016703 017740
007234 012723 007366
007240 116723 020742
007244 105723
007246 012723 007342
007252 116713 020731
007256 012711 004000
007262 012706 001100
007266 004767 017620
007272 012711 001000

15:

```

MOV      #15, SLPERR      ;SET UP THE ERROR LOOP RETURN
MOV      R1, R2           ;MAKE IT REGADR TOO
MOV      DHVCT, R3        ;GET FIRST VECTOR ADDR
MOV      #35, (R3)+       ;GO TO 35 IF RCVR INTR
MOV      DHRLVL, (R3) +   ;
TSTB     (R3)+            ;UPDATE POINTER
MOV      #25, (R3)+       ;GO TO 25 ON XMITTR INTR
MOV      DHRLVL, (R3)
MOV      #BIT11, (R1)     ;CLR THE DH11
MOV      #STACK, SP       ;RESET THE SP FOR ERROR LOOPS
JSR      PC, CHPS2        ;GO LOCK OUT INTR
MOV      #BIT09, (R1)     ;SET MAINT MODE BIT

```


MAINDEC-11-DZDMM-B
 DZDMMB.P11 T27

 MACY11 27.732) 23-SEP-76 14:33 PAGE 66
 TEST SILO OVFLW. I.E. WITH INTR. CONDITION ACTIVE

SEQ 0184

007326	052711	040000	BIS	#BIT14,(R1)	:SET SILO OVFLW I.E. BIT
007302	052711	010000	BIS	#BIT12,(R1)	:SET THE SILO FULL BIT TO FORCE INTR
007306	004767	017564	JSR	PC,CHPS1	:GO CLEAR PSW
007312	000240		NOP		:GIVE IT A LITTLE TIME
007314	004767	017606	JSR	PC,SAPS	:SAVE THE ERROR PSW
007320	011103		MOV	(R1),R3	:GET THE WAS DATA
007322	005011		CLR	(R1)	:CLEAR OUT THE SCR
007324	005011		CLR	(R1)	
007326	012704	051000	MOV	#51000,R4	:SET UP S/B DATA
007332	004767	015016	JSR	PC,SUER2A	:GO SET UP ERROR INFO
007336	104043		ERROR	43	:TIMED OUT AWAITING SILO OVFLW INTR
007340	000412		SR	35	:GO EXIT TEST
007342	004767	017560	JSR	PC,SAPS	:SAVE THE ERROR PSW
007346	011103		MOV	(R1),R3	:GET WAS DATA
007350	012704	051000	MOV	#51000,R4	:SET UP S/B DATA
007354	005011		CLR	(R1)	:CLR OUT SCR REG
007356	005011		CLR	(R1)	
007360	004767	014770	JSR	PC,SLER2A	:GO SET UP ERROR INFO
007364	104012		ERROR	12	:UNEXPECTED XMITTR INTR
007366	012706	001100	MOV	#STACK,SP	:RESET THE SP
007372	004767	017320	JSR	PC,RESTAP	:GO RESTORE TRAP CATCHER

25:

35:

J

007376 000004

```

*****
:TEST 30      TEST NON EX MEM I.E. WITH INTR. CONDITION ACTIVE
*****
:ST30: 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
007406 010102
007410 016703 017560
007414 012723 007522
007420 116723 020562
007424 105723
007426 012723 007546
007432 116713 020551
007436 012711 004000
007442 012706 001100
007446 004767 017440
007452 012711 001000

```

15:

```

MOV #15, $LPERR ;SET UP THE ERROR LOOP RETURN
MOV R1, R2 ;MAKE IT REGADR TOO
MOV DHVCT, R3 ;GET FIRST VECTOR ADDR
MOV #25, (R3)+ ;GO TO 25 IF RCVR INTRs
MOVB DHRLVL, (R3)+
TSTB (R3)+ ;UPDATE POINTER
MOV #35, (R3)+ ;GO TO 35 ON XMITTR INTRs
MOVB DHTLVL, (R3)
MOV #BIT11, (R1) ;CLR THE DH11
MOV #STACK, SP ;RESET THE SP FOR ERROR LOOPS
JSR PC, CHPS2 ;GO LOCK OUT INTRs
MOV #BIT09, (R1) ;SET MAINT MODE BIT

```

E15

 MAINDEC-11-DZDMM-5
 DZDMMB.P11 T30

 MACY11 27(732) 23-SEP-76 14:33 PAGE 69
 TEST NON EX MEM I.E. WITH INTR. CONDITION ACTIVE

SEQ 0196

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

F15

MAINDEC-11-DZDMM-B
DZDMMB.P11 T3:MACY11 27(732) 23-SEP-76 14:33 PAGE 69
TEST XMITTR DONE I.E. WITH INTR. CONDITION ACTIVE

SEQ 0197

2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931

007556 000004

```

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

```

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

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

ERRORS:

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

SYNC: M7277 SH3 INIT A H EF2

DEBUG:

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

KEY LOGIC:

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

%

007560 012767 007616 171322
007566 010102
007570 015703 017400
007574 012723 007702
007600 116723 020402
007604 105723
007606 012723 007726
007612 116713 020371
007616 012711 004000
007622 012706 001100
007626 004767 017260
007632 012711 001000
007636 052711 020000
007642 052711 100000

1\$:

```

MOV #1$,SLPERR ;SET UP THE ERROR LOOP RETURN
MOV R1,R2 ;MAKE IT REGADR TOO
MOV DHVCT,R3 ;GET FIRST VECTOR ADDR
MOV #2$, (R3)+ ;GO TO 2$ IF RCVR INTRs
MOVB DHRLVL, (R3)+
TSTB (R3)+ ;UPDATE POINTER
MOV #3$, (R3)+ ;GO TO 3$ ON XMITTR INTRs
MOVB DHTLVL, (R3)
1$: MOV #BIT11, (R1) ;CLR THE DH11
MOV #STACK, SP ;RESET THE SP FOR ERROR LOOPS
JSR PC,CHPS2 ;GO LOCK OUT INTRs
MOV #BIT09, (R1) ;SET MAINT MODE BIT
BIS #BIT13, (R1) ;SET XMIT DONE I.E. BIT
BIS #BIT15, (R1) ;SET THE XMITTR DONE BIT TO FORCE INTR

```

G15

MAINDEC-11-DZDMM-B
DZDMMB.P11 T31MACY11 27(732) 23-SEP-76 14:33 PAGE 70
TEST XMITTR DONE I.E. WITH INTR. CONDITION ACTIVE

SEQ 0188

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	33	;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,RESTAP	;GO RESTORE TRAP CATCHER

H15

MAINDEC-11-DZDMM-B
DZDMMB.P11 T32MACY11 27(732) 23-SEP-76 14:33 PAGE 71
BASIC TRANSMITTER "NPR" LOGIC TEST 1

SEQ 0189

2954
2955
2956
2957 007736 000004
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009

```

*****
:TEST 32      BASIC TRANSMITTER "NPR" LOGIC TEST 1
*****
:ST32:  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 GOT 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

MAINDEC-11-DZDMM-8
DZDMMB.P11 T32

MACY11 27(732) 23-SEP-76 14:33 PAGE 72
BASIC TRANSMITTER "NPR" LOGIC TEST 1

SEQ 0190

3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065

2. CLOCK SIGNAL OR BYTE COUNT DECODER ON M7278 SH3 (XMIT FINISHED PULSE L)
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

%

```

MOV #2$, $LPERR ;SET UP ERROR LOOP RETURN
MOV DHVCT, R3 ;GET THE FIRST VECTOR ADDRESS
ADD #4, R3 ;POINT TO XMITTR ENTRY
MOV #4$, (R3)+ ;GO TO 4$ ON XMITTR INTR
MOV DHTLVL, (R3)
MOVB PC, SELINE ;GO SELECT A LINE NO. TO TEST
BR B$ ;BR IF TESTED ALL SELECTED LINES

2$: MOV #STACK, SP ;RESET SP FOR ERROR LOOPS
MOV R1, R2 ;SET UP REGADR
MOV #120000, R4 ;SET UP S/B DATA
BISB LINE, R4
JSR PC, CLCABC ;GO CLEAR CAR AND BCR MEMORIES
JSR PC, CHPS2 ;GO LOCK OUT INTRs
MOV #BIT11, (R1) ;CLEAR THE DH11 INTERFACE
BISB LINE, (R1) ;SELECT A LINE NO.
MOV #-1, BCR(R1) ;SET BYTE COUNT TO -1
MOV #33500, LPR(R1) ;SET UP LINE PARAMETERS
BIS LINMSK, BAR(R1) ;ACTIVATE SELECTED LINE

```

007740	012767	007774	171142
007746	016703	017222	
007752	062703	000004	
007756	012723	010140	
007762	116713	020221	
007766	004767	014510	
007772	000552		
007774	012706	001100	
010000	010102		
010002	012704	120000	
010006	156704	020200	
010012	004767	014636	
010016	004767	017070	
010022	012711	004000	
010026	156711	020160	
010032	012761	177777	000010
010040	012761	033500	000004
010046	056761	017132	000012

MAINDEC-11-DZDMM-B
DZDMMB.P11 T32MACY11 27(732) 23-SEP-76 14:33 PAGE 73
BASIC TRANSMITTER "NPR" LOGIC TEST 1

SEQ 0191

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 IIMER 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 ;THIS BRANCH IF TIMEOUT OCCURS
3074							
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	RS,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							

K15

MAINDEC-11-DZDHM-B
DZDHMB.P11 T32MACY11 27(732) 23-SEP-76 14:33 PAGE 74
BASIC TRANSMITTER "NPR" LOGIC TEST 1

SEQ 0192

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	014344		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

3144
3145
3146
3147 010370 000004
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199

```

*****
*TEST 33      TRANSMITTR NPR LOGIC TEST 2
*****
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)

3194 010372 012767 010406 170510
3195 010400 004767 014076
3196 010404 000544
3197 010406 052711 004000
3198 010412 004767 014236
3199 010416 004767 014274

```

%      MOV      #25,$LPERR      ;SET UP ERROR LOOP RETURN
1$:    JSR      PC,SELINE      ;GO SELECT A LINE TO TEST
      BR       TST34          ;BR IF DONE ALL SELECTED LINES
2$:    BIS      #BIT11,(R1)     ;CLEAR THE DH11
      JSR      PC,CLCABC      ;GO CLEAR "CAR" AND "BCR" MEMORIES
      JSR      PC,LDBCR       ;GO LOAD "BCR" MEMORY WITH ALL ONES

```

M15

MAINDEC-11-DZDMM-B
DZDHMB.P11 T33MACY11 27(732) 23-SEP-76 14:33 PAGE 76
TRANSMITTER NPR LOGIC TEST 2

SEQ 0194

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			TST	(R1)	; XMITR 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			EM50+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\$: MOV	\$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	MOV	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	MOV	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

N15

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 77
DZDMMB.P11 T33 TRANSMITTR NPR LOGIC TEST 2

SEQ 0195

3256	010676	104051		ERROR	51	:CAR REG INCORRECT
3257						
3258	010700	005267	170314	9\$:	INC	\$TMP7
3259	010704	022767	000020	170306	CMP	#20,\$TMP7
3260	010712	001307			BNE	5\$
3261	010714	000631			BR	1\$

:GENERATE NEW LINE NO.
:TESTED ALL LINES
:BR IF NOT
:GO SELECT NEXT ACTIVE LINE

M7279	SH1	SILO DATA MU'Y'S (74157'S) SSR15 H	CR1
	SH2	DATA READY L NFC 15 H SILO MEMORY LOAD SILO L 5.068 MHZ (CLOCK) SSR (13:00) H	DV1 DL1 (E13,E17,E8,E3) (3341'S) DJ2 DN1 (E20,E24)
M7289	SH6	RCV INT REQ H	DP1
M7278	SH8	SSR (03:00) H	

C16

 MAINDEC-11-DZHM-8
 DZHM8.P11 134

 MACY11 27(732) 23-SEP-76 14:33 PAGE 79
 TEST THAT CHARACTER AVAILABLE CAN CAUSE RCVR INTERRUPT

SEQ 0197

SH7 SSR (07:00) 4

3318									
3319									
3320									
3321									
3322	010720	012767	010742	170162	MOV	#15,SLPERR		:SET UP THE ERROR LOOP RETURN	
3323	010726	015703	015242		MOV	DHVC7,R3		:GET FIRST VECTOR ADDR	
3324	010732	012723	011026		MOV	#35,(R3)+		:GO TO 35 ON RCVR INTERRUPT	
3325	010736	116713	017244		MOVB	CHRLVL,(R3)			
3326	010742	004767	016144	15:	JSR	PC,CHPS2		:GO LOCK OUT INTR	
3327	010746	012706	001100		MOV	#STACK,SP		:RESET SP FOR ERROR LOOPS	
3328	010752	012711	004000		MOV	#BIT11,(R1)		:CLEAR THE DH11	
3329	010756	052761	100000	000016	BIS	#BIT15,SSR(R1)		:SET SILO MAINT. BIT TO LOAD SILO	
3330	010764	012711	000100		MOV	#BIT06,(R1)		:ENABLE CHAR. AVAIL INTERRUPT	
3331	010770	004767	015102		JSR	PC,CHPS1		:GO CLEAR PSW	
3332	010774	012703	001000		MOV	#1000,R3		:INIT TIMER	
3333	011000	005303		25:	DEC	R3		:DEC TIMER	
3334	011002	001376			BNE	25		:BR IF NO TIMEOUT	
3335									
3336	011004	004767	016116		JSR	PC,SAPS		:SAVE THE ERROR PSW	
3337	011010	011103			MOV	(R1),R3		:GET THE WAS DATA	
3338	011012	012704	000300		MOV	#300,R4		:SET UP S/B DATA	
3339	011016	004767	013332		JSR	PC,SUER2A		:GO SET UP ERROR INFO	
3340	011022	104013			ERROR	13		:CHAR AVAIL FAILED TO SET ON TIME	
3341	011024	000436			BR	55		:ESCAPE FROM THIS TEST - CATASTROPHIC ERROR	
3342									
3343	011026	016105	000016	35:	MOV	SSR(R1),R5		:SAVE THE SILO STATUS REG.	
3344	011032	016103	000002		MOV	NRC(R1),R3		:GET THE WAS DATA	
3345	011036	012704	125252		MOV	#125252,R4		:SET UP S/B DATA	
3346	011042	020334			CMP	R3,R4		:WAS = S/B = 125252 ??	
3347	011044	001410			BEG	45		:BR IF IT IS	
3348									
3349	011046	004767	016054		JSR	PC,SAPS		:SAVE THE ERROR PSW	
3350	011052	062702	000002		ADD	#NRC,R2		:SET UP REGADR	
3351	011056	004767	013272		JSR	PC,SUER2A		:GO SET UP ERROR INFO	
3352	011062	104052			ERROR	52		:DATA COMPARE ERROR	
3353	011064	000416			BR	55		:GET CLT	
3354									
3355	011066	010503		45:	MOV	R5,R3		:NOW GET THW SILO STATUS REG AGAIN	
3356	011070	042703	140377		BIC	#140377,R5		:CLR OUT JUNK	
3357	011074	012704	000400		MOV	#400,R4		:SET UP S/B DATA	
3358	011100	020304			CMP	R3,R4		:SSR CHAR COUNT = 1 ??	
3359	011102	001407			BEG	55		:BR IF IT IS	
3360									
3361	011104	004767	016016		JSR	PC,SAPS		:SAVE THE ERROR PSW	
3362	011110	062702	000016		ADD	#SSR,R2		:SET UP REGADR	
3363	011114	004767	013234		JSR	PC,SUER2A		:SET UP ERROR INFO	
3364	011120	104006			ERROR	6		:SSR COUNT NOT CORRECT	
3365									
3366	011122	012706	001100	55:	MOV	#STACK,SP		:RESET THE STACK POINTER	
3367	011126	004767	015744		JSR	PC,CHPS1		:GO CLEAR PSW	
3368	011132	005011			CLR	(R1)		:RESET I.E. BIT	
3369	011134	015703	016034		MOV	DHVC7,R3		:GET FIRST VECTOR ADDR	
3370	011140	010313			MOV	R3,(R3)		:RESTORE TRAP CATCHER	
3371	011142	062723	000002		ADD	#2,(R3)+			
3372	011146	005013			CLR	(R3)			

INDEX-11-DZDM-B
DZDM-B.P11 735

MACY11 27(732) 23-SEP-76 14:33 PAGE 80
TEST THAT THE SILO STATUS REG COUNTS UP CORRECTLY

SE0 0198

01:152 000034

```

*****
:TEST 35      TEST THAT THE SILO STATUS REG COUNTS UP CORRECTLY
*****
:ST35: 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 "STMP7" 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 "STMP7" AND REPEAT 1 THRU 4
UNTIL ALL COUNTS (01-77) HAVE BEEN TESTED

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 BAC M7279 MODULE

KEY LOGIC:

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

011152	010102	
011154	062702	000016
011160	012767	000001 170032
011166	012711	004000
011172	016705	170022
011176	005004	
011200	052712	100000
011204	012703	001000
011210	005303	
011212	001376	
011214	042712	100000
011220	005204	
011222	005305	

```

%      MOV      R1,R2      :MAKE REGADR = SSR
      ADD      #SSR,R2
      MOV      #1,STMP7    :START WITH COUNT OF 1
15:    MOV      #BIT11,(R1) :CLEAR THE DH11
      MOV      STMP7,R5    :SAVE CHARACTER COUNT BEING TESTED
      CLR      R4         :INIT A CHAR COUNTER
25:    BIS      #BIT15,(R2) :SET THE SILO MAINT BIT
      MOV      #1000,R3    :INIT A TIMER
35:    DEC      R3         :STALL TO ALLOW TIME TO LOAD SILO
      BNE      35
      BIC      #BIT15,(R2) :CLEAR SILO MAINT. BIT
      INC      R4         :COUNT A CHAR LOADED
      DEC      R5         :DECREMENT TEST COUNT
    
```

E16

MAINDEC-11-DZCHM-B
DZCHMB.P11 T35

MACY11 27(732) 23-SEP-76 14:33 PAGE 81
TEST THAT THE SILO STATUS REG COUNTS UP CORRECTLY

SEQ 0199

3430	011224	001365			BNE	25		: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			SEQ	45		:BR IF YES
3437								
3438	011242	004767	013102		FC,SUER2			:GO SET UP ERROR INFO
3439	011246	012767	011166	167634	MOV	#15,SLPERR		:SET UP ERROR LOOP RETURN
3440	011254	104006			ERROR	6		:SSR FAILED TO UP-COUNT CORRECTLY
3441								
3442	011256	005267	167736		INC	\$TMP7		:INCREMENT TO NEXT COUNT TO TEST
3443	011260	022767	000100	167730	CMP	#100,\$TMP7		:MAXIMUM COUNT ??
3444	011270	001336			BNE	15		:BR IF NOT


```

*****
:TEST 36      TEST THAT SILO STATUS REGISTER DOWN COUNTS CORRECTLY
*****
:ST36: 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)

%

```

MOV R1,R2      :SET UP REGADR
ADD #SSR,R2
MOV #1,$TMP7    :START WITH COUNT = 1
:1$: MOV #BIT11,(R1) :CLR THE DH11
MOV #100,R5     :TEST COUNT SHOULD BE 64(10) MINUS
SUB $TMP7,R5    :THE NO. OF CHARS READ
MOV #100,R3     :COUNTER USED TO FILL SILO
:2$: MOV #1000,R4  :INIT TIMER
BIS #BIT15,(R2) :SET SILO MAINT. BIT
:3$: DEC R4       :STALL TO ALLOW SILO TO LOAD
BNE 3$
BIC #BIT15,(R2) :CLEAR THE SILO MAINT BIT
DEC R3         :COUNT ONE CHAR LOADED
BNE 2$        :BR UNTIL ALL LOADED

MOV $TMP7,R3   :INIT COUNTER FOR READING SILO
:4$: MOV #1000,R4 :INIT TIMER
TST NRC(R1)    :READ THE SILO
:5$: DEC R4      :GIVE IT TIME TO SETTLE
BNE 5$
DEC R3         :COUNT ONE READ
BNE 4$        :BR UNTIL WE'VE READ TEST COUNT

```

```

3446
3447
3448
3449 011272 000004
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480 011274 010102
3481 011276 062702 000016
3482 011302 012767 000001 167710
3483 011310 012711 004000
3484 011314 012705 000100
3485 011320 166705 167674
3486 011324 012703 000100
3487 011330 012704 001000
3488 011334 052712 100000
3489 011340 005304
3490 011342 001376
3491 011344 042712 100000
3492 011350 005303
3493 011352 001366
3494
3495 011354 016703 167640
3496 011360 012704 001000
3497 011364 005761 000002
3498 011370 005304
3499 011372 001376
3500 011374 005303
3501 011376 001370

```

MAINDEC-11-DZDHM-B
DZDHMB.P11 T36

MACY11 27(732) 23-SEP-76 14:33 PAGE 83
TEST THAT SILO STATUS REGISTER DOWN COUNTS CORRECTLY

SEQ 0201

Address	Op Code	Op	Op 2	Op 3	Op 4	Op 5	Op 6	Op 7	Op 8	Op 9	Op 10	Op 11	Op 12	Op 13	Op 14	Op 15	Op 16	Op 17	Op 18	Op 19	Op 20	Op 21	Op 22	Op 23	Op 24	Op 25	Op 26	Op 27	Op 28	Op 29	Op 30	Op 31	Op 32	Op 33	Op 34	Op 35	Op 36	Op 37	Op 38	Op 39	Op 40	Op 41	Op 42	Op 43	Op 44	Op 45	Op 46	Op 47	Op 48	Op 49	Op 50	Op 51	Op 52	Op 53	Op 54	Op 55	Op 56	Op 57	Op 58	Op 59	Op 60	Op 61	Op 62	Op 63	Op 64	Op 65	Op 66	Op 67	Op 68	Op 69	Op 70	Op 71	Op 72	Op 73	Op 74	Op 75	Op 76	Op 77	Op 78	Op 79	Op 80	Op 81	Op 82	Op 83	Op 84	Op 85	Op 86	Op 87	Op 88	Op 89	Op 90	Op 91	Op 92	Op 93	Op 94	Op 95	Op 96	Op 97	Op 98	Op 99	Op 100	Op 101	Op 102	Op 103	Op 104	Op 105	Op 106	Op 107	Op 108	Op 109	Op 110	Op 111	Op 112	Op 113	Op 114	Op 115	Op 116	Op 117	Op 118	Op 119	Op 120	Op 121	Op 122	Op 123	Op 124	Op 125	Op 126	Op 127	Op 128	Op 129	Op 130	Op 131	Op 132	Op 133	Op 134	Op 135	Op 136	Op 137	Op 138	Op 139	Op 140	Op 141	Op 142	Op 143	Op 144	Op 145	Op 146	Op 147	Op 148	Op 149	Op 150	Op 151	Op 152	Op 153	Op 154	Op 155	Op 156	Op 157	Op 158	Op 159	Op 160	Op 161	Op 162	Op 163	Op 164	Op 165	Op 166	Op 167	Op 168	Op 169	Op 170	Op 171	Op 172	Op 173	Op 174	Op 175	Op 176	Op 177	Op 178	Op 179	Op 180	Op 181	Op 182	Op 183	Op 184	Op 185	Op 186	Op 187	Op 188	Op 189	Op 190	Op 191	Op 192	Op 193	Op 194	Op 195	Op 196	Op 197	Op 198	Op 199	Op 200	Op 201	Op 202	Op 203	Op 204	Op 205	Op 206	Op 207	Op 208	Op 209	Op 210	Op 211	Op 212	Op 213	Op 214	Op 215	Op 216	Op 217	Op 218	Op 219	Op 220	Op 221	Op 222	Op 223	Op 224	Op 225	Op 226	Op 227	Op 228	Op 229	Op 230	Op 231	Op 232	Op 233	Op 234	Op 235	Op 236	Op 237	Op 238	Op 239	Op 240	Op 241	Op 242	Op 243	Op 244	Op 245	Op 246	Op 247	Op 248	Op 249	Op 250	Op 251	Op 252	Op 253	Op 254	Op 255	Op 256	Op 257	Op 258	Op 259	Op 260	Op 261	Op 262	Op 263	Op 264	Op 265	Op 266	Op 267	Op 268	Op 269	Op 270	Op 271	Op 272	Op 273	Op 274	Op 275	Op 276	Op 277	Op 278	Op 279	Op 280	Op 281	Op 282	Op 283	Op 284	Op 285	Op 286	Op 287	Op 288	Op 289	Op 290	Op 291	Op 292	Op 293	Op 294	Op 295	Op 296	Op 297	Op 298	Op 299	Op 300	Op 301	Op 302	Op 303	Op 304	Op 305	Op 306	Op 307	Op 308	Op 309	Op 310	Op 311	Op 312	Op 313	Op 314	Op 315	Op 316	Op 317	Op 318	Op 319	Op 320	Op 321	Op 322	Op 323	Op 324	Op 325	Op 326	Op 327	Op 328	Op 329	Op 330	Op 331	Op 332	Op 333	Op 334	Op 335	Op 336	Op 337	Op 338	Op 339	Op 340	Op 341	Op 342	Op 343	Op 344	Op 345	Op 346	Op 347	Op 348	Op 349	Op 350	Op 351	Op 352	Op 353	Op 354	Op 355	Op 356	Op 357	Op 358	Op 359	Op 360	Op 361	Op 362	Op 363	Op 364	Op 365	Op 366	Op 367	Op 368	Op 369	Op 370	Op 371	Op 372	Op 373	Op 374	Op 375	Op 376	Op 377	Op 378	Op 379	Op 380

H16

MAINDEC-11-DZDMM-B
DZDMMB.F11 T37

MACY11 27(732) 23-SEP-76 14:33 PAGE 84
TEST SILO ALARM LEVEL FOR COUNTS 0,1,2,4,8,16, AND 32

SEQ 0202

3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574

011446 000004

: TEST 37 TEST SILO ALARM LEVEL FOR COUNTS 0,1,2,4,8,16, AND 32

TEST37: 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
011456 010102
011460 062702 000016
011464 005067 167530
011470 012711 004000
011474 016705 167520
011500 010512

```

MOV #15,$LPERR ;SET UP THE ERROR LOOP RETURN
MOV R1,R2 ;SET UP REGADR
ADD #SSR,R2
CLR $TMP7 ;START WITH LEVEL 00
1$: MOV #BIT11,(R1) ;CLEAR THE DH11
MOV $TMP7,R5 ;SAVE IT IN R5
MOV R5,(R2) ;SET ALARM LEVEL IN SSR

```

MAINDEC-11-DZDMM-8
 DZDMMB.P11 T37

 MACY11 27(732) 23-SEP-76 14:33 PAGE 85
 TEST SILO ALARM LEVEL FOR COUNTS 0,1,2,4,8,16, AND 32

SEQ 0203

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	:RETRN 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	011650	001703			BEQ	1\$	:BR IF NOT

J16

MAINDEC-11-DZDMM-B
DZDMMB.P11 T40

MACY11 27(732) 23-SEP-76 14:33 PAGE 86
TRANSMITTER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS

SEQ 0204

3620
3621
3622
3623 011662 000004
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675

::*****
:*TEST 40 TRANSMITTER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS
:*****

TST40: SCOPE
.REM %
TEST ABSTRACT:

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

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

ERRORS:

1. [ERROR 53] IS CALLED TO REPORT XMIT TIMEOUT ERRORS
2. [ERROR 17] IS CALLED TO REPORT TIMING ERRORS

SYNC: M7277 SH3 INIT A H EF2

DEBUG:

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

KEY LOGIC:

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

K16

MAINDEC-11-DZDMM-B
DZDMMB.P11 T40MACY11 27(732) 23-SEP-76 14:33 PAGE 87
TRANSMITTER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS

SEQ 0205

3676	011664	012767	011714	167216	MOV	#2\$,SLPERR	:SET UP ERROR LOOP RETURN
3677	011672	004767	012604		JSR	PC,SELINE	: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		MOV	#BIT11,(R1)	:CLEAR THE DH11
3682	011720	156711	016266		SISB	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			TST	(R1)	:XMITR 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		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		JSR	PC,SAPS	:SAVE THE ERROR PSW
3720	012106	016702	167106		MOV	\$TMP7,R2	:GET SPEED CODE AND RIGHT JUSTIFY
3721	012112	000302			SWAB	F.2	
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	MOV	TIMEB,TIMEC	:SET UP NEW CHECK TIMER COUNT

L16

MAINDEC-11-DZDHM-B
DZDHMB.P11 T40

MACY11 27(732) 23-SEP-76 14:33 PAGE 89
TRANSMITTER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS

SEQ 0206

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							

M16

MAINDEC-11-DZDMM-B
DZDMMB.P11 T41

MACY11 27(732) 23-SEP-76 14:33 PAGE 89
RECEIVER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS

SEQ 0207

3737
3738
3739
3740 012170 000004
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792

```

*****
*TEST 41 RECEIVER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS
*****
TST41: SCOPE
      REM %
TEST ABSTRACT:
*****

```

THIS TEST IS IDENTICAL TO TEST 42 EXCEPT IT WAITS FOR "DATA READ"
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

```

%
1$: MOV #25,$LPERR ;SET UP ERROR LOOP RETURN
    JSR PC,SELINE ;GO SELECT A LINE TO TEST
    BR TST42 ;BR IF TESTED ALL SELECTED LINES
    MOV #2100,$TMP7 ;INIT TO START WITH LOWEST SPEED
    MOV #-1,$TIMEC ;INIT RELATIVE TIME CHECKER
2$: MOV #BIT11,(R1) ;CLEAR THE DH11
3$: ETSB LINE,(R1) ;SELECT IT IN THE SCR
    MOV #-1,$BCR(R1) ;SET BYTE COUNT TO XFER 1 CHAR
    CLR CAR(R1) ;GET TEST DATA STARTING AT LOC. 0
    MOV $TMP7,$LPR(R1) ;SELECT A XMIT SPEED
    MOV LINMSK,$BAR(R1) ;ACTIVATE THE TRANSMITTER

4$: MOV #1,$TIMEA ;INIT TIMER A
    CLR $TIMEB ;INIT TIMER B
    TSTB (R1) ;RCVR DONE YET ??
    BMI $5 ;BR IF YES
    JSR PC,$TIMEIT ;CALL THE TIMER
    BR $4 ;TIMER ROUTINE WILL MOVE RETURN PC
    ; AROUND THIS BRANCH IF TIME OUT OCCURS

MOV $TMP7,$TMP0 ;SAVE AND SET UP THE SPEED CODE
ASL $TMP0
ASL $TMP0
SWAB $TMP0

```


SECRET

520 0208

Address	Hex	Assembly	Comment
000000	000000	BIC	#177760, STMP0
000001	000000	MOV	(R1), R3
000002	000000	R1, R2	:GET THE WAS DATA
000003	000000	MOV	#BIT15+BIT07, R4
000004	000000	LINE, R4	:MAKE REGADR = DEVADR
000005	000000	PC, SUER2A	:SET UP S/B DATA
000006	000000	R5, SUNUM	:GO SET UP ERROR INFO
000007	000000		:GO SET LINE NO. IN MSG
000008	000000		
000009	000000		
000010	000000		
000011	000000		
000012	000000		
000013	000000		
000014	000000		
000015	000000		
000016	000000		
000017	000000		
000018	000000		
000019	000000		
000020	000000		
000021	000000		
000022	000000		
000023	000000		
000024	000000		
000025	000000		
000026	000000		
000027	000000		
000028	000000		
000029	000000		
000030	000000		
000031	000000		
000032	000000		
000033	000000		
000034	000000		
000035	000000		
000036	000000		
000037	000000		
000038	000000		
000039	000000		
000040	000000		
000041	000000		
000042	000000		
000043	000000		
000044	000000		
000045	000000		
000046	000000		
000047	000000		
000048	000000		
000049	000000		
000050	000000		
000051	000000		
000052	000000		
000053	000000		
000054	000000		
000055	000000		
000056	000000		
000057	000000		
000058	000000		
000059	000000		
000060	000000		
000061	000000		
000062	000000		
000063	000000		
000064	000000		
000065	000000		
000066	000000		
000067	000000		
000068	000000		
000069	000000		
000070	000000		
000071	000000		
000072	000000		
000073	000000		
000074	000000		
000075	000000		
000076	000000		
000077	000000		
000078	000000		
000079	000000		
000080	000000		
000081	000000		
000082	000000		
000083	000000		
000084	000000		
000085	000000		
000086	000000		
000087	000000		
000088	000000		
000089	000000		
000090	000000		
000091	000000		
000092	000000		
000093	000000		
000094	000000		
000095	000000		
000096	0000		

C01

MAINCES-11-DZDM-B
DZDMB.F11 T41

MACY11 27(732) 23-SEP-76 14:33 PAGE 91
RECEIVER TIMING TEST - ALL SELECTED LINES - ALL SPEEDS

SEP 2209

3829

001

MAINTFC-11-02DMM-B
2274 B.P.11 T42

MACY11 27(732) 23-SEP-76 14:33 PAGE 92
VERIFY STORAGE OVERFLOW - NON MAINT. MODE - ALL SELECTED LINES

SEP 0210

:TEST 42 VERIFY STORAGE OVERFLOW - NON MAINT. MODE - ALL SELECTED LINES

012472 000004

TST42: SCOPE
REM %
TEST ABSTRACT:

THIS TEST VERIFIES THAT THE STORAGE OVERFLOW BIT (SCR14) SETS
AND CLEARS PROPERLY FOR ALL SELECTED LINES. THE TEST SEQUENCE IS AS
FOLLOWS:

1. SET UP THE ERROR RETURN
2. SELECT A LINE NUMBER TO TEST - GO TO NEXT TEST IF ALL
SELECTED LINES HAVE BEEN TESTED.
3. PRIME THE SELECTED LINE TO XMIT 65(10) CHARS.
4. ACTIVATE THE SELECTED LINE AND WAIT FOR STORAGE OVERFLOW
TO SET (SCR14=1)
5. IF SCR14 FAILS TO SET ON TIME - REPORT ERROR AND THEN
CONTINUE WITH THE NEXT LINE (STEP 2).
6. IF IT SETS OK - READ THE "NRC" REG TWICE TO EMPTY TWO
WORDS FROM THE SILO.
7. AFTER A BRIEF STALL, VERIFY THAT SCR14 HAS CLEARED -
IF NOT REPORT ERROR AND CONTINUE WITH NEXT LINE (STEP2)
8. IF IT CLEARS OK, VERIFY THAT THE FILL COUNT (SSR:15:08)
CONTAINS A 77(8) - IF NOT REPORT ERROR AND CONTINUE
WITH NEXT LINE (STEP 2).
9. REPEAT STEPS 2 THRU 8 UNTIL ALL SELECTED LINES HAVE BEEN
TESTED.

ERRORS:

1. IERROR 571 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
012502 004767 011774

%
18: MOV #23,SLPERR :SET UP ERROR RETURN
JSR PC,SELIN :GO SELECT A LINE NO. TO TEST

E01

23-SEP-76 14:33 PAGE 93
 23-SEP-76 14:33 PAGE 93
 23-SEP-76 14:33 PAGE 93

MACY:1 27(732) 23-SEP-76 14:33 PAGE 93
 VERIFY STORAGE OVERFLOW - NON MAINT. MODE - ALL SELECTED LINES

SEQ 0211

000000	012536	000535	004000	25:	BR	TST43	::BR IF DONE ALL SELECTED LINES
000001	012537	012537			MOV	#BIT11,(R1)	:CLEAR THE DH11
000002	012538	010102			MOV	R1,R2	:MAKE REGADR = DEVAOR
000003	012539	016711	015470		MOVB	LINE,(R1)	:SELECT A LINE
000004	012540	005261	000006		CLR	CAR(R1)	:SET UP CURRENT ADDRESS
000005	012541	012761	177677	000010	MOV	#-65,BCR(R1)	:SET UP BYTE COUNT
000006	012542	012761	033500	000004	MOV	#33500,LPR(R1)	:SET UP LPR: 9600 BAUD.5 BIT CHARS
000007	012543	016761	014436	000012	MOV	LINMSK,BAR(R1)	:ACTIVATE THE SELECTED LINE
000008	012550	012767	000001	015462	MOV	#1,TIMEA	:INIT TIMERS
000009	012551	005067	015460		CLR	TIMEB	
000010	012552	032711	040000	35:	BIT	#BIT14,(R1)	:STOREGE OVERFLOW YET ??
000011	012553	001024			BNE	45	:BR IF YES YOU SHOULD GET IT
000012	012554	004767	014150		JSR	PC,TIMEIT	:CALL TIMER
000013	012574	000772			BR	35	:BR IF NO TIME OUT
000014	012576	004767	014324		JSR	PC,SAPS	:GO SAVE PSW
000015	012602	012704	040000		MOV	#BIT14,R4	:SET UP S/B DATA
000016	012606	156704	015400		BISB	LINE,R4	
000017	012612	011103			MOV	(R1),R3	:SET UP WAS DATA
000018	012614	042703	137760		BIC	#137760,R3	:CLEAR UNINTERESTING BITS
000019	012620	004767	011530		JSR	PC,SUER2A	:GO SET UP ERROR INFO
000020	012624	004567	011744		JSR	R5,SUNUM	:PUT LINE NO. IN MESSAGE
000021	012630	030212			LINE		
000022	012632	034631			EM57+44		
000023	012634	104057			ERROR	57	:STORAGE OVERFLOW FAILED TO SET
000024	012636	000721			BR	15	:GO TRY NEXT LINE
000025	012640	016167	000002	166352	45:	MOV	NRC(R1),STMP7
000026	012646	016167	000002	166344		MOV	NRC(R1),STMP7
000027	012654	012705	001000		MOV	#1000,R5	:INIT STALL COUNTER
000028	012660	005305			DEC	R5	:COUNT TIMER
000029	012662	001376			BNE	415	:BR IF NO TIMEOUT
000030	012664	032711	040000		BIT	#BIT14,(R1)	:DID OVERFLOW GO AW
000031	012670	001420			BEO	55	:BR IF YES
000032	012672	004767	014230		JSR	PC,SAPS	:GO SAVE THE PSW
000033	012676	005004			CLR	R4	:SET UP S/B DATA
000034	012700	156704	015306		BISB	LINE,R4	
000035	012704	011103			MOV	(R1),R3	:SET UP WAS DATA
000036	012706	042703	137760		BIC	#137760,R3	:CLEAR UNINTERESTING BITS
000037	012712	004767	011436		JSR	PC,SUER2A	:GO SET UP ERROR INFO
000038	012716	004567	011652		JSR	R5,SUNUM	:PUT LINE NO. IN MSG
000039	012722	030212			LINE		
000040	012724	034631			EM57+44		
000041	012726	104057			ERROR	57	:STORAGE BIT FAILED TO CLEAR
000042	012730	000664			BR	15	:GO TRY NEXT LINE
000043	012732	122761	000077	000017	55:	CMPB	#77,SSR+1(R1)
000044	012740	001660			BEO	15	:WAS IT REALLY 65. ??
000045	012742	004767	014160		JSR	PC,SAPS	:GO SAVE PSW
000046	012746	012704	037400		MOV	#37400,R4	:SET UP S B DATA
000047	012752	062702	000016		ADD	#SSR,R2	:SET UP REGADR
000048	012756	016103	000016		MOV	SSR(R1),R3	:SAVE WAS DATA
000049	012762	004767	011366		JSR	PC,SUER2A	:GO SET UP ERROR INFO

F01

MAINDEC-11-DZDMM-B
DZDMMB.P11 T42

MACY11 271732) 23-SEP-76 14:33 PAGE 94
VERIFY STORAGE OVERFLOW - NON MAINT. MODE - ALL SELECTED LINES

SEQ C212

3942	012766	004567	211602	JSR	RS,SUNUM	;PUT LINE NO. IN MSG
3943	012772	030212		LINE		
3944	012774	034631		EMS7+44		
3945	012776	104057		ERROR	57	:READING SILO F *LED TO DEC SSP OR
3946						:STORAGE OVFL SE *AT WRONG COUNT
3947	013000	000640		BR	18	:GO TRY NEXT LINE

GO1

MAINDEC-11-DZDMM-B
DZDMMB.P11 T43

MACY11 27(732) 23-SEP-76 14:33 PAGE 95
BASIC DATA TEST - ALL SELECTED LINES ALL CHAR LENGTHS

SEC 0213

3948
3949
3950
3951 013002 000004
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003

:TEST 43 BASIC DATA TEST - ALL SELECTED LINES ALL CHAR LENGTHS

TEST43: 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 9.
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 551 IS CALLED TO REPORT RCVR TIMEOUT.
2. ERROR 231 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

MOV #35,\$LPERR ;SET UP ERROR LOOP RETURN

013004 012767 013032 166076

H01

MAINDEC-11-DDDM-B
DDDMB.P11 T43MACY11 27(732) 23-SEP-76 14:33 PAGE 96
BASIC DATA TEST - ALL SELECTED LINES/ALL CHAR LENGTHS

SEQ 0214

4004	013012	004767	011464	15:	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	25:	MOV	(R5)+, \$TMP7	:PUT TEST CHAR IN XMIT BUFFER
4009	013032	012711	004000	35:	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		45:	TSTB	(R1)	:RCVR DONE YET ??
4020	013112	100424			BMI	55	:BR IF YES
4021	013114	004767	013624		JSR	PC, TIMEIT	:CALL THE TIMER
4022	013120	000773			BR	45	: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	65	:GO TEST NEXT CHAR LENGTH
4036							
4037	013164	016103	000002	55:	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	65	: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		65:	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	25	:BR IF NOT
4054	013240	000664			BR	15	:GO DO NEXT LINE

4055
4056
4057
4058 013242 000004
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110

```

*****
:TEST 44      SINGLE LINE DATA TEST - ALL SELECTED LINES
*****
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 (SCRD9=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.

4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166

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)

```
%
1$:  MOV    #2$,SLPERR      ;SET UP ERROR LOOP RETURN
      JSR    PC,SELINE     ;GO SELECT A LINE TO TEST
      BR     11$           ;BR IF ALL SELECTED LINES DONE
      BR     2$            ;GO TEST IT
11$:  JMP     2$            ;EXIT TEST
2$:   MOV    DHADR,R1       ;RESET DEVADR
      MOV    #BIT11,(R1)    ;CLEAR THE DH11
      BISB   LINE,(R1)      ;SET SELECT BITS IN SCR
      MOV    #TBUF,CAR(R1)  ;SET UP BUS ADDRESS REG
      MOV    #-400,BCR(R1)  ;SET UP BYTE COUNT
      MOV    #33503,LPR(R1) ;SET LINE PARAMETERS
      MOV    #RBUF,R5       ;SET UP POINTER TO INPUT DATA BUFFER
      BIS    #BIT09,(R1)    ;SET MAINT MODE BIT
      MOV    LINMSK,BAR(R1) ;ACTIVATE THE SELECTED LINE

      MOV    #2,TIMEA       ;INIT TIMER A
      CLR    TIMES         ;INIT TIMER B
3$:   TSTB   (R1)           ;RCVR DONE YET ??
      BMI    4$            ;BR IF YES
      JSR    PC,TIMEIT      ;CALL THE TIMER
      BR     3$            ;TIMER ROUTINE WILL MOVE RETURN PC
                               ;AROUND THIS BRANCH IF TIME OUT OCCURS

      JSR    PC,SAPS        ;SAVE THE ERROR PSW
      MOV    R1,R2         ;SET UP REGADR
      MOV    (R2),R3        ;GET THE WAS DATA
      BIC    #176400,R3     ;CLEAR JUNK BITS
      MOV    #1200,R4       ;SET UP S/B DATA
      BISB   LINE,R4
      JSR    PC,SUER2A      ;GO SET UP ERROR INFO
      JSR    R5,SUNUM       ;PUT LINE NO. IN MESSAGE

      EM22+51              ;CHAR AVAIL TIMEOUT
      ERROR 22             ;GO TRY NEXT LINE
      BR     1$

4$:   MOV    NRC(R1),(R5)+   ;SAVE THE RECEIVED DATA
      CMP    #RBUF+1000,R5  ;INPUT BUFFER FULL ??
      BNE    3$            ;BR IF NOT

5$:   MOV    #TBUF,R2       ;SET UP POINTER TO OUTPUT BUFFER
      MOV    #RBUF,R1       ;SET UP POINTER TO INPUT BUFFER
      MOVB   (R2),R4        ;SET UP S/B DATA IN R4
      BIC    #177400,R4
      SWAB   R4
```

K01

MAINDEC-11-DZDMM-B
DZDMMB.P11 T44MACY11 27(732) 23-SEP-76 14:33 PAGE 99
SINGLE LINE DATA TEST - ALL SELECTED LINES

SEQ 0217

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

L01

MAINDEC-11-DZDMM-8
DZDHMB.P11 T44

MACY11 27(732) 23-SEP-76 14:33 PAGE 100
SINGLE LINE DATA TEST - ALL SELECTED LINES

SEQ 0218

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		8\$:	NOP		:EXIT POINT

```

013714 000004
*****
TEST 45 BASIC PARITY LOGIC TEST - ALL SELECTED LINES - ODD PARITY
*****
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 LPRO4 L FF2
PEV LPRO5 L FN1

%:
MOV #25,$LPERR ;SET UP ERROR LOOP RETURN
1$: JSR PC,SELINE ;GO SELECT A LINE NO.
BR TST46 ;BR IF ALL SELECTED LINES DONE
2$: MOV #BIT11,(R1) ;CLEAR OUT THE DH11
MOV #260,R4 ;SET UP THE S/B DATA IN R4
BISB LINE,R4
SWAB R4
BISB LINE,(R1) ;SET LINE NO. IN SCR
MOV #377,TDATA1 ;LOAD XMIT BUFFER WITH TEST CHARACTER
MOV #73563,LPR(R1) ;SET UP THE LINE PARAMETERS
MOV #-1,BCR(R1) ;LOAD THE BYTE COUNT REG
MOV #TDATA1,CAR(R1) ;LOAD THE BUS ADDR REG
MOV LINMSK,BKR(R1) ;SET BREAK BIT FOR SELECTED LINE
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

```

NO1

MACY11-11-DZDMM-B
DZDMM-B.P11 T45

MACY11 27(732) 23-SEP-76 14:33 PAGE 102
BASIC PARITY LOGIC TEST - ALL SELECTED LINES - ODD PARITY

SEQ 0220

014044	004767	013056		JSR	PC,SAPS	;AROUND THIS BRANCH IF TIME OUT OCCURS
014050	011103			MOV	(R1),R3	;SAVE THE ERROR PSW
014052	012704	100200		MOV	#100200,R4	;GET THE WAS DATA
014056	156704	014130		BISB	LINE,R4	;SET UP THE S/B DATA
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			BEG	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

014146 000004

```

*****
*TEST 46      MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES
*****
*ST46: SCOPE
*REM %
*TEST ABSTRACT:
*****

```

THIS TEST VERIFIES ALL SELECTED LINES CAN TRANSMIT AND RECEIVE A BINARY COUNT PATTERN WHEN RUN CONCURRENTLY. ALL CHAR LENGTHS (5, 6, 7, AND 8 BITS) ARE TESTED WITH BOTH EVEN AND ODD PARITY CHECKING SPECIFIED. THE TEST ACTUALLY INCLUDES EIGHT SUB-TESTS - THE PARAMETERS FOR EACH SUB-TEST RETRIEVED FROM A TABLE TAGGED "PRTYTB:". REFER TO THIS TABLE TO DETERMINE THE TEST SEQUENCE.

ERRORS:

1. [ERROR 41] IS CALLED TO REPORT FALSE RECEIVER INTRs.
2. [ERROR 42] IS CALLED TO REPORT SILO OVERFLOW ERRORS
3. [ERROR 34] IS CALLED TO REPORT PARITY/DATA ERRORS
4. [ERROR 35] IS CALLED TO REPORT TEST TIMEOUTS

SYNC: (NONE)

DEBUG: (REFER TO TEST 45)

KEY LOGIC: (REFER TO TEST 45)

```

%
MOV      #15, $LPERR      ;SET UP THE ERROR LOOP RETURN
MOV      #PRTYTB+4, R5    ;SET UP POINTER TO TEST PARAMETERS
CLR      STMP7            ;START WITH SUB TEST #00
15:      SUB      #4, R5   ;RESET POINTER FOR ERROR LOOPS
DEC      STMP7           ;RESET SUB TEST # FOR ERROR LOOP
25:      CMP      #PRTYTB+40, R5 ;DONE ALL 9. SUB TESTS ??
BEQ      21$            ;BR IF YES
MOV      #STACK, SP      ;RESET STACK POINTER FOR ERROR LOOPS
MOV      @ADR, R1        ;RESET DEVAR FOR ERROR LOOPS
MOV      (R5)+, STMP6    ;GET THE BYTE COUNT PARAMETER
MOV      (R5)+, STMP5    ;GET THE LINE PARAMETERS
INC      STMP7           ;GENERATE NEW SUB-TEST NO.
MOV      #BIT11, (R1)    ;CLEAR THE DH11
JSR      PC, SUPPAR      ;GO SET UP PARAMETERS
MOV      LPA(R1), STMP0  ;SAVE CURRENT LINE PARAMETERS
MOV      LINSEL, STMP3   ;SAVE SELECTED LINES PARAMETER
JSR      PC, CHPS2       ;GO LOCK OUT INTRs
MOV      DHVCT, R2      ;SET UP THE VECTOR
MOV      #35, (R2)+     ;GO TO 35 ON RCVR INTERRUPT
MOVB     DHRLVL, (R2)
MOV      #100, (R1)     ;ENABLE CHAR AVAIL INTERRUPTS
MOV      LINSEL, LINACT ;FLAG ALL SELECTED LINES ACTIVE

```

014:66 164732

15: 000004

25: 027310

001100

012756

164776

164770

164770

004000

010516

164734

164734

012632

012710

014056

010100

012676

013:42

C02

MAYDEC-11-DZDM-B
DZDM-B.P11MACY11 27(732) 23-SEP-76 14:33 PAGE 104
MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES

SEQ C222

```

43: 014306 016761 012670 000012 MOV LINSEL, BAR(R1) :ACTIVATE ALL SELECTED LINES
44: 014314 116761 164562 164664 MOV STNM, STMP2 :SAVE THE TEST NO.
45: 014322 042767 177400 164656 BIC #177400, STMP2
46: 014330 004767 012542 JSR PC, CHPS1 :GO CLEAR PSW
47: 014334 000167 000232 JMP TS :GO WAIT FOR INTERRUPTS

014340 012706 001100 218: MOV #STACK, SP :RESTORE THE SP
014344 004767 012526 JSR PC, CHPS1 :GO CLEAR PSW
014350 004767 012342 JSR PC, RESTAP :RESTORE TRAP CATCHER
014354 000554 BR TS47 :GO TO NEXT TEST

:RECEIVER INTERRUPT SERVICE ROUTINE

35: 014356 005067 164630 CLR STMP4
014360 005711 (R1) :CHAR AVAIL SET
014364 000404 BMI 45 :BR IF YES

014366 012711 004000 MOV #BIT11, (R1) :CLEAR OUT THE DH11
014370 004041 41 :RCVR FALSE INTERRUPT - CHAR AVAIL NOT SET
014374 000700 BR 25 :GO TRY NEXT SUB TEST

45: 014376 003271 040000 BIT #BIT14, (R1) :SILO OVERFLOW ??
014380 001404 BEQ 55 :BR IF NOT

014382 001041 MOV #BIT11, (R1) :CLEAR OUT THE DH11
014386 001041 42 :SILO OVERFLOW ERROR
014390 000611 BR 25 :GO TRY NEXT SUB TEST

55: 014392 001041 MOV NRC(R1), R3 :GET THE WAS DATA
014396 001041 MOV R3, R2 :EXTRACT AND SAVE LINE NO.
014400 001041 SWAB R2
014404 001041 BIC #177760, R2
014408 001041 R2 :GENERATE TABLE OFFSETR
014412 001041 INC STMP4
014416 001041 CMP #101, STMP4
014420 001041 BEQ 35
014424 001041 BIT LINSEL(R2), LINSEL :IS THIS ONE OF THE SELECTED LINES?
014428 001041 BEQ 55 :IF NOT, THIS IS GARBAGE. SO CHECK NEXT SILO ENTRY.
014432 001041 MOV R2, STMP4
014436 001041 ASR STMP4
014440 001041 CMP RBUF(R2), R3 :CORRECT DATA RECEIVED ??
014444 001041 BEQ 65 :BR IF YES

014446 001041 JSR PC, SAPS :SAVE THE ERROR PSW
014450 001041 MOV #BIT11, (R1) :CLEAR OUT THE DH11
014454 001041 MOV RBUF(R2), R4 :SET UP S B DATA
014458 001041 ADD #NRC, R1 :SET UP WAS ADDRESS
014462 001041 ADD #RBUF, R2 :SET UP S B ADDRESS
014466 001041 JSR PC, SUER2A :GO SET UP ERROR INFO
014470 001041 JSR R5, SUNUM :PUT LINE NO. IN MESSAGE
014474 001041 STMP4
014478 001041 EM34+51
014482 001041 JSR R5, SUNUM :PUT SUBTEST NO. IN MESSAGE
014486 001041 STMP7
014490 001041 EM34+67
014494 001041 ERROR 34 :PARITY DATA COMPARE ERROR

```

23-SEP-76 14:33 PAGE 105
 MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES

 MACV11 27(732) 23-SEP-76 14:33 PAGE 105
 MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES

500 0223

014546	000613		BR	25		:GO TRY NEXT SUBTEST
014550	105262	036112	65:	INCB	RBUF(R2)	:GENERATE NEW RCVD DATA
014554	005362	027452		INC	MULPTB(R2)	:COUNT ONE BYTES RECEIVED
014560	001303			BNE	615	:BR IF NOT DONE
014562	046267	027410	012660	BIC	LINBIT(R2),LINACT	:FLAG THIS LINE DONE
014570	000002		615:	RTI		:RETURN TO WAIT ROUTINE
						:WAIT ROUTINE
014572	012767	000002	013440	75:	MOV	#2,TIMEA
014573	005067	013436			CLR	TIMEB
014574	005761	000012		85:	TST	BAR(R1)
014575	001413				BEQ	95
014576	004767	012126			JSR	PC,TIMEIT
014577	000772				BR	95
						:INIT TIMER A
						:INIT TIMER B
						:ALL LINES DONE XMITTING ??
						:BR IF YES
						:CALL THE TIMER
						:TIMER ROUTINE WILL MOVE RETURN PC
						:AROUND THIS BRANCH IF TIME OUT OCCURS
014623	016167	000012	164362		MOV	BAR(R1),STMP3
014626	012711	004000			MOV	#BIT11,(R1)
014632	104035				ERROR	35
014634	000167	177336			JMP	25
						:SAVE THE ACTIVE LINES FLAG
						:CLEAR OUT THE DH11
						:TIMED OUT WAITING FOR TRANSMITTERS TO FINISH
						:GO TRY NEXT SUBTEST
014640	012767	000001	013372	95:	MOV	#1,TIMEA
014646	005067	013370			CLR	TIMEB
014652	005767	012572		105:	TST	LINACT
014655	001411				BEQ	115
014656	004767	012060			JSR	PC,TIMEIT
014664	000772				BR	105
						:INIT TIMER A
						:INIT TIMER B
						:ALL CHARS RECEIVED ?
						:BR IF YES
						:CALL THE TIMER
						:TIMER ROUTINE WILL MOVE RETURN PC
						:AROUND THIS BRANCH IF TIME OUT OCCURS
014666	016767	012556	164314		MOV	LINACT,STMP3
014674	012711	004000			MOV	#BIT11,(R1)
014700	104035				ERROR	35
014702	000167	177270		115:	JMP	25
						:SET UP ACTIVE LINE PARAMETER
						:CLEAR OUT THE DH11
						:SILO EMPTY TIMEOUT
						:GO TRY NEXT SUB TEST

MAINDEC-11-DZDMM-B
DZDMM.P11 T47

MACY11 27(732) 23-SEP-76 14:33 PAGE 106
AUTO ECHO TEST 1 - ALL SELECTED LINES

SEQ 0224

```

*****
*TEST 47      AUTO ECHO TEST 1 - ALL SELECTED LINES
*****
*ST47: 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 "AETABC:" (0'S DATA)
AND REPEAT STEPS 2 THRU 9.

ERRORS:

1. [ERROR 24] IS CALLED TO REPORT ALL ERRORS

SYNC: M7277 SH4 LOAD BAR LB+HB L CN2

DEBUG:

1. IF ALL LINES FAIL, SUSPECT EITHER THE M7277 OR M7289
2. IF ONLY ONE LINE FAILS SUSPECT THE M7288
3. LOOP ON THE FAILING LINE AND TRACK BACK THROUGH THE KEY LOGIC.

KEY LOGIC:

```

M7277 SH3 AE GO L CS2
          7402 "OR" GATE CHIPS E38 OR E41
          74157 MUX CHIPS E39 OR 342
          SH4 E35 - PIN 2 STUCK LOW

```

F02

 MAINDEC-11-DZDMM-8
 220MB.P11

 MACY11 27(732) 23-SEP-76 14:33 PAGE 107
 AUTO ECHO TEST 1 - ALL SELECTED LINES

SEQ C225

 M7289 SH3 AE GO L EK1
 AE SCAN MUX E22 PIN 10
 SH4 SAMPLE STATUS H E21-12
 M7288 SH45. 7. 9. 11
 AE ENABLE "NN" H CONTROL FLOPS
 74174 CHIPS PIN 15

014710	012767	014772	164172	MOV	#25,SLPERR	:SET UP ERROR LOOP RETURN
014716	005067	164266		CLR	STMP3	:INIT I/O DATA FLAG
014720	012711	004000		MOV	#BIT11,(R1)	:CLEAR THE DH11
014726	012705	027310		MOV	#AETAB,R5	:GET POINTER TO AUTO ECHO DATA TABLE
014730	005067	012250		CLR	LMSK1	:INIT BIT TEST MARKER
014736	000261		75:	SEC		:SET "C" BIT FOR MARKER
014740	000401			BR	135	:GO SHIFT MASK
014744	000241		15:	CLR		:INIT THE "C" BIT
014748	006167	012236	135:	ROL	LMSK1	:SHIFT BIT MARKER
014750	001407			BEQ	125	:BR IF DONE ALL LINES
014752	012504			MOV	(R5)+,R4	:SET UP THE S/B DATA
014756	036767	012226	012220	BIT	LMSK1,LINSEL	:TEST THIS LINE ?
014760	001767			BEQ	15	:BR IF NOT
014764	004767	007512		JSR	PC,SELINE	:GO SELECT A LINE TO TEST
014768			115:			
014772	000522		125:	BR	65	:BR IF ALL SELECTED LINES TESTED
014776	004767	007656	25:	JSR	PC,CLCABC	:GO CLEAR "CAR" AND "BCR" MEMORIES
014780	116711	013210		MOV	LINE,(R1)	:SET SELECT BITS IN SCR REG
014784	012761	177777	000010	MOV	#-1,BCR(R1)	:SET UP TO XFER ONE CHAR
014788	010561	000006		MOV	R5,CAR(R1)	:SET UP THE BUS ADDRESS REG
014792	162761	000002	000006	SUB	#2,CAR(R1)	:CORRECT BUS ADDRESS
014796	012767	000100	164170	MOV	#100,STMP7	:COUNT 64 CHARS TO BE RECEIVED IN AUTO ECHO
014800	005067	164162		CLR	STMP6	:INIT CHAR COUNTER
014804	012761	133503	000004	MOV	#133503,LPR(R1)	:SET UP LINE PARAMETER REG
014808	016761	012136	000012	MOV	LINMSK,BAR(R1)	:ACTIVATE THE LINE
014812						
014816	012767	000002	013162	MOV	#2,TIMEA	:INIT TIMER A
014820	005067	013160		CLR	TIMEB	:INIT TIMERB
014824	105711		35:	TSTB	(R1)	:CHAR AVAIL SET ??
014828	100427			BMI	45	:BR IF YES
014832	004767	011652		JSR	PC,TIMEIT	:CALL THE TIMER
014836	000773			BR	35	:TIMER ROUTINE WILL MOVE RETURN PC
014840						:AROUND THIS BRANCH IF TIME OUT OCCURS
014844						
014848	012026			JSR	PC,SAPS	:SAVE THE ERROR PSW
014852	005061	000004		CLR	LPR(R1)	:TURN OFF AUTO ECHO MODE
014856	010102			MOV	R1,R2	:MAKE REGADR = DEVADR
014860	011103			MOV	(R1),R3	:GET THE WAS DATA
014864	042703	100000		BIC	#BIT15,R3	:CLEAR JUNK BIT
014868	012704	000200		MOV	#200,R4	:SET UP S/B DATA
014872	156704	013066		BISB	LINE,R4	
014876	004767	007224		JSR	PC,SUER2A	:GO SET UP ERROR INFO
014880	004567	007440		JSR	R5,SUNUM	:GO SET LINE NO. IN MSG
014884	030212			LINE		
014888	032164			EM24+35		
014892	104024			ERROR	24	:DATA AVAIL FAILED TO SET ON TIME

G02

MAINDEC-11-DZDMM-B
DZDMMB.P11 T47MACY11 27(732) 23-SEP-76 14:33 PAGE 108
AUTO ECHO TEST 1 - ALL SELECTED LINES

SEQ 0226

45: 015142	000677		BR	15	;GO TRY NEXT LINE
015144	005267	164046	45: INC	STMP6	;COUNT ONE CHAR RECVD
015150	016103	000002	MOV	NRC(R1),R3	;GET THE WAS DATA
015154	020304		CMP	R3,R4	;WAS CHAR AUTO ECHOED CORRECTLY ?
015156	001417		BEG	55	;BR IF YES
015160	004767	011742	JSR	PC,SAPS	;SAVE THE ERROR PSW
015164	005061	000004	CLR	LPR(R1)	;DISABLE AUTO ECHO
015170	010102		MOV	R1,R2	;SET UP REGADR
015172	062702	000002	ADD	#NRC,R2	
015176	004767	007152	JSR	PC,SUER2A	;GO SET UP ERROR INFO
015202	004567	007366	JSR	R5,SUNUM	;PUT LINE NO. IN ERROR MSG
015206	030212		LINE		
015210	032164		EM24+35		
015212	104024		ERROR	24	;CHAR AUTO ECHOED INCORRECTLY
015214	000652		BR	15	;GO TRY NEXT LINE
015216	005367	163776	55: DEC	STMP7	;COUNT ONE CHAR READ OUT OF 54
015222	003317		BGT	35	;BR IF NOT LAST ONE
015224	100646		BMI	15	;BR IF LAST ONE READ
015226	042761	100000 000004	BIC	#BIT15,LPR(R1)	;DISABLE AUTO ECHO
015234	000712		BR	35	;GO READ LAST CHAR
015236	005167	163746	65: COM	STMP3	;TOGGLE I/O FLAG
015242	001406		BEG	TST50	;BR IF DONE BOTH I/O DATA
015244	005067	012742	CLR	LINE	;INIT LINE NO TO CO
015250	012705	027350	MOV	#AETAB0,R5	;SET POINTER TO O'S TABLE
015254	000167	177452	JMP	75	;REPEAT TEST FOR ZERO PATTERNS

4605
4606
4607
4608 015260 000004
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660

```

*****
: *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 AROUND A BINARY COUNT TEST PATTERN IN NON-AUTO ECHO MODE AND THE A.E. TEST LINE IS TESTED FOR ALL 1'S DATA ONLY.

ERRORS:

1. [ERROR 32] IS CALLED TO REPORT A.E. TEST TIMEOUTS
2. [ERROR 31] IS CALLED TO REPORT ALL DATA COMPARE ERRORS

SYNC: M7277 SH4 LOAD BAR LB+HB L CN2

DEBUG:

REFER TO TEST 47

KEY LOGIC:

REFER TO TEST 47

015262	012767	015336	163620	%	MOV	#25,\$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
015352	156711	012634			BISB	LINE,(R1)	:ALL LINES OTHER THAN THE SELECTED ONE
015356	010561	000006			MOV	R5,CAR(R1)	:SELECT THE LINE FOR A.E. TEST
015362	162761	000002	000006		SUB	#2,CAR(R1)	:SET BUS ADDR TO XMIT TEST CHAR
015370	012761	177777	000010		MOV	#-1,BCR(R1)	:CORRECT THE ADDRESS
							:XMIT ONE CHAR ON THIS LINE

MAINDEC-11-DZDMM-B
 DZDMMB.P11 T50

 MACY11 27(732) 23-SEP-76 14:33 PAGE 110
 AUTO ECHO TEST 2 - ALL SELECTED LINES

SEG 0228

4661	015376	012761	133503	000004		MOV	#133503,LPR(R1)	;DO IT AT 9600 BAUD/8 BITS
4662	015412	011676	163472	163574		MOV	\$STNM,\$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			CLR	TIMEB	;INIT TIMER B
4669	015446	016103	000002		3\$:	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	\$LINSEL(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

J02

 MAINDEC-11-DZDMM-B
 DZDMMB.P11 T50

 MACY11 27(732) 23-SEP-76 14:33 PAGE 111
 AUTO ECHO TEST 2 - ALL SELECTED LINES

SEQ 0229

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	015652	105262	036112	6\$:	INCB	RBUF(R2)	; GENERATE NEXT EXPECTED DATA ON THIS LINE -
4723	015656	001262			SNE	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			SNE	8\$	; BR IF NOT
4730	015720	000167	177360		JMP	1\$	; GO TEST NEXT LINE
4731	015724	000167	177504	8\$:	JMP	21\$	; GO EMPTY IT

K02

MAINDEC-11-DZDMM-B
DZDMMB.P11

TS1

MACY11 27(732) 23-SEP-76 14:33 PAGE 112
AUTO ECHO TEST 3 - ALL SELECTED LINES

SEQ 0230

4732
4733
4734
4735 015730 000004
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787

```

*****
:TEST 51      AUTO ECHO TEST 3 - ALL SELECTED LINES
*****
TEST51:  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

```

%
1$:  MOV    #16,SLPERR      ;SET UP THE ERROR LOOP RETURN
      MOV    #BIT11,(R1)    ;CLEAR OUT THE DH11
      MOV    #20,R5         ;INIT COUNTER TO SET UP 16. LINES
      MOV    #AETAB,R2      ;SET UP POINTER TO AUTO ECHO TEST DATA
      MOV    #RCNT,R3       ;R3 POINTS TO TABLE OF CHAR COUNTERS
2$:  MOV    R2,CAR(R1)      ;SET UP BUS ADDRESS REG
      MOV    #-1,BCR(R1)    ;SET UP BYTE COUNT REG
      MOV    #131403,LPR(R1);SET UP LINE PARAMETERS
      CLR    (R3)+          ;CLEAR A COUNTER
      ADD    #2,R2          ;UPDATE POINTERS
      INC    (R1)           ;SELECT NEXT LINE
      DEC    R5             ;COUNT ONE DONE
      BNE    2$            ;BR TILL 16. DONE
      MOVB   STSTNM,STMP2   ;SAVE THE TEST NO.
      BIC    #177400,STMP2
      MOV    LINSEL,LINACT  ;SET FLAG TO INDICATE ALL 16. ACTIVE
      MOV    LINSEL,BAR(R1);ACTIVATE ALL XMITTERS

      MOV    #2,TIMER A     ;INIT TIMER A
      CLR    TIMEB          ;INIT TIMERB
3$:  CLR    STMP4           ;
      TSTB   (R1)           ;CHAR AVAIL SET YET ?
      BMI    4$            ;BR IF YES
      JSR    PC,TIMEIT     ;CALL THE TIMER

```

MAINDEC-11-DZDMM-B
DZDMMB.P11 T51MACY11 27(732) 23-SEP-76 14:33 PAGE 113
AUTO ECHO TEST 3 - ALL SELECTED LINES

SEQ 0231

```

4798 016072 000771          BR      3$          ;TIMER ROUTINE WILL MOVE RETURN PC
4799                                     ;AROUND THIS BRANCH IF TIME OUT OCCURS
4790
4791 016074 016167 000004 162100      MOV     LPR,R1),STMP0      ;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$:  MOV     NRC(R1),R3          ;GET THE WAS DATA
4796 016112 010302          MOV     R3,R2          ;BUILD AND SAVE LINE NO.
4797 016114 000302          SWAB    R2
4798 016116 042702 177760          BIC     #177760,R2
4799 016122 010267 163070          MOV     R2,STMP6          ;SAVE THE LINE NO.
4800 016126 006302          ASL     R2          ;GENERATE TABLE OFFSET
4801 016130 005267 163056          INC     STMP4
4802 016134 022767 000101 163050      CMP     #101,STMP4
4803 016142 001745          BEQ     3$
4804 016144 036267 030136 011030      BIT     $LNSEL(R2),LNSEL      ;IS THIS ONE C THE SELECTED LINES?
4805 016152 001755          BEQ     4$          ;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          STMP6
4817 016220 032520          EM31+45
4818 016222 104031          ERROR    31
4819 016224 000415          BR      TST52          ;DATA COMPARE ERROR
4820                                     ;EXIT TEST ON ERROR
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     STMP6,(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

```


4827
4828
4829
4830 016260 000004
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882

```

::*****
: *TEST 52      BREAK BIT TEST - ALL SELECTED LINES
: *****
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

```

4883
4884
4885
4886 016262 012767 016342 162620 % MOV #2$, $LPERR ;SET UP ERROR LOOP RETURN
4887 016270 012705 027512 MOV #BRKTAB, R5 ;SET UP POINTER TO BREAK DATA TABLE
4888 016274 005067 010706 CLR LMSK1 ;INIT BIT TEST MASK
4889 016300 000261 SEC ;SET BIT MARKER IN "C"
4890 016302 000401 BR 12$ ;GO SHIFT MASK
4891 016304 000241 1$: CLC ;INIT THE "C" BIT
4892 016306 006167 010674 12$: ROL LMSK1 ;SHIFT TEST MARKER
4893 016312 001411 BEQ 11$ ;BR IF ALL LINES DONE
4894 016314 012504 MOV (R5)+, R4 ;GET TEST DATA FOR THIS LINE
4895 016316 036767 010664 010656 BIT LMSK1, LINSEL ;LINE SELECTED ?
4896 016324 001767 BEQ 1$ ;BR IF NOT
4897 016326 004767 006150 JSR PC, SELINE ;GO SELECT A LINE TO TEST
4898 016332 000401 BR 11$ ;BR IF DONE ALL SELECTED LINES
4899 016334 000402 BR 2$ ;GO TEST THE SELECTED LINE
4900 016336 000167 000454 11$: JMP 9$ ;GO EXIT TEST
4901 016342 012711 004000 2$: MOV #BIT11, (R1) ;CLEAR THE DH11
4902 016346 004767 006302 JSR PC, CLCABC ;GO CLR THE "CAR" AND "BCR" MEMORIES
4903 016352 116711 011634 MOVB LINE, (R1) ;SELECT THE LINE
4904
4905 016356 012761 030246 000006 MOV #TNUL, CAR(R1) ;SET UP TO OUTPUT TWO NULL CHARS
4906 016364 012761 177776 000010 MOV #-2, BCR(R1) ;SET BYTE COUNT TO 2
4907 016372 012761 033503 000004 MOV #33503, LPR(R1) ;SET UP LINE PARAMETERS
4908 016400 016761 010600 000012 MOV LINMSK, BAR(R1) ;ACTIVATE SELECTED LINE
4909
4910 016406 012767 000001 011624 MOV #1, TIMEA ;INIT TIMER A
4911 016414 005067 011622 CLR TIMEB ;INIT TIMER B
4912 016420 122761 000002 000017 3$: CMPE #2, SSR+1(R1) ;TWO CHARS RECEIVED ??
4913 016426 001432 BEQ 4$ ;BR IF YES
4914 016430 004767 010310 JSR PC, TIMEIT ;CALL THE TIMER
4915 016434 000771 BR 3$ ;TIMER ROUTINE WILL MOVE RETURN PC
4916 ; AROUND THIS BRANCH IF TIME OUT OCCURS
4917
4918 016436 004767 010464 JSR PC, SAPS ;SAVE THE ERROR PSW
4919 016442 010467 162536 MOV R4, $TMP1 ;SAVE S/B DATA
4920 016446 010102 MOV R1, R2 ;SET UP REGADR
4921 016450 062702 000016 ADD #SSR, R2
4922 016454 011203 MOV (R2), R3 ;GET THE WAS DATA
4923 016456 042703 100377 BIC #100377, R3 ;CLEAR JUNK
4924 016462 012704 000002 MOV #2, R4 ;SET UP S/B DATA
4925 016466 000304 SWAB R4
4926 016470 004767 005660 JSR PC, SUER2A ;GO SET UP ERROR INFO
4927 016474 004567 006074 JSR R5, SUNUM ;GO PUT LINE NO. IN MESSAGE
4928 016500 030212 LINE
4929 016502 032223 EM25+34
4930 016504 016704 162474 MOV $TMP1, R4 ;RESTORE S/B DATA
4931 016510 104025 ERROR 25 ;TIMED OUT WAITING FOR TWO NULLS
4932 016512 000674 BR 1$ ;GO TRY NEXT LINE
4933
4934 016514 012711 004000 4$: MOV #BIT11, (R1) ;CLEAR THE INTERFACE
4935 016520 116711 011466 MOVB LINE, (R1) ;SELECT THE LINE
4936 016524 012761 037112 000006 MOV #TBUF, CAR(R1) ;SET UP BUS ADDRESS REG FOR XMITTR
4937 016532 012761 177400 000010 MOV #-400, BCR(R1) ;SET BYTE COUNT TO XMIT 256(10) CHARS
4938 016540 012761 033503 000004 MOV #33503, LPR(R1) ;SET UP LINE PARAMETERS

```

MAYNCEC-11-02DM-8
022448.P11 T52MACY11 27(732) 23-SEP-76 14:33 PAGE 116
BREAK BIT TEST - ALL SELECTED LINES.

SEG 0234

```

48339 016546 016761 010432 000014      MOV      LINMSK,BAR(R1)  :SET BREAK BIT FOR ACTIVE LINE
48340 016554 016761 010424 000012      MOV      LINMSK,BAR(R1)  :ACTIVATE - IF SELECTED LINE
48341 016562 012767 010005 011450      MOV      R5,TIMER      :INIT TIMER A
48342 016570 005067 011446              :INIT TIMER B
48343 016574 005761 000012      53:      BRR,R1        :BAR BIT CLEARED ??
48344 016600 001426              :BR IFD YES
48345 016602 004767 010136      :CALL THE TIMER
48346 016606 000772              :TIMER ROUTINE WILL MOVE RETURN PC
48347              :ROUND THIS BRANCH IF TIME OUT OCCURS

48348 016610 004767 010312      JSR      PC,SAPS        :SAVE THE ERROR PSW
48349 016614 010467 162364      MOV      R4,$TMP1      :SAVE THE S/B DATA
48350 016620 010102              :SET UP REGADR
48351 016622 002702 000012      MOV      R1,R2
48352 016626 011203              :GET THE WAS DATA
48353 016630 005004              :SET UP S/B DATA
48354 016632 004767 005516      JSR      PC,SUER2A      :GO SET UP ERROR INFO
48355 016636 004567 005732      JSR      R5,SUNUM      :PUT LINE NO IN MESSAGE
48356 016642 000212      LINE
48357 016644 002223      EM25+34
48358 016646 016704 162332      MOV      $TMP1,R4      :RESTORE THE S/B DATA
48359 016652 104025      ERROR      25        :BAR BIT FAILED TO CLEAR
48360 016654 000613      BR        15          :GO TRY NEXT LINE

48361 016656 122761 000001 000017 65:  CMPB     #1,SSR+1(R1)   :ONE CHAR RECEIVED ?
48362 016664 001430      BEQ      75          :BR IF YES

48363 016666 004767 010234      JSR      PC,SAPS        :SAVE THE ERROR PSW
48364 016672 010467 162306      MOV      R4,$TMP1      :SAVE THE S/B DATA
48365 016676 010102              :SET UP REGADR
48366 016700 062702 000016      ADD      #SSR,R2
48367 016704 011203              :GET THE WAS DATA
48368 016706 042703 100377      MOV      (R2),R3      :CLEAR JUNK
48369 016712 012704 000001      BIC      #100377,R3   :SET UP S/B DATA
48370 016716 000304      MOV      #1,R4
48371 016720 004767 005430      SWAB     R4
48372 016724 004567 005644      JSR      PC,SUER2A      :GO SET UP ERROR INFO
48373 016730 000212      JSR      R5,SUNUM      :GO PUT LINE NO. IN MESSAGE
48374 016732 002223      LINE
48375 016734 016704 162244      EM25+34
48376 016736 104025      MOV      $TMP1,R4      :RESTORE THE S/B DATA
48377 016742 000167 177336      ERROR      25        :FAILED TO RECEIVE THE ONE CHAR
48378              JMP      15          :GO TRY NEXT LINE

48379 016746 016103 000002      75:      MOV      NRC(R1),R3  :GET THE WAS DATA
48380 016752 020304      CMP      R3,R4        :WAS IT A BREAK CHAR ?
48381 016754 001002      BNE     95          :BR IF NOT CORRECT
48382 016756 000167 177322      JMP      15          :GO TEST NEXT LINE

48383 016762 004767 010140      85:      JSR      PC,SAPS        :SAVE THE ERROR PSW
48384 016766 010102      MOV      R1,R2        :SET UP REGADR
48385 016770 062702 000002      ADD      #NRC,R2
48386 016774 004767 005354      JSR      PC,SUER2A      :GO SET UP ERROR INFO
48387 016780 004567 005570      JSR      R5,SUNUM      :PUT LINE NO IN MESSAGE
48388 016784 000212      LINE

```

000000-11-00000-8
 000000.P11 152

SEP 22 1955

[illegible]

17-256

95:

EM25-34
ERROR 25
JMP 13
NCF

```

:INCORRECT DATA RECEIVED
:GO TRY NEXT LINE
:EXIT THIS TEST

```

017020 000004

```

*****
:TEST 53      HALF DUPLEX TEST - ALL SELECTED LINES
*****
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
017030 004767 005446
017034 000477
017036 012711 004000
017042 004767 005606
017046 156711 011140
017052 012761 037112 000006

```

%
15: MOV #25,SLPERR ;SET UP ERROR LOOP RETURN
     JSR PC,SELINE
     BR TST54      ;BR IF ALL LINES TESTED
25: MOV #BIT11,(R1) ;CLEAR THE INTERFACE
     JSR PC,CLCABC  ;GO CLR THE "CAR" AND "BCR" MEMORIES
     BISB LINE,(R1) ;SELECT THE LINE
     MOV #TBUF,CAR(R1) ;POINT TO XMIT BUFFER

```

E03

 MACV11-11-0004M-B
 000000.P11 T53

 MACV11 27(732) 23-SEP-76 14:33 PAGE 119
 HALF DUPLEX TEST - ALL SELECTED LINES

SEC 0237

017060	012761	177400	000010	MOV	#400,BCR(R1)	:XMIT 256(10) CHARS
017060	012761	073503	000004	MOV	#73503,LPR(R1)	:SET UP THE LINE PARAMETERS
017060	012761	010104	000012	MOV	LINMSK,BAR(R1)	:ACTIVATE THE SELECTED LINE
017060	012761	000001	01:130	MOV	#1,TIMEA	:INIT TIME A
017060	012761	000002		CLR	TIMEB	:INIT TIME B
017060	012761	000003	35:	TST	BAR,R1	:WAIT FOR XMITTR TO FINISH
017060	012761	000004		BNE	45	:BR IF XMITTR FINISHED
017060	012761	000005		JSR	PC,TIMEIT	:CALL TIMER
017060	012761	000006		BR	35	:TIMER WILL MOVE RETURN PC AROUND
017060	012761	000007				:THIS BRANCH IF TIMEOUT OCCURS
017060	012761	000008		JSR	PC,SAPS	:SAVE THE ERROR PSW
017060	012761	000009		MOV	BAR(R1),R3	:GET THE WAS DATA
017060	012761	000010		MOV	R1,R2	:SET UP REGADR
017060	012761	000011		ADD	#BAR,R2	
017060	012761	000012		CLR	R4	:SET UP NEW S/B DATA
017060	012761	000013		JSR	PC,SUER2A	:GO SET UP THE ERROR INFO
017060	012761	000014		JSR	RS,SUNUM	:PUT LINE NO. IN MESSAGE
017060	012761	000015		LINE		
017060	012761	000016		EM26+37		
017060	012761	000017		ERROR	26	:BAR BIT FAILED TO CLEAR ON TIME
017060	012761	000018		BR	15	:GO TRY NEXT LINE
017060	012761	000019	45:	TSTB	(R1)	:CHAR AVAIL SET ??
017060	012761	000020		BPL	15	:BR IF NOT IT SHOULDN'T BE
017060	012761	000021		JSR	PC,SAPS	:SAVE THE ERROR PSW
017060	012761	000022		MOV	R1,R2	:SET UP REGADR
017060	012761	000023		MOV	(R1),R3	:GET WAS DATA
017060	012761	000024		BIC	#BIT15,R3	:CLEAR JUNK BIT
017060	012761	000025		MOV	LINE,R4	:SET UP S/B DATA
017060	012761	000026		JSR	PC,SUER2A	:GO SETUP ERROR INFO
017060	012761	000027		JSR	RS,SUNUM	:PUT LINE NO. IN MSG
017060	012761	000028		LINE		
017060	012761	000029		EM26+37		
017060	012761	000030		ERROR	26	:HALF DUPLEX FAILED TO BLIND RECV
017060	012761	000031		BR	15	:GO SELECT NEXT LINE

F03

MACV 11 27.732 23-SEP-76 14:33 PAGE 120
 VERIFY THAT 0 ERRUN CAN SET PROPERLY - ALL SELECTED LINES

MACV 11 27.732 23-SEP-76 14:33 PAGE 120
 VERIFY THAT 0 ERRUN CAN SET PROPERLY - ALL SELECTED LINES

SEQ 0238

017234 000004

 :TEST 54 VERIFY THAT OVERRUN CAN SET PROPERLY - ALL SELECTED LINES
 :*****

TEST54: 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 CALLED 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 (BLF OR LINE NN)

%

017236 012767 017252 161644
 017244 004767 005232
 017250 000512
 017252 012711 004000
 017256 116711 010730

15: MOV #25, \$LPERR ;SET UP ERROR LOOP RETURN
 JSR PC, SELINE ;GO SELECT A LINE # TO TEST
 BR TS155 ;BR IF DONE ALL SELECTED LINES
 25: MOV #BIT11, (R1) ;CLEAR OUT THE DH11
 MOVB LINE, (R1) ;SELECT THE LINE TO TEST

MAINDEC-11-02DMM-8
02DMMB.P11 TS4

MACY11 27(732) 23-SEP-76 14:33 PAGE 121
VERIFY THAT OVERRUN CAN SET PROPERLY - ALL SELECTED LINES

SEQ 0239

Address	Instruction	Comment
017262	MOV #TBUF,CAR(R1)	;SET UP CURRENT ADDRESS
017270	MOV #-68,BCR(R1)	;SET UP BYTE COUNT REG
017276	MOV #33503,LPR(R1)	;DO IT AT 9600 BAUD - 8 BITS
017304	MOV LINMSK,BAR(R1)	;ACTIVATE THE SELECTED LINE
017312	MOV #1,TIMEA	;INIT TIMERS A AND B
017320	CLR TIMEB	
017324	TST (R1)	;TRANSMITTER DONE ??
017326	BMI 4\$	;BR IF YES
017330	JSR PC,TIMEIT	;CALL TIMER
017334	BR 3\$	;BR IF NO TIMEOUT
017336	JSR PC,SAPS	;GO SAVE PSW
017342	MOV (R1),R3	;GET THE WAS DATA
017344	BIC #77760,R3	;CLEAR UNINTERESTING BITS
017350	MOV LINE,R4	;SET UP S/B DATA
017354	BIS #BIT15,R4	
017360	MOV R1,R2	;SET UP REGADR
017362	JSR PC,SUER2A	;GO SET UP ERROR INFO
017366	JSR R5,SUNUM	;PUT LINE NO. IN MESSAGE HEADER
017372	LINE	
017374	EM50+53	
017376	ERROR	
017400	BR 5\$	;REPORT XMIT DONE TIME OUT
017402	BR 1\$	;GO TRY NEXT LINE
017402	MOV #65,\$TMP1	;SET UP TO READ 65. WORDS FROM SILO
017410	MOV LINE,R4	;SET UP S/B DATA
017414	SWAB R4	
017416	BISB #65,R4	
017422	BIS #BIT15+BIT14,R4	;PUT IN OVERRUN AND VALID DATA BITS
017426	MOV NRC(R1),R3	;GET WAS DATA FROM SILO
017432	DEC \$TMP1	;COUNT ONE WORD READ
017436	BNE 5\$	;BR TIL 65. READ
017440	CMP R3,R4	;WAS DATA AND OVERRUN CORRECT ??
017442	BEG 1\$	;BR IF YES TRY NEXT SELECTED LINE
017444	JSR PC,SAPS	;GO SAVE PSW
017450	MOV R1,R2	;SET UP REGADR
017452	ADD #NRC,R2	
017456	JSR PC,SUER2A	;GO SET UP ERROR INFO
017462	JSR R5,SUNUM	;GO PUT LINE NO. IN MSG HDR
017466	LINE	
017470	EM56+42	
017472	ERROR	
017474	BR 5\$	;OVERRUN OR DATA INCORRECT
017476	BR 1\$	;GO TEST NEXT SELECTED LINE

017500	032777	C02000	161432		BIT	#BIT10,DSWR	:CHECK DM11-BB?
017506	001002				BNE	ZS	:BRANCH IF YES.
017510	000167	001024			JMP	ENDA	:OTHERWISE, GET OUT.
017514	012700	030034		ZS:	MOV	#DMADRS,R0	:RO POINTS TO BEGINNING OF DM ADDRESS TABLE.
017520	016767	010464	161454		MOV	DHNMJ,STMPO	
017526	006367	161450			ASL	STMPO	:DOUBLE TMPO
017532	066700	161444			ADD	STMPO,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

MAINDEC-11-DZOHM-B
DZOHMB.P11 T55

MACY11 27(732) 23-SEP-76 14:33 PAGE 123
ABBREVIATED DM11-BB MODEM CONTROL DIAGNOSTIC

SEQ 0241

5252	017566	020103		CMP	R1,R3	:IF LINE NUMBER=SELECTED LINE NUMBER,
5253	017570	001002		BNE	MUX11C	:EXCEPT LINE ENABLE FUNCTION FLIP FLOP
5254	017672	012705	000001	MOV	#LINENA,R5	
5255						:TO BE SET
5256	017576	020504		MUX11C: CMP	R5,R4	:COMPARE EXPECTED AND RECEIVED
5257	017700	001403		BEQ	MUX11D	:RESULTS
5258	017702	104401		TYPE		
5259	017704	035755		MSG4		:ERROR
5260	017706	000000		HALT		
5261	017710	052777	000400 010262	MUX11D: BIS	#STEP,JDHMCSS	:EXAMINE NEXT LINE
5262	017716	005302		DEC	R2	
5263	017720	001353		BNE	MUX11B	
5264	017722	005005		CLR	R5	
5265	017724	010177	010250	MUX11E: MOV	R1,JDHMCSS	
5266	017730	010103		MOV	R1,R3	:SET LINE COUNTER TO SELECTED LINE
5267	017732	005077	010244	CLR	JDHMLSR	:CLEAR LINE ENABLE FLIP FLOP
5268	017736	105227	000000	INCB	#0	:DELAY FOR CABLE
5269	017742	001375		BNE	.-4	:DITTO
5270	017744	017704	010232	MOV	JDHMLSR,R4	:READ LINE STATUS REGISTER
5271	017750	005704		TST	R4	:WAS LINE ENABLE FUNCTION FLIP FLOP
5272	017752	001710		BEQ	DOIT11	:CLEARED
5273	017754	104401		TYPE		
5274	017756	035755		MSG4		:ERROR
5275	017760	000000		HALT		

```

5278      ;*****
5279      ;*TEST 56      DM11-BB DIAGNOSTIC CONTINUED
5280      ;*****
5281      017762 000004  ;TEST56: SCOPE
5282      ;REM %
5283      ;TEST ABSTRACT:
5284      ;*****
5285
5286      ;THIS TEST VERIFIES THAT CLEAR TO SEND AND CARRIER ARE SET
5287      ;IF "LINE ENABLE" AND TERMINAL ARE SET FOR THE SELECTED LINE.
5288
5289      ERRORS:
5290      ;*****
5291
5292      ;IF ANY ERRORS OCCUR, RUN THE DM11-BB MODEM CONTROL DIAGNOSTIC.
5293      DZDHK.
5294
5295      %
5296      017764 005077 010212  DOIT15: CLR      ;DHMLSR      ;CLEAR LINE STATUS REGISTER
5297      017770 004767 004506      JSR      PC,SELINE ;GO SELECT A LINE.
5298      017774 000467      BR      TS157      ;;BR IF DONE ALL SELECTED LINES
5299      017776 005077 010176      MUX15: CLR      ;DHMCSR      ;CLEAR CONTROL REGISTER
5300      020002 042767 000340      BIC      #340,PS ;ENABLE INTERRUPTS
5301      020010 116701 010176      MOV      LINE,R1
5302      020014 012702 000020      MUX15A: MOV     #16,R2      ;16 LINES
5303      020020 010177 010154      MOV      R1,;DHMCSR ;SELECT A LINE
5304      020024 012777 000003      MOV      #LINENA+TRMRDY,;DHMLSR ;SET LINE ENABLE +TRMRDY
5305      020032 005077 010142      CLR      ;DHMCSR      ;CLEAR CONTROL REGISTER
5306      020036 005005      MUX15B: CLR      R5      ;CLEAR EXPECTED RESULT
5307      020040 017704 010136      MOV      ;DHMLSR,R4 ;READ LINE STATUS
5308      020044 117703 010130      MOV      ;DHMCSR,R3 ;READ LINE NUMBER
5309      020050 042703 177760      BIC      #177760,R3 ;CLEAR UNWANTED BITS
5310      020054 020103      CMP      R1,R3      ;IF RECEIVED LINE=SELECTED LINE
5311      020056 001002      BNE      MUX15C ;EXPECT LINE ENABLE AND
5312      020060 012705 000143      MOV      #LINENA+TRMRDY+CC+CS,R5
5313
5314      020064 020405      MUX15C: CMP      R4,R5      ;CLEAR TO SEND AND CARRIER ARE SET
5315      020066 001403      BEQ      MUX15D ;COMPARE EXPECTED AND
5316      020070 104401      TYPE      MSG4      ;RECEIVED RESULTS
5317      020072 035755      MSG4
5318      020074 000000      HALT
5319      020076 052777 000400 010074 MUX15D: BIS      #STEP,;DHMCSR ;UPDATE LINE COUNTER
5320      020104 005302      DEC      R2      ;CONTINUE IF ALL CHECKS
5321      020106 001353      BNE      MUX15B ;ARE NOT DONE FOR THIS LINE
5322      020110 012705 000001      MOV      #LINENA,R5 ;EXPECT LINE ENABLE
5323      020114 010103      MUX15E: MOV     R1,R3      ;ON SELECTED LINE
5324      020116 010177 010056      MOV      R1,;DHMCSR ;SELECT LINE
5325      020122 042777 000002 010052 BIC      #TRMRDY,;DHMLSR ;CLEAR TERMINAL
5326      020130 105227 000000      INCB     #0      ;DELAY FOR CABLE
5327      020134 001375      BNE      #-4      ;DITTO
5328      020136 017704 010040      MOV      ;DHMLSR,R4 ;READ LINE STATUS REGISTER
5329      020142 020504      CMP      R5,R4 ;ONLY LINE ENABLE SHOULD BE
5330      020144 001707      BEQ      DOIT15 ;SET ON THIS LINE
5331      020146 104401      TYPE      MSG4
5332      020150 035755      MSG4
5333      020152 000000      HALT

```

MAINDEC-11-DZDM-B
DZDMB.P11 T56

MACY11 27(732) 23-SEP-76 14:33 PAGE 125
DM11-98 DIAGNOSTIC CONTINUED

K03

SEC 0243

5334
5335

```

5336      ;*****
5337      ;*TEST 57      DM11-BB DIAGNOSTIC CONTINUED
5338      ;*****
5339      020154 000004  T57: SCOPE
5340      REM %
5341      TEST ABSTRACT:
5342      *****
5343
5344      THIS TEST VERIFIES THAT RING IS SET IF "LINE ENABLE" AND
5345      REQUEST TO SEND ARE SET FOR THE SELECTED LINE.
5346
5347      ERRORS:
5348      *****
5349
5350      IF ANY ERRORS OCCUR, RUN THE DM11-BB MODEM CONTROL DIAGNOSTIC,
5351      DZDHK.
5352
5353      %
5354      020156 005077 010020  DOIT16: CLR      2DHMLSR      ;CLEAR LINE STATUS REGISTER
5355      020162 004767 004314      JSR      PC,SELINE  ;GO SELECT A LINE.
5356      020166 000467      BR      TST60      ;;BR IF DONE ALL SELECTED LINES
5357      020170 005077 010004  MUX16: CLR      2DHMCSR      ;CLEAR CONTROL REGISTER
5358      020174 042767 000340      BIC      #340,PS      ;ENABLE INTERRUPTS
5359      020202 116701 010004      MOV      LINE,R1
5360      020206 012702 000020  MUX16A: MOV      #16,R2      ;16 LINES
5361      020212 010177 007762      MOV      R1,2DHMCSR    ;SELECT A LINE
5362      020216 012777 000005 007756  MOV      #LINENA+RS,2DHMLSR ;SET LINE ENABLE +RS
5363      020224 005077 007750      CLR      2DHMCSR      ;CLEAR CONTROL REGISTER
5364      020230 005005      MUX16B: CLR      R5          ;CLEAR EXPECTED RESULT
5365      020232 017704 007744      MOV      2DHMLSR,R4    ;READ LINE STATUS
5366      020236 117703 007736      MOV      2DHMCSR,R3    ;READ LINE NUMBER
5367      020242 042703 177760      BIC      #177760,R3    ;CLEAR UNWANTED BITS
5368      020246 020103      CMP      R1,R3              ;IF RECEIVED LINE=SELECTED LINE
5369      020250 001002      BNE      MUX16C              ;EXPECT LINE ENABLE AND
5370      020252 012705 000205      MOV      #LINENA+RS+RING,R5
5371
5372      020256 020405      MUX16C: CMP      R4,R5          ;RING IS SET
5373      020260 001403      BEQ      MUX16D              ;COMPARE EXPECTED AND
5374      020262 104401      TYPE      MSG4              ;RECEIVED RESULTS
5375      020264 035755      MSG4
5376      020266 000000      HALT
5377      020270 052777 000400 007702  MUX16D: BIS      #STEP,2DHMCSR ;UPDATE LINE COUNTER
5378      020276 005302      DEC      R2                ;CONTINUE IF ALL CHECKS
5379      020300 001353      BNE      MUX16B              ;ARE NOT DONE FOR THIS LINE
5380      020302 012705 000001      MOV      #LINENA,R5    ;EXPECT LINE ENABLE
5381      020306 010103      MUX16E: MOV      R1,R3        ;ON SELECTED LINE
5382      020310 010177 007664      MOV      R1,2DHMCSR    ;SELECT LINE
5383      020314 042777 000004 007660  BIC      #RS,2DHMLSR ;CLEAR REQUEST TO SEND
5384      020322 105227 000000      INCB      #0          ;DELAY FOR CABLE
5385      020326 001375      BNE      -4                ;DITTO
5386      020330 017704 007646      MOV      2DHMLSR,R4    ;READ LINE STATUS REGISTER
5387      020334 020504      CMP      R5,R4              ;ONLY LINE ENABLE SHOULD BE
5388      020336 001707      BEQ      DOIT16              ;SET ON THIS LINE
5389      020340 104401      TYPE      MSG4
5390      020342 035755      MSG4
5391      020344 000000      HALT

```

MAINDEC-11-DZDMM-B
DZDHMB.P11 T57

MACY11 27(732) 23-SEP-76 14:33 PAGE 127
DM11-BB DIAGNOSTIC CONTINUED

M03

SEQ 0245

5392
5393

5394
5395
5396
5397
5398
5399
5400
5401
5402
5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428
5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449

020346 000004

```

*****
:TEST 60      DM11-BB DIAGNOSTIC CONTINUED
*****
+ST60: SCOPE
.REM %
TEST ABSTRACT:
*****

```

THIS TEST VERIFIES THAT SECONDARY RECEIVE IS SET IF "LINE
ENABLE" AND SECONDARY TRANSMIT ARE SET FOR THE SELECTED LINE.

ERRORS:

IF ANY ERRORS OCCUR, RUN THE DM11-BB MODEM CONTROL DIAGNOSTIC,
DZDHK.

```

%
DOIT17: CLR      @DHMLSR      ;CLEAR LINE STATUS REGISTER.
        JSR      PC, SELINE    ;GO SELECT A LINE.
        BR      ENDA
MUX17:  CLR      @DHMCSR      ;CLEAR CONTROL REGISTER
        BIC      #340, PS      ;ENABLE INTERRUPTS
        MOVB     LINE, R1
MUX17A: MOV      #16, R2      ;16 LINES
        MOV      R1, @DHMCSR   ;SELECT A LINE
        MOV      #LINENA+SECTX, @DHMLSR ;SET LINE ENABLE +SECTX
        CLR      @DHMCSR      ;CLEAR CONTROL REGISTER
MUX17B: CLR      R5          ;CLEAR EXPECTED RESULT
        MOV      @DHMLSR, R4   ;READ LINE STATUS
        MOVB     @DHMCSR, R3   ;READ LINE NUMBER
        BIC      #177760, R3  ;CLEAR UNWANTED BITS
        CMP      R1, R3       ;IF RECEIVED LINE=SELECTED LINE
        BNE      MUX17C      ;EXPECT LINE ENABLE AND
        MOV      #LINENA+SECTX+SECRX, R5 ;SECONDARY RECEIVE IS SET
MUX17C: CMP      R4, R5      ;COMPARE EXPECTED AND
        BEQ      MUX17D      ;RECEIVED RESULTS
        TYPE     MSG4
        MSG4     HALT
MUX17D: BIS      #STEP, @DHMCSR ;UPDATE LINE COUNTER
        DEC      R2          ;CONTINUE IF ALL CHECKS
        BNE      MUX17B      ;ARE NOT DONE FOR THIS LINE
        MOV      #LINENA, R5  ;EXPECT LINE ENABLE
        MOV      R1, R3       ;ON SELECTED LINE
MUX17E: MOV      R1, @DHMCSR   ;SELECT LINE
        BIC      #SECTX, @DHMLSR ;CLEAR SECONDARY TRANSMIT
        INCB     #0          ;DELAY FOR CABLE
        BNE     -4          ;DITTO
        MOV      @DHMLSR, R4  ;READ LINE STATUS REGISTER
        CMP      R5, R4       ;ONLY LINE ENABLE SHOULD BE
        BEQ      DOIT17      ;SET ON THIS LINE
        TYPE     MSG4
        MSG4     HALT

```

.S9YL END OF PASS ROUTINE

SEOP:

```

.SBTTL SCOPE HANDLER ROUTINE

```

[illegible]

C04

23-SEP-76 14:33 PAGE 130

SEG 0248

```

*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
*AND LOAD THE TEST NUMBER (STSTNM) INTO THE DISPLAY REG. (DISPLAY 7:0)
*AND LOAD THE ERROR FLAG (SERFLG) 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=10T

```

```

SSCOPE:
CKSWR
CLR LINE
MOV DMAADR,R1
BIT #BIT14,JSWR
BNE $OVER
*****START OF CODE FOR THE XOR TESTER*****
STSTR: BR 65
MOV 2*ERRVEC, -(SP)
MOV 2*ERRVEC
TST 2*177060
MOV (SP)+, 2*ERRVEC
BR $SVLAD
SS: CMP (SP)+, (SP)+
MOV (SP)+, 2*ERRVEC
BR 75
65: *****END OF CODE FOR THE XOR TESTER*****
BIT #BIT08,JSWR
BEQ 25
CMPB JSWR,STSTNM
BEQ $OVER
25: TSTB SERFLG
BEQ 35
CMPB SERMAX,SERFLG
BH 35
BIT #BIT09,JSWR
BEQ 45
75: MOV $LPERR,$LPADR
BR $OVER
45: CLRB SERFLG
CLR $TIMES
BR 15
35: BIT #BIT11,JSWR
BNE 15
TST $PASS
BEQ 15
INC $ICNT
CMP $TIMES,$ICNT
BGE $OVER
15: MOV #1,$ICNT
MOV $MXCNT,$TIMES
SSVLAD: INCB STSTNM
MOV STSTNM,STSTNM
MOV (SP), $LPADR

```

```

::TEST FOR CHANGE IN SOFT-SWR
::INIT THE LINE NO. TO ZERO
::SET UP DEVAADR IN R1
::LOOP ON PRESENT TEST?
::YES IF SW14=1
::IF RUNNING ON THE "YOR" TESTER CHANGE
::THIS INSTRUCTION TO A "NOP" (NOP=240)
::SAVE THE CONTENTS OF THE ERROR VECTOR
::SET FOR TIMEOUT
::TIME OUT ON XOR?
::RESTORE THE ERROR VECTOR
::GO TO THE NEXT TEST
::CLEAR THE STACK AFTER A TIME OUT
::RESTORE THE ERROR VECTOR
::LOOP ON THE PRESENT TEST
::TESTER*****
::LOOP ON SPEC. TEST?
::BR IF NO
::ON THE RIGHT TEST? SWR 7:0
::BR IF YES
::HAS AN ERROR OCCURRED?
::BR IF NO
::MAX. ERRORS FOR THIS TEST OCCURRED?
::BR IF NO
::LOOP ON ERROR?
::BR IF NO
::SET LOOP ADDRESS TO LAST SCOPE
::ZERO THE ERROR FLAG
::CLEAR THE NUMBER OF ITERATIONS TO MAKE
::ESCAPE TO THE NEXT TEST
::INHIBIT ITERATIONS?
::BR IF YES
::IF FIRST PASS OF PROGRAM
::INHIBIT ITERATIONS
::INCREMENT ITERATION COUNT
::CHECK THE NUMBER OF ITERATIONS MADE
::BR IF MORE ITERATION REQUIRED
::REINITIALIZE THE ITERATION COUNTER
::SET NUMBER OF ITERATIONS TO DO
::COUNT TEST NUMBERS
::SET TEST NUMBER IN APT MAILBOX
::SAVE SCOPE LOOP ADDRESS

```

```

001202 011667 157702      MOV      (SP), $LPERR      ;; SAVE ERROR LOOP ADDRESS
001203 005067 160012      CLR      $ESCAPE      ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
001204 112767 000001 157675  $OVER:  MOV      #1, $SERMAX      ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
001205 016777 157656 157714  MOV      $STNM, $DISPLAY  ;; DISPLAY TEST NUMBER
001206 016776 157654      MOV      $LPADR, (SP)      ;; FUDGE RETURN ADDRESS
001207 000002      RTI      ;; FIXES PS
001208 000010  SMXCN: 10      ;; MAX. NUMBER OF ITERATIONS
001209 000010  .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

$ERRCR:
001236 104407 157637 7$:  CKSWR      ;; TEST FOR CHANGE IN SOFT-SWR
001237 105267 157637  INCB      $ERFLG      ;; SET THE ERROR FLAG
001238 001777 157637  BEQ      7$      ;; DON'T LET THE FLAG GO TO ZERO
001239 016777 157637  MOV      $STNM, $DISPLAY  ;; DISPLAY TEST NUMBER AND ERROR FLAG
001240 005267 157637  INC      $ERTTL      ;; INC THE ERROR COUNT
001241 016667 157637  MOV      (SP), $ERRPC      ;; GET ADDRESS OF ERROR INSTRUCTION
001242 162767 000002 157624  SUB      #2, $ERRPC
001243 117767 157620 157614  MOV      $ERRPC, $ITEMB      ;; STRIP AND SAVE THE ERROR ITEM CODE
001244 032777 020000 157632  BIT      #BIT13, $SWR      ;; SKIP TYPEOUT IF SET
001245 001004 20$      BNE      20$      ;; SKIP TYPEOUTS
001246 004767 000106  JSR      PC, $ERRTYP      ;; GO TO USER ERROR ROUTINE
001247 104401 001227  TYPE      $SRLF

20$:  CMPB      #APTENV, $ENV      ;; RUNNING IN APT MODE
001248 001007 2$      BNE      2$      ;; NO SKIP APT ERROR REPORT
001249 116767 157560 000004  MOV      $ITEMB, 21$      ;; SET ITEM NUMBER AS ERROR NUMBER
001250 004767 001166  JSR      PC, $ATY4      ;; REPORT FATAL ERROR TO APT
001251 000      21$:  .BYTE      0
001252 000      .BYTE      0
001253 000777 22$:  BR      22$      ;; APT ERROR LOOP
001254 005777 157566 2$:  TST      $SWR      ;; HALT ON ERROR
001255 100002  BPL      3$      ;; SKIP IF CONTINUE
001256 000000  HALT      ;; HALT ON ERROR!
001257 104407  CKSWR      ;; TEST FOR CHANGE IN SOFT-SWR
001258 032777 001000 157552 3$:  BIT      #BIT09, $SWR      ;; LOOP ON ERROR SWITCH SET?
001259 001402 4$      BEQ      4$      ;; BR IF NO
001260 016716 157514  MOV      $LPERR, (SP)      ;; FUDGE RETURN FOR LOOPING
001261 005767 157624 4$:  TST      $ESCAPE      ;; CHECK FOR AN ESCAPE ADDRESS
001262 001402 5$      BEQ      5$      ;; BR IF NONE
001263 016716 157616  MOV      $ESCAPE, (SP)      ;; FUDGE RETURN ADDRESS FOR ESCAPE
001264 022737 020712 000042 5$:  CMP      #SENDAC, 2#42      ;; ACT-11 AUTO-ACCEPT?
001265 001001 6$      BNE      6$      ;; BRANCH IF NO
001266 000000  HALT      ;; YES
6$:

```

E04

MAINDEC-11-02DMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 132
 02DMMB.P11 ERROR HANDLER ROUTINE

SEG 0250

```

5618 021420 000002          RFI          ;:RETURN
5619          .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
5620
5621          ;:*****
5622          ;:THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
5623          ;:ERROR IS TO BE REPORTED. IT THEN OBTAINS FROM THE "ERROR TABLE" $ERRTB.
5624          ;:AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
5625
5626          $ERRTYP:
5627          TYPE      $CRLF          ;:"CARRIAGE RETURN" & "LINE FEED"
5628          MOV      RD,-(SP)        ;:SAVE RD
5629          CLR      RD              ;:PICKUP THE ITEM INDEX
5630          SISB     2,$ITEMB,RD
5631          BNE      IS              ;:IF ITEM NUMBER IS ZERO, JUST
5632          ;:TYPE THE PC OF THE ERROR
5633          MOV      $ERRPC,-(SP)    ;:SAVE $ERRPC FOR TYPEOUT
5634          ;:ERROR ADDRESS
5635          TYP0C     ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
5636          BR       6$              ;:GET OUT
5637          ;:ADJUST THE INDEX SO THAT IT WILL
5638          ;:WORK FOR THE ERROR TABLE
5639
5640          1$:
5641          DEC      RD
5642          ASL      RD
5643          ASL      RD
5644          ASL      RD
5645          ADD      $ERRTB,RD        ;:FORM TABLE POINTER
5646          MOV      (RD)+,2$        ;:PICKUP "ERROR MESSAGE" POINTER
5647          BEQ      3$              ;:SKIP TYPEOUT IF NO POINTER
5648          TYPE     0               ;:TYPE THE "ERROR MESSAGE"
5649          ;:"ERROR MESSAGE" POINTER GOES HERE
5650          ;:"CARRIAGE RETURN" & "LINE FEED"
5651          2$:
5652          TYPE     $CRLF
5653          MOV      (RD)+,4$        ;:PICKUP "DATA HEADER" POINTER
5654          BEQ      5$              ;:SKIP TYPEOUT IF 0
5655          TYPE     0               ;:TYPE THE "DATA HEADER"
5656          ;:"DATA HEADER" POINTER GOES HERE
5657          ;:"CARRIAGE RETURN" & "LINE FEED"
5658          3$:
5659          MOV      (RD),RD          ;:PICKUP "DATA TABLE" POINTER
5660          BNE      7$              ;:GO TYPE THE DATA
5661          MOV      (SP)+,RD        ;:RESTORE RD
5662          TYPE     $CRLF          ;:"CARRIAGE RETURN" & "LINE FEED"
5663          RTS      PC              ;:RETURN
5664
5665          4$:
5666          TYPE     0               ;:"DATA HEADER" POINTER GOES HERE
5667          ;:"CARRIAGE RETURN" & "LINE FEED"
5668          5$:
5669          MOV      (RD),RD          ;:PICKUP "DATA TABLE" POINTER
5670          BNE      7$              ;:GO TYPE THE DATA
5671          MOV      (SP)+,RD        ;:RESTORE RD
5672          TYPE     $CRLF          ;:"CARRIAGE RETURN" & "LINE FEED"
5673          RTS      PC              ;:RETURN
5674
5675          6$:
5676          MOV      2(RD)+,-(SP)    ;:SAVE 2(RD)+ FOR TYPEOUT
5677          TYP0C     ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
5678          TST      (RD)            ;:IS THERE ANOTHER NUMBER?
5679          BEQ      6$              ;:BR IF NO
5680          TYPE     2$              ;:TYPE TWO(2) SPACES
5681          BR       7$              ;:LOOP
5682
5683          7$:
5684          MOV      2(RD)+,-(SP)    ;:SAVE 2(RD)+ FOR TYPEOUT
5685          TYP0C     ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
5686          TST      (RD)            ;:IS THERE ANOTHER NUMBER?
5687          BEQ      6$              ;:BR IF NO
5688          TYPE     2$              ;:TYPE TWO(2) SPACES
5689          BR       7$              ;:LOOP
5690
5691          8$:
5692          .ASCIZ   / /              ;:TWO(2) SPACES
5693          .EVEN
5694          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
5695
5696          ;:*****
5697          ;:THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
5698          ;:OCTAL (ASCII) NUMBER AND TYPE IT.
5699          ;:$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
5700          ;:CALL:
5701          ;:MOV      NUM,-(SP)      ;:NUMBER TO BE TYPED

```

F04

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 133
 DZDMMB.P11 BINARY TO OCTAL (ASCII) AND TYPE

SEQ C251

```

*      TYPEOS      ::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
*STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*STYPOS OR STYPOC
*CALL:
*      MOV      NUM,-(SP)      ::NUMBER TO BE TYPED
*      TYPON      ::CALL FOR TYPEOUT
*
*STYPOC----ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*      MOV      NUM,-(SP)      ::NUMBER TO BE TYPED
*      TYPOC      ::CALL FOR TYPEOUT
*
021556 017646 000000 000211 STYPOS: MOV      2(SP),-(SP)      ::PICKUP THE MODE
021557 116667 000001      MOV8      1(SP),SOFILL      ::LOAD ZERO FILL SWITCH
021558 112667 000207      MOV8      (SP)+,SOMODE+1      ::NUMBER OF DIGITS TO TYPE
021559 062716 000002      ADD      #2,(SP)      ::ADJUST RETURN ADDRESS
021560 000406      BR      STYPON
021561 112767 000001 000171 STYPOC: MOV8      #1,SOFILL      ::SET THE ZERO FILL SWITCH
021562 112767 000006 000165      MOV8      #6,SOMODE+1      ::SET FOR SIX(6) DIGITS
021563 112767 000005 000154 STYPON: MOV8      #5,SOCNT      ::SET THE ITERATION COUNT
021564 010346      MOV      R3,-(SP)      ::SAVE R3
021565 010446      MOV      R4,-(SP)      ::SAVE R4
021566 010546      MOV      R5,-(SP)      ::SAVE R5
021567 116704 000145      MOV8      SOMODE+1,R4      ::GET THE NUMBER OF DIGITS TO TYPE
021568 005404      NEG      R4
021569 062704 000006      ADD      #6,R4      ::SUBTRACT IT FOR MAX. ALLOWED
021570 110467 000132      MOV8      R4,SOMODE      ::SAVE IT FOR USE
021571 116704 000125      MOV8      SOFILL,R4      ::GET THE ZERO FILL SWITCH
021572 016605 000012      MOV      12(SP),R5      ::PICKUP THE INPUT NUMBER
021573 005003      CLR      R3      ::CLEAR THE OUTPUT WORD
021574 006105      15: ROL      R5      ::ROTATE MSB INTO "C"
021575 000404      BR      35      ::GO DO MSB
021576 006105      25: ROL      R5      ::FORM THIS DIGIT
021577 006105      ROL      R5
021578 010503      MOV      R5,R3
021579 006103      35: ROL      R3      ::GET LSB OF THIS DIGIT
021580 105367 000076      DECB      SOMODE      ::TYPE THIS DIGIT?
021581 100016      BPL      75      ::BR IF NO
021582 042703 177770      BIC      #177770,R3      ::GET RID OF JUNK
021583 001002      BNE      45      ::TEST FOR 0
021584 005704      TST      R4      ::SUPPRESS THIS 0?
021585 001403      BEQ      55      ::BR IF YES
021586 005204      45: INC      R4      ::DON'T SUPPRESS ANYMORE 0'S
021587 052703 000060      BIS      #'0,R3      ::MAKE THIS DIGIT ASCII
021588 052703 000040      55: BIS      #' ,R3      ::MAKE ASCII IF NOT ALREADY
021589 110367 000040      MOV8      R3,R5      ::SAVE FOR TYPING
021590 104401 021776      TYPE      R5      ::GO TYPE THIS DIGIT
021591 105367 000032      75: DECB      SOCNT      ::COUNT BY 1
021592 003347      BGT      25      ::BR IF MORE TO DO
021593 002402      BLT      65      ::BR IF DONE

```

G04

 MAINDEC-11-DZDMM-8 MACV11 27 732) 23-SEP-76 14:33 PAGE 134
 DZDMM8.P11 BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0252

```

5730 021752 005204 INC R4 ::INSURE LAST DIGIT ISN'T A BLANK
5731 021754 000744 BR 2$ ::GO DO THE LAST DIGIT
5732 021756 012605 6$: MOV (SP)+,R5 ::RESTORE R5
5733 021760 012604 MOV (SP)+,R4 ::RESTORE R4
5734 021762 012603 MOV (SP)+,R3 ::RESTORE R3
5735 021764 016666 000002 000004 MOV 2(SP),4(SP) ::SET THE STACK FOR RETURNING
5736 021772 012616 MOV (SP)+,(SP)
5737 021774 000002 RTI ::RETURN
5738 021776 000 .BYTE 0 ::STORAGE FOR ASCII DIGIT
5739 021777 000 .BYTE 0 ::TERMINATOR FOR TYPE ROUTINE
5740 022000 000 $OCNT: .BYTE 0 ::OCTAL DIGIT COUNTER
5741 022001 000 $OFILL: .BYTE 0 ::ZERO FILL SWITCH
5742 022002 000000 $OMODE: .WORD 0 ::NUMBER OF DIGITS TO TYPE
5743 .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

$TYPDS: MOV R0,-(SP) ::PUSH R0 ON STACK
MOV R1,-(SP) ::PUSH R1 ON STACK
MOV R2,-(SP) ::PUSH R2 ON STACK
MOV R3,-(SP) ::PUSH R3 ON STACK
MOV R5,-(SP) ::PUSH R5 ON STACK
MOV #20200,-(SP) ::SET BLANK SWITCH AND SIGN
MOV 20(SP),R5 ::GET THE INPUT NUMBER
BPL 1$ ::BR IF INPUT IS POS.
NEG R5 ::MAKE THE BINARY NUMBER POS.
MOV #'-,1(SP) ::MAKE THE ASCII NUMBER NEG.
1$: CLR R0 ::ZERO THE CONSTANTS INDEX
MOV $DBLK,R3 ::SETUP THE OUTPUT POINTER
MOV #'. ,R3)+ ::SET THE FIRST CHARACTER TO A BLANK
2$: CLR R2 ::CLEAR THE BCD NUMBER
MOV $DTBL(R0),R1 ::GET THE CONSTANT
3$: SUB R1,R5 ::FORM THIS BCD DIGIT
BLT 4$ ::BR IF DONE
INC R2 ::INCREASE THE BCD DIGIT BY 1
4$: ADD R1,R5 ::ADD BACK THE CONSTANT
TST R2 ::CHECK IF BCD DIGIT=0
BNE 5$ ::FALL THROUGH IF 0
TSTB (SP) ::STILL DOING LEADING 0'S?
BMI 7$ ::BR IF YES
5$: ASLB (SP) ::MSD?
BCC 6$ ::BR IF NO
MOV 1(SP),-1(R3) ::YES--SET THE SIGN
6$: BIS #0,R2 ::MAKE THE BCD DIGIT ASCII
7$: BIS #'. ,R2 ::MAKE IT A SPACE IF NOT ALREADY A DIGIT
MOVB R2,(R3)+ ::PUT THIS CHARACTER IN THE OUTPUT BUFFER

```

H04

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 135
 DZDMMB.P11 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

SEQ 0253

5796	022126	005720		TST	R0,4	:: JUST INCREMENTING
5797	022130	020027	000010	CMP	R0, #10	:: CHECK THE TABLE INDEX
5798	022134	002746		BLT	25	:: GO DO THE NEXT DIGIT
5799	022136	003002		BGT	85	:: GO TO EXIT
5800	022140	010502		MOV	R3,R2	:: GET THE LSD
5801	022142	000764		BR	85	:: GO CHANGE TO ASCII
5802	022144	105726	85:	TSTB	(SP)+	:: WAS THE LSD THE FIRST NON-ZERO?
5803	022146	100003		BPL	95	:: BR IF NO
5804	022150	116663	177777 177776	MOV	-1(SP),-2 R3)	:: YES--SET THE SIGN FOR TYPING
5805	022156	105013	95:	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						
5824						
5825						
5826						
5827	022230	105767	156723	\$TYPE:	TSTB	\$TFPLG
5828	022234	100002			BPL	15
5829	022236	000000			HALT	
5830	022240	000430			BR	35
5831	022242	010046		15:	MOV	R0,-(SP)
5832	022244	017600	000002		MOV	32(SP),R0
5833	022250	122767	000001 156774		CMPB	\$APTENV,\$ENV
5834	022256	001011			BNE	625
5835	022260	132767	000100 156765		BITB	\$APTSPool,\$ENV
5836	022266	001405			BEQ	625
5837	022270	010067	000004		MOV	R0,615
5838	022274	004767	000220		JSR	PC,\$ATY3
5839	022300	000000		615:	.WORD	0
5840	022302	132767	000040 156743	625:	BITB	\$APTCSJP,\$ENV
5841	022310	001003			BNE	605

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

 *ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 *NOTE1: \$NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 *NOTE2: \$FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 *NOTE3: \$FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 *

*CALL:
 *1) USING A TRAP INSTRUCTION
 * TYPE .MESADR ; MESADR IS FIRST ADDRESS OF AN ASCII STRING
 *OR
 * TYPE
 * MESADR
 *

:: IS THERE A TERMINAL?
 :: BR IF YES
 :: HALT HERE IF NO TERMINAL
 :: LEAVE
 :: SAVE R0
 :: GET ADDRESS OF ASCII STRING
 :: RUNNING IN APT MODE
 :: NO, GO CHECK FOR APT CONSOLE
 :: SPOOL MESSAGE TO APT
 :: NO, GO CHECK FOR CONSOLE
 :: SETUP MESSAGE ADDRESS FOR APT
 :: SPOOL MESSAGE TO APT
 :: MESSAGE ADDRESS
 :: APT CONSOLE SUPPRESSED
 :: YES, SKIP TYPE OUT


```

5842 022312 112046      2$:  MOVB      (RO)+, -(SP)      ;; PUSH CHARACTER TO BE TYPED ONTC STACK
5843 022314 001005      BNE      4$              ;; BR IF IT ISN'T THE TERMINATOR
5844 022316 005726      TST      (SP)+          ;; IF TERMINATOR POP IT OFF THE STACK
5845 022320 012600      60$:  MOV      (SP)+, RO      ;; RESTORE RO
5846 022322 062716 000002 3$:  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, $TYPEPC      ;; GO TYPE THIS CHARACTER
5858 022364 126726 156566 5$:  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, $TYPEPC      ;; 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, $TYPEPC      ;; 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 $TYPEPC: TSTB      $STPB      ;; WAIT UNTIL PRINTER IS READY
5877 022446 100375      BPL      $TYPEPC
5878 022450 116677 000002 156474      MOVB      2(SP), $STPB      ;; LOAD CHAR TO BE TYPED INTO DATA REG.
5879 022456 122766 000015 000002      CMPB      #CR, 2(SP)      ;; IS CHARACTER A CARRIAGE RETURN?
5880 022464 001003      BNE      1$              ;; BRANCH IF NO
5881 022466 105067 000014      CLRB      $CHARCNT      ;; YES--CLEAR CHARACTER COUNT
5882 022472 000406      BR      $TYPEPC      ;; EXIT
5883 022474 122766 000012 000002 1$:  CMPB      #LF, 2(SP)      ;; IS CHARACTER A LINE FEED?
5884 022502 001402      BEQ      $TYPEPC      ;; BRANCH IF YES
5885 022504 105227      INCB      (PC)+          ;; COUNT THE CHARACTER
5886 022506 000000      $CHARCNT: .WORD      0      ;; CHARACTER COUNT STORAGE
5887 022510 000207      $TYPEPC: 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      RO, -(SP)      ;; PUSH RO ON STACK

```

J04

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 137
 DZDMMB.P11 APT COMMUNICATIONS ROUTINE

SEQ 0255

```

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     $APTPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
5904 022566 001425      SEQ      3$          ;;IF NOT: BR
5905 022570 017600 000004      MOV      24(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      $MSGAD,R0      ;;SUB START OF MESSAGE
5913 022624 006200      RO          ;;GET MESSAGE LGTH IN WORDS
5914 022626 010067 156416      MOV      R0,$MSGGLT      ;;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$:  MOV      24(SP),4$      ;;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$:  10$:  TSTB     $FFLG      ;;SHOULD REPORT FATAL ERROR?
5923 022670 105767 000062      BEQ      12$      ;;IF NOT: BR
5924 022674 001416      TST      $ENV          ;;RUNNING UNDER APT?
5925 022676 005767 156350      BEQ      12$      ;;IF NOT: BR
5926 022702 001413      TST      $MSGTYPE      ;;FINISHED LAST MESSAGE?
5927 022704 005767 156322 11$:  BNE      11$      ;;IF NOT: WAIT
5928 022710 001375      MOV      24(SP),$FATAL      ;;GET ERROR #
5929 022712 017667 000004 156314  ADD      #2,4(SP)      ;;BUMP RETURN ADDR.
5930 022720 062766 000002 000004  INC      $MSGTYPE      ;;TELL APT TO TAKE ERROR
5931 022726 005267 156300 12$:  CLRB     $FFLG      ;;CLEAR FATAL FLAG
5932 022732 105067 000020      CLRB     $LFLG      ;;CLEAR LOG FLAG
5933 022736 105067 000013      CLRB     $MFLG      ;;CLEAR MESSAGE FLAG
5934 022742 105067 000006      MOV      (SP)+,R1      ;;POP STACK INTO R1
5935 022746 012601      MOV      (SP)+,R0      ;;POP STACK INTO R0
5936 022750 012600      RTS      PC          ;;RETURN
5937 022752 000207      $MFLG: .BYTE      0      ;;MESSG. FLAG
5938 022754 000      $LFLG: .BYTE      0      ;;LOG FLAG
5939 022755 000      $FFLG: .BYTE      0      ;;FATAL FLAG
5940 022756 000      .EVEN
5941 022760 000200  APTSIZE=200
5942 000001  APTENV=001
5943 000100  APTPOOL=100
5944 000040  APTCSUP=040
5945 000040  .SBTTL  TTY INPUT ROUTINE
5946
5947
5948 *****
5949 .ENABL .LSB
5950
5951 *****
5952 *SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
5953 *ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL

```


K04

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 138
 DZDMMB.P11 TTY INPUT ROUTINE

SEQ 0256

```

5954      : *SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
5955      : *WHEN OPERATING IN TTY FLAG MODE.
5956 022760 022767 000176 156152 $CKSWR: CMP      $SWREG,SWR      ;; IS THE SOFT-SWR SELECTED?
5957 022766 001074      BNE      15$      ;; BRANCH IF NO
5958 022770 105777 156150      TSTB     $STKS      ;; CHAR THERE?
5959 022774 100071      BPL      15$      ;; IF NO, DON'T WAIT AROUND
5960 022776 117746 156144      MOVB     $STKB,-(SP)    ;; SAVE THE CHAR
5961 023002 042716 177600      BIC      $C177,(SP)    ;; STRIP-OFF THE ASCII
5962 023006 022726 000007      CMP      $7,(SP)+      ;; IS IT A CONTROL G?
5963 023012 001062      BNE      15$      ;; NO, RETURN TO USER
5964 023014 126727 156114 000001      CMPEB    $AUTOB.#1    ;; ARE WE RUNNING IN AUTO-MODE?
5965 023022 001456      BEQ      15$      ;; BRANCH IF YES
5966
5967 023024 104401 023633      TYPE     , $CNTLG      ;; ECHO THE CONTROL-G (↑G)
5968 023030 104401 023640      $GTSWR: TYPE     , $MSWR      ;; TYPE CURRENT CONTENTS
5969 023034 016746 155136      MOV      $SWREG,-(SP)    ;; SAVE SWREG FOR TYPEOUT
5970 023040 104402      TYPEOC      ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
5971 023042 104401 023651      TYPE     , $MNEW      ;; PROMPT FOR NEW SWR
5972 023046 005046 19$:      CLR      -(SP)      ;; CLEAR COUNTER
5973 023050 005046      CLR      -(SP)      ;; THE NEW SWR
5974 023052 105777 156066 7$:      TSTB     $STKS      ;; CHAR THERE?
5975 023056 100375      BPL      7$      ;; IF NOT TRY AGAIN
5976
5977 023060 117746 156062      MOVB     $STKB,-(SP)    ;; PICK UP CHAR
5978 023064 042716 177600      BIC      $C177,(SP)    ;; MAKE IT 7-BIT ASCII
5979
5980
5981
5982 023070 021627 000025 9$:      CMP      (SP),#25      ;; IS IT A CONTROL-U?
5983 023074 001005      BNE      10$      ;; BRANCH IF NOT
5984 023076 104401 023626      TYPE     , $CNTLU      ;; YES, ECHO CONTROL-U (↑U)
5985 023102 062706 000006 20$:      ADD      #6,SP      ;; IGNORE PREVIOUS INPUT
5986 023106 000757      BR      19$      ;; LET'S TRY IT AGAIN
5987
5988
5989 023110 021627 000015 10$:      CMP      (SP),#15      ;; IS IT A <CR>?
5990 023114 001022      BNE      16$      ;; BRANCH IF NO
5991 023116 005766 000004      TST      4(SP)      ;; YES, IS IT THE FIRST CHAR?
5992 023122 001403      BEQ      11$      ;; BRANCH IF YES
5993 023124 016677 000002 156006      MOV      2(SP), $SWR      ;; SAVE NEW SWR
5994 023132 062706 000006 11$:      ADD      #6,SP      ;; CLEAR UP STACK
5995 023136 104401 001227 14$:      TYPE     , $CRLF      ;; ECHO <CR> AND <LF>
5996 023142 126727 155767 000001      CMPEB    $INTAG.#1    ;; RE-ENABLE TTY KBD INTERRUPTS?
5997 023150 001003      BNE      15$      ;; BRANCH IF NOT
5998 023152 012777 000100 155764      MOV      #100,$STKS    ;; RE-ENABLE TTY KBD INTERRUPTS
5999 023160 000002 15$:      RTI      ;; RETURN
6000 023162 004767 177254 16$:      JSR      PC,$TYPEC      ;; ECHO CHAR
6001 023166 021627 000060      CMP      (SP),#60      ;; CHAR < 0?
6002 023172 002420      BLT      18$      ;; BRANCH IF YES
6003 023174 021627 000067      CMP      (SP),#67      ;; CHAR > 7?
6004 023200 003015      BGT      18$      ;; BRANCH IF YES
6005 023202 042726 000060      BIC      #60,(SP)+      ;; STRIP-OFF ASCII
6006 023206 005766 000002      TST      2(SP)      ;; IS THIS THE FIRST CHAR
6007 023212 001403      BEQ      17$      ;; BRANCH IF YES
6008 023214 006316      ASL      (SP)      ;; NO, SHIFT PRESENT
6009 023216 006316      ASL      (SP)      ;; CHAR OVER TO MAKE

```

L04

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 139
 DZDMMB.P11 TTY INPUT ROUTINE

SEQ 0257

```

6010 023220 006316          ASL      (SP)          ;; ROOM FOR NEW ONE.
6011 023222 005266 000002 17$: INC      2(SP)      ;; KEEP COUNT OF CHAR
6012 023226 056616 177776  BIS      -2(SP), (SP)  ;; SET IN NEW CHAP
6013 023232 000707          BR       7$          ;; 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
6021
6022
6023
6024
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     2$TKS      ;; WAIT FOR
6030 023256 100375          BPL       1$          ;; A CHARACTER
6031 023260 117766 155662 000004 MOVB     2$TKB, 4(SP)  ;; READ THE TTY
6032 023266 042766 177600 000004 BIC      #1C177, 4(SP)  ;; GET RID OF JUNK IF ANY
6033 023274 026627 000004 000023 CMP      4(SP), #23  ;; IS IT A CONTROL-S?
6034 023302 001013          BNE       3$          ;; BRANCH IF NO
6035 023304 105777 155634 2$: TSTB     2$TKS      ;; WAIT FOR A CHARACTER
6036 023310 100375          BPL       2$          ;; LOOP UNTIL ITS THERE
6037 023312 117746 155630 MOVB     2$TKB, -(SP)  ;; GET CHARACTER
6038 023316 042716 177600 BIC      #1C177, (SP)  ;; MAKE IT 7-BIT ASCII
6039 023322 022627 000021 CMP      (SP)+, #21  ;; IS IT A CONTROL-Q?
6040 023326 001366          BNE       2$          ;; 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
6050
6051
6052
6053
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

```

M04

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 140
 DZDMMB.P11 TTY INPUT ROUTINE

SEQ 0258

```

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              DEC     R3                ;;BACKUP BY ONE
6070 023436 020327 023616 6$:          CMP     R3,$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$
6074 023454 000746              BR       2$                ;;GO TYPE
6075 023456 005716 5$:          TST     (SP)                ;;GO READ ANOTHER CHAR.
6076 023460 001406              BEQ     7$                ;;RUBOUT KEY SET?
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    ,SCNTLU      ;;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$:          TYPE    ,QUES        ;;TYPE A "?"
6091 023540 000712 3$:          BR       1$                ;;CLEAR THE BUFFER AND LOOP
6092 023542 111367 000046              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    ,SLF        ;;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      MC      ,#$TTYIN,4(SP)
6103 023612 000002              RTI                ;;RETURN
6104 023614 000 9$:          .BYTE    0      ;;STORAGE FOR ASCII CHAR. TO TYPE
6105 023616 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 .SBTTL READ AN OCTAL NUMBER FROM THE TTY
6114
6115 ;;*****
6116 ;;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
6117 ;;*CHANGE IT TO BINARY.
6118 ;;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
6119 ;;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
6120 ;;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
6121 ;;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.

```

N04

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 141
 DZDMMB.P11 READ AN OCTAL NUMBER FROM THE TTY

SEQ 0259

```

6122      ;*CALL:
6123      ;*      RDOCT
6124      ;*      RETURN HERE
6125      ;*
6126      ;*
6127      023662 011646      SRDOCT: MOV      (SP), -(SP)      ;; PROVIDE SPACE FOR THE
6128      023664 016566 000004 000002      MOV      4(SP), 2(SP)      ;; INPUT NUMBER
6129      023672 010046      MOV      R0, -(SP)      ;; PUSH R0 ON STACK
6130      023674 010146      MOV      R1, -(SP)      ;; PUSH R1 ON STACK
6131      023676 010246      MOV      R2, -(SP)      ;; PUSH R2 ON STACK
6132      023700 104411      1$:      RDLIN      ;; READ AN ASCII LINE
6133      023702 012600      MOV      (SP)+, R0      ;; GET ADDRESS OF 1ST CHARACTER
6134      023704 010067 000100      MOV      R0, 5$      ;; AND SAVE IT.
6135      023710 005001      CLR      R1      ;; CLEAR DATA WORD
6136      023712 005002      CLR      R2
6137      023714 112046      2$:      MOVB      (R0)+, -(SP)      ;; PICKUP THIS CHARACTER
6138      023716 001420      BEQ      3$      ;; IF ZERO GET OUT
6139      023720 122716 000060      CMPB      #'0, (SP)      ;; MAKE SURE THIS CHARACTER
6140      023724 003026      BGT      4$      ;; IS AN OCTAL DIGIT
6141      023726 122716 000067      CMPB      #'7, (SP)
6142      023732 002423      BLT      4$
6143      023734 006301      ASL      R1      ;; *2
6144      023736 006102      ROL      R2
6145      023740 006301      ASL      R1      ;; *4
6146      023742 006102      ROL      R2
6147      023744 006301      ASL      R1      ;; *8
6148      023746 006102      ROL      R2
6149      023750 042716 177770      BIC      #'07, (SP)      ;; STRIP THE ASCII JUNK
6150      023754 062601      ADD      (SP)+, R1      ;; ADD IN THIS DIGIT
6151      023756 000756      BR      2$      ;; LOOP
6152      023760 005726      3$:      TST      (SP)+      ;; CLEAN TERMINATOR FROM STACK
6153      023762 010166 000012      MOV      R1, 12(SP)      ;; SAVE THE RESULT
6154      023766 010267 000026      MOV      R2, $HI OCT
6155      023772 012602      MOV      (SP)+, R2      ;; POP STACK INTO R2
6156      023774 012601      MOV      (SP)+, R1      ;; POP STACK INTO R1
6157      023776 012600      MOV      (SP)+, R0      ;; POP STACK INTO R0
6158      024000 000002      RTI      ;; RETURN
6159      024002 005726      4$:      TST      (SP)+      ;; CLEAN PARTIAL FROM STACK
6160      024004 105010      CLRB      (R0)      ;; SET A TERMINATOR
6161      024006 104401      TYPE      ;; TYPE UP THRU THE BAD CHAR.
6162      024010 000000      5$:      .WORD      0
6163      024012 104401 001226      TYPE      $QUES      ;; "?" "CR" & "LF"
6164      024016 000730      BR      1$      ;; TRY AGAIN
6165      024020 000000      $HI OCT: .WORD      0      ;; HIGH ORDER BITS GO HERE
6166      .SBTTL TRAP DECODER
6167
6168      ;*****
6169      ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
6170      ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
6171      ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
6172      ;*GO TO THAT ROUTINE.
6173
6174      024022 010046      $TRAP: MOV      R0, -(SP)      ;; SAVE R0
6175      024024 016600 000002      MOV      2(SP), R0      ;; GET TRAP ADDRESS
6176      024030 005740      TST      -(R0)      ;; BACKUP BY 2
6177      024032 111000      MOVB      (R0), R0      ;; GET RIGHT BYTE OF TRAP

```

```

ASL      R0      :: POSITION FOR INDEXING
MOV      STRPAD(R0),R0  :: INDEX TO TABLE
RTS      R0      :: GO TO ROUTINE

```

::THIS IS USE TO HANDLE THE "GETPRI" MACRO

```

STRAP2:  MOV      (SP),-(SP)  :: MOVE THE PC DOWN
          MOV      4(SP),2(SP) :: MOVE THE PSW DOWN
          RTI              :: RESTORE THE PSW

```

.SBTTL TRAP TABLE

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

ROUTINE

```

STRPAD:  .WORD      STRAP2
          .TYPE      ::CALL=TYPE      TRAP+1(104401)  TTY TYPEOUT ROUTINE
          .TYPEPC    ::CALL=TYPEPC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
          .TYPEPOS   ::CALL=TYPEPOS   TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
          .TYPON     ::CALL=TYPON     TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
          .TYPDS     ::CALL=TYPDS     TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

          .SGTSWR    ::CALL=SGTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

          .CKSWR     ::CALL=CKSWR     TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
          .RDCHR     ::CALL=RDCHR     TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
          .RDLIN     ::CALL=RDLIN     TRAP+11(104411) TTY TYPEIN STRING ROUTINE
          .RDOUT     ::CALL=RDOUT     TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY

```

.SBTTL POWER DOWN AND UP ROUTINES

 :POWER DOWN ROUTINE

```

$PWRDN:  MOV      $SILLUP,2,$PWRVEC  :: SET FOR FAST UP
          MOV      $340,2,$PWRVEC+2 :: PRI3:7
          MOV      R0,-(SP)          :: PUSH R0 ON STACK
          MOV      R1,-(SP)          :: PUSH R1 ON STACK
          MOV      R2,-(SP)          :: PUSH R2 ON STACK
          MOV      R3,-(SP)          :: PUSH R3 ON STACK
          MOV      R4,-(SP)          :: PUSH R4 ON STACK
          MOV      R5,-(SP)          :: PUSH R5 ON STACK
          MOV      $SWR,-(SP)        :: PUSH $SWR ON STACK
          MOV      SP,$$SAVR6        :: SAVE SP
          MOV      $PWRUP,2,$PWRVEC  :: SET UP VECTOR
          HALT
          BR        .-2              :: HANG UP

```

 :POWER UP ROUTINE

```

$PWRUP:  MOV      $SILLUP,2,$PWRVEC  :: SET FOR FAST DOWN
          MOV      $$SAVR6,SP         :: GET SP
          CLR      $$SAVR6           :: WAIT LOOP FOR THE TTY
          IS:      INC      $$SAVR6   :: WAIT FOR THE INC
          BNE      IS              :: OF WORD

```

POWER DOWN AND UP ROUTINES

[illegible]

MAINDEC-11-DECHM-B
DECHMB.F11 PO

M-8 MACY 11 27(732) 23
POWER DOWN AND UP ROUTINES

```

*****
COMMON DH11 SERVICE ROUTINES
*****

```

```
:THIS ROUTINE IS CALLED DURING START UP TO LOAD THE XMITTER
:OUTPUT BUFFER WITH A BINARY COUNT TEST PATTERN
```

```

LDTBF1: MOV      ,TBUF,R1      ;POINT TO START OF BUFFER
        CLR      R2            ;INIT DATA BYTE GENERATOR
1S:     MOV      R2,(R1)+      ;LOAD ONE CHAR
        INC      R2            ;GENERATE NEXT CHAR
        CMP      #400,R2      ;LOADED 256(10) BYTES
        SNE      1S           ;BR IF NOT
        RTS      PC           ;RETURN TO START TESTING

```

```
:THIS ROUTINE SETS UP THE ERROR INFORMATION REQUIRED BY ANY TEST
:USING A "DHI" HEADER
```

```

SUER1:  JSR      PC,SAPS          ;SAVE THE ERROR PSW
        MOV     ST$TNM,RO        ;SAVE THE TEST NO.
        MOV     RO,$REGO         ;SAVE THE TEST NO. FOR ERROR PRINT
        MOV     R1,$REG1        ;SAVE THE DH11 ADDR
        MOV     R2,$REG2        ;SAVE THE REG ADDRESS
        MOV     R6,$REG6        ;SAVE THE SP
        ADD     #2,$REG6        ;CORRECT FOR CALLING JSR
        RTS     PC              ;RETURN TO CALLING ROUTINE

```

```

;THIS ROUTINE IS CALLED BY THOSE TESTS USING A "DH2" HEADER TO
;SAVE THE ERROR INFORMATION IN "DT2"

```

```

SUER2:  JSR      PC,SAPS
SUER2A:  MOV     $TSTNM,R0
        MOV     R0,$REG40
        MOV     R1,$REG41
        MOV     R2,$REG42
        MOV     R3,$REG43
        MOV     R4,$REG44
        MOV     R6,$REG46
        ADD     #2,$REG46
        RTS     PC
        :SAVE THE ERROR PSW
        :GET THE TEST NO.
        :SAVE THE REGISTERS-TEST#
        :SAVE THE OH ADDRESS
        :SAVE THE REGISTER ADDRESS
        :SAVE THE WAS DATA
        :SAVE THE S & B DATA
        :SAVE THE STACK POINTER
        :CORRECT FOR CALLING JSR
        :RETURN TO REPORT ERROR

```

```
:THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
:BUS ERROR AND RSVD INSTR ERROR ROUTINES
```

```
SLER3:  MOV     R0,$REG0      :SAVE THE REGS
        MOV     R1,$REG1
        MOV     R2,$REG2
        RTS     PC           :RETURN TO REPORT ERROR
```

```

:THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
:CAR/BCR MEMORY PATTERNS TESTS

```

```

SUER4: CLR      STMPD      :SAVE THE LINE NO. WRITTEN
        MOVB     LINEA,STMPD
        MOVB     $STSTNM,RO :SAVE THE TEST NUMBER
        MOV      RO,$REGD   :SAVE THE REGISTER INFORMATION

```

[illegible][illegible]

00	00	004767	002552
01	01	0116700	154522
02	02	0210067	154576
03	03	0310167	154574
04	04	0410267	154572
05	05	0510367	154570
06	06	0610467	154568
07	07	0710567	154566
08	08	0810667	154566
09	09	0910767	000002
10	10	1000000	154560

024420	010067	154536
024424	010167	154534
024430	010267	154532
024434	000207	

68	024936	005367	154540
69	024436	116767	003546
70	024436	116700	154426
71	024436	C10067	154502

154532

024460 010167 154500
024464 010266 154506
024470 010366 154514
024474 010467 154522
024500 000207

MOV R1,\$REG1
MOV R2,\$REG2
MOV R3,\$REG3
MOV R4,\$REG4
RTS PC

;RETURN TO PATTERNS TEST

;THIS ROUTINE IS CALLED TO SELECT A NEW LINE NO. BASED ON THE
;VALUE OF THE LINE SELECTION PARAMETER

;CALLING SEQUENCE:

;JSR PC,SELINE
;BR 1\$

;CALL THE ROUTINE
;EXIT BRANCH-ROUTINE MOVES THE RETURN
;PC AROUND THIS BR IF MORE LINES ARE
;YET TO BE TESTED

024502 105767 003505
024506 001010
024510 105167 003477
024514 012767 000001 002462
024522 105067 003464
024526 000405
024530 105267 003456
024534 006367 002444
024540 001407
024544 006767 002436 002432
024548 001767
024552 002716 000002
024556 000402
024560 005067 003426
024564 142777 000017 002400
024572 000207

SELINE: TSTB LINE+1
BNE 1\$
COMB LINE+1
MOV #1,LINMSK
CLRB LINE
BR 2\$
1\$: INCB LINE
ASL LINMSK
SEQ 3\$
2\$: BIT LINMSK,LINSEL
SEQ 1\$
ADD #2,(SP)
BR 4\$
3\$: CLR LINE
4\$: BICB #17,DHADR
RTS PC

;FIRST TIME THROUGH FOR ANY TEST ?
;BR IF NOT
;SET ENTRY FLAG
;INIT SELECT TEST MASK TO TEST LINE 00
;START WITH LINE #00
;GO TEST FOR LINE #00
;GENERATE NEW LINE NO.
;SHIFT SELECT MASK TO TEST NXT LINE
;RETURN TO EXIT BRANCH - ALL LINES DONE
;IS THE LINE SELECTED FOR TEST ??
;BR IF NOT
;MOVE RETURN PC AROUND EXIT BRANCH
;RETURN TO TEST SELECTED LINE
;INIT ENTRY FLAG AND LINE NO. TO 000
;INIT LINE SELECT BITS IN "SCR"
;RETURN TO CALLING TEST

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

;CALLING SEQUENCE

;JSR R5,SUNUM
;ADDR1
;ADDR2

;CALL TO THIS ROUTINE
;ADDRESS OF THE NUMBER TO BE CONVERTED
;ADDRESS OF THE MSG BUFFER SLOT

SUNUM:

MOV R0,-(SP)
MOV R1,-(SP)
MOV R2,-(SP)
MOV (R5)+,R0
MOV (R5)+,R1
MOVB (R0),R0
MOV R0,R2
ASR R2
ASR R2
ASR R2
BIC #177770,R2
ADD #60,R2

::PUSH R0 ON STACK
::PUSH R1 ON STACK
::PUSH R2 ON STACK
;GET ADDRESS OF NUMBER
;GET MSG BUFFER ADDR
;GET NO. TO BE CONVERTED
;SAVE IT IN R2
;SHIFT MSD TO LSD POSITION

024574
024574 010046
024576 010146
024600 010246
024602 012500
024604 012301
024606 111000
024610 010002
024612 006202
024614 006202
024616 006202
024620 042702 177770
024624 062702 000060

;CLR JUNK BITS
;MAKE IT ASCII


```

6366 024630 110221      MOV B    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      MOV B    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

;THIS ROUTINE IS CALLED TO CLEAR THE "CAR" AND "BCR" MEMORIES
;IT ASSUMES THAT THE ADDRESS OF THE "SCR" IS IN R1

6374
6375
6376
6377
6378 024654 005067 154340 CLCABC: CLR      STMP7      ;INIT A COUNTER
6379 024660 116711 154334 15:  MOV B    STMP7,(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      STMP7      ;GENERATE NEW LINE NO.
6383 024700 022767 000020 154312 CMP      #20,STMP7    ;DONE ALL LINES ?
6384 024706 001364      BNE      15          ;BR IF NOT
6385 024710 142711 000017      BICB     #17,(R1)      ;SET "SCR" TO SELECT LINE 00
6386 024714 000207      RTS      PC          ;RETURN TO CALLER

;THIS ROUTINE IS CALLED TO LOAD THE "BCR" MEMORY WITH ALL ONES
;IT ASSUMES THAT THE ADDRESS OF THE SCR IS IN R1

6387
6388
6389
6390 024716 005067 154276 LDBCR: CLR      STMP7      ;INIT A COUNTER
6391 024722 116711 154272 15:  MOV B    STMP7,(R1)    ;SELECT A LINE
6392 024726 012761 177777 000010 MOV      #-1,BCR(R1)  ;LOAD BCR LOC. WITH 177777
6393 024734 005267 154260      INC      STMP7      ;GENERATE NEXT LINE NO.
6394 024740 022767 000020 154252 CMP      #20,STMP7    ;DONE ALL LINES ?
6395 024746 001365      BNE      15          ;BR IF NOT
6396 024750 142711 000017      BICB     #17,(R1)      ;SET "SCR" TO SELECT LINE 00
6397 024754 000207      RTS      PC          ;RETURN TO CALLER

;THIS ROUTINE CALLED TO SET UP FOR PARITY TESTS

6400
6401
6402 024756 012767 000020 154226 SUPPAR: MOV      #20,STMP4    ;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 15:  MOV      #TBUF,CAR(R1)  ;LOAD BUS ADDRESS REWG
6408 025006 016761 154204 000010      MOV      STMP6,BCR(R1)  ;LOAD BYTE COUNT REG
6409 025014 016761 154174 000004      MOV      STMP5,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      MOV B    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      STMP4
6417 025052 001352      BNE      15          ;COUNT ONE LINE SETUP
6418 025054 012704 027452      MOV      #MULTB,R4      ;SET UP TABLE POINTER
6419 025060 016724 154132 25:  MOV      STMP6,(R4)+    ;SET UP BYTE COUNT ENTRY
6420 025064 022704 027512      CMP      #MULTB+40,R4    ;SET UP ALL COUNTS
6421 025070 001373      BNE      25          ;BR IF NOT

```

```

6422 025072 105011          CLRB (R1)          ;INIT SCR TO SELECT LINE 00
6423 025074 000207          RTS PC             ;RETURN TO PARITY TEST
6424
6425          ;THIS ROUTINE AUTOSIZES THE SYSTEM TO DETERMINE THE ADDRESSES AND
6426          ;VECTORS OF THE DH11'S AND DM11-BB'S.
6427
6428 025076 010046          AUTOSZ: MOV R0, -(SP)
6429 025100 005003          CLR R3
6430 025102 012702 027732    MOV #DHADRS,R2      ;POINT TO BEGINNING OF TABLE
6431 025106 005022          25$: CLR (R2)+        ;CLEAR AUTOSIZER TABLES.
6432 025110 005263          INC R3
6433 025112 020327 000102    CMP R3, #102        ;HAVE WE CLEARED ALL ENTRIES?
6434 025116 001373          BNE 25$              ;BRANCH IF NOT.
6435 025120 013746 000004    MOV #4, -(SP)        ;SAVE TRAP VECTOR.
6436 025124 012737 025232 000004    MOV #4, #4      ;SETUP FOR NON-EXISTENT MEMORY TRAP.
6437 025132 012703 030034    MOV #DMADRS,R3      ;SETUP DM ADDRESS TABLE POINTER.
6438 025136 012732 027732    MOV #DHADRS,R2      ;SET UP DH ADDRESS TABLE POINTER.
6439
6440 025142 012701 160020    MOV #160020,R1      ;R1=FIRST ADDRESS TO BE TESTED.
6441
6442 025146 005711          1$: TST (R1)           ;SEE IF ADDRESS IN R1 RESPONDS.
6443 025150 005761 000016    TST 16(R1)          ;CHECK TO SEE IF DEVICE IS MODULO 20.
6444 025154 052711 004000    BIS #4000,(R1)      ;IF IT IS, CONTINUE
6445          ;AND CHECK TO SEE
6446 025160 052711 001000    BIS #1000,(R1)      ;IF THIS ADDRESS CONTAINS
6447 025164 052711 002000    BIS #2000,(R1)      ;A DH-11.
6448 025170 032711 003000    BIT #3000,(R1)      ;CHECK TO INSURE THESE BITS SET.
6449 025174 001410          BEQ 3$                ;IF NOT, BRANCH.
6450
6451 025176 052711 000400    BIS #400,(R1)        ;SET THE MAINTENANCE BIT, THE NON-
6452 025202 032711 002400    BIT #2400,(R1)      ;EXISTENT MEMORY BIT AND THE CLEAR
6453          ;NON-EXISTENT MEMORY INTERRUPT BIT.
6454 025206 001003          BNE 3$                ;IS THIS A DH-11? (BITS 8 AND 10 SHOULD
6455 025210 042711 001000    BIC #1000,(R1)      ;CLEAR IF THIS IS A DH11.)
6456 025214 010122          MOV R1,(R2)+        ;SAVE THE ADDRESS IN THE DH ACR TABLE.
6457
6458
6459
6460
6461 025216 020127 163760    3$: CMP R1, #163760 ;HAVE WE REACHED THE TOP OF THE FLOATING ADDRESSES.
6462 025222 001406          BEQ 5$                ;IF YES, GET OUT.
6463 025224 062701 000020    ADD #20,R1          ;IF NOT, UPDATE ADDRESS AND
6464 025230 000746          BR 1$                ;GO CHECK IT.
6465
6466 025232 012716 025216    4$: MOV #3$, (SP)    ;IF DH ADDRESS DOES NOT RESPOND, GO TO 3$.
6467 025236 000002          RTI
6468
6469          ;TEST FOR DM11 BB ADDRESS
6470
6471 025240 012737 025272 000004 5$: MOV #6$, #4    ;SETUP FOR NON-EXISTENT MEMORY TRAP.
6472 025246 012701 170500    MOV #170500,R1      ;R1=FIRST ADDRESS TO BE TESTED.
6473 025252 005711          21$: TST (R1)         ;SEE IF ADDRESS RESPONDS.
6474 025254 010123          MOV R1,(R3)+        ;IF IT DOES, THIS IS A DM11-BB.
6475          ;SO SAVE THE ADDRESS.
6476 025256 020127 170670    23$: CMP R1, #170670 ;HAVE WE REACHED THE TOP OF THE DM11 ADDRESSES?
6477 025262 001406          BEQ 22$              ;IF YES, GET OUT.

```

H05

 MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 148
 DZDMMB.P11 POWER DOWN AND UP ROUTINES

SEQ 0266

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)+, 2#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							
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, 2#IOTVEC+2	:SET IOT TRAP PRIORITY TO 7.
6504	025360	012737	025470	000020	MOV	#12\$, 2#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, 2(R2)	:SET MAINTENANCE BIT
6521	025432	052772	000300	000000	BIS	#300, 2(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, 2(R2)	:DO A MASTER CLEAR
6529	025460	042772	001000	000000	BIC	#1000, 2(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      #105,(SP)      ;SETUP FOR RETURN.
6537 025506 052772 004000 000000  BIS      #4000,2(R2)      ;DO A MASTER CLEAR
6538 025514 042732 001000      BIC      #1000,2(R2)+      ;CLEAR MAINTENANCE BIT.
6539 025520 000002      RTI
6540
6541      ;FIND DM11 BB VECTORS:
6542
6543 025522 012702 030034      13$:  MOV      #DMADRS,R2      ;SET POINTERS TO BEGINNING OF
6544 025526 012705 030076      MOV      #DMVEC,R5      ;ADR TABLE & VECTOR TABLE.
6545 025532 012737 025614 000020  MOV      #165,2#IOTVEC      ;SET IOT TRAP VECTOR.
6546
6547 025540 005712      14$:  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      #1000,2(R2)      ;SET MAINTENANCE BIT.
6551 025556 052772 000300 000000  BIS      #300,2(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      #4000,2(R2)      ;CLEAR BITS PREVIOUSLY SET.
6558 025604 042772 001000 000000  BIC      #1000,2(R2)      ;CLEAR MAINTENANCE BIT.
6559 025612 000752      BR      14$
6560
6561 025614 011601      16$:  MOV      (SP),R1      ;CALCULATE VECTOR ADDRESS.
6562 025616 162701 000004      SUB      #4,R1      ;SAVE VECTOR ADDRESS.
6563 025622 010125      MOV      R1,(R5)+      ;POP STACK.
6564 025624 022626      CMP      (SP)+,(SP)+      ;SETUP FOR RETURN.
6565 025626 012716 025540      MOV      #145,(SP)      ;CLEAR BITS PREVIOUSLY SET.
6566 025632 052772 004000 000000  BIS      #4000,2(R2)      ;CLEAR MAINTENANCE BIT AND
6567 025640 042732 001000      BIC      #1000,2(R2)+      ;POINT TO NEXT DM11-BB ADDRESS.
6568
6569 025644 000002      RTI
6570
6571 025646 012737 020746 000020  17$:  MOV      #SCOPE,2#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$:  MOV      R4,(R3)+      ;FILL VECTOR AREA WITH ADDRESS OF NEXT
6577 025670 012724 000000      MOV      #0,(R4)+      ;INSTRUCTION (.+2).
6578 025674 022324      CMP      (R3)+,(R4)+      ;NEXT INSTRUCTION IS A HALT.
6579 025676 020427 001000      CMP      R4,#1000      ;UPDATE R3 & R4.
6580 025702 101771      BLOS      18$      ;ARE WE DONE?
6581
6582 025704 013701 027774      MOV      2#DHVEC,R1      ;IF NOT, REPEAT UNTIL ADDRESSES
6583 025710 005737 027776      TST      2#DHVEC+2      ;377 TO 777 ARE DONE.
6584 025714 001403      BEQ      26$      ;LET R1 POINT TO 1ST DH VECTOR ADDRESS.
6585 025716 163701 027776      SUB      2#DHVEC+2,R1      ;IS THERE MORE THAN ONE VECTOR?
6586
6587 025722 005401      NEG      R1      ;BRANCH IF NOT.
6588 025724 010167 002246      26$:  MOV      R1,ADRVEC      ;DETERMINE NUMBER OF ADDRESSES
6589                                ;BETWEEN DH VECTORS (10(8) OR 20(8)).
                                ;MAKE R1 POSITIVE.
                                ;SAVE THAT NUMBER.

```

```

6590 025730 032777 000002 153202 BIT #BIT1,DSWR ;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 TYPDS ;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 TYPDS ;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 TYPDS ;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 TYPDS ;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 $CRLF
6619 026030 005711 TST (R1) ;HAVE WE TYPED ALL CH ENTRIES?
6620 026032 001354 BNE 19$ ;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)+,RO ;GET THE NO. HE TYPED
6633 026056 001407 BEQ 3$ ;BR IF HE TYPED <CR>
6634 026060 022700 000010 CMP #10,RO ;10(8) ADDRESSES BETWEEN VECTORS ?
6635 026064 001406 BEQ 4$ ;BR IF YES
6636 026066 022700 000020 CMP #20,RO ;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$: MOV #20,RO ;SET UP CONSTANT FOR 20(8) ADDRESSES
6640 026102 000207 4$: RTS PC ;RETURN TO CALLER
6641
6642
6643
6644 026104 012700 177777 INPARC: MOV #-1,RO ;SET FLAG IN RO
6645 026110 000167 154034 JMP BEGINA ;GO ASK FOR SELECT PARAMETER

```

6646											
6647	026114	012767	177777	001546	INPARX:	MOV	#-1,VCFLG		:SET SETUP FLAG		
6648	026122	000167	154022			JMP	BEGINA		:GO START UP		
6649											
6650	026126	013701	027732		INPAR:	MOV	3#DHADRS,R1		:MOVE ADDRESS OF FIRST DH INTO R1.		
6651	026132	032777	000001	153000		BIT	#BIT0,3SWR		:ARE PARAMETERS TO BE INPUT MANUALLY?		
6652	026140	001405				BEQ	2\$		:BRANCH IF NOT.		
6653	026142	104401			1\$:	TYPE			:ASK FOR DEVICE ADDRESS		
6654	026144	034731				INMSG1					
6655	026146	104412				RDOCT			:READ IN WHAT IS TYPED		
6656	026150	012601				MOV	(SP)+,R1		:GET THE NO. HE TYPED		
6657	026152	001403				BEQ	INPAR1		:BR IF DEFAULT		
6658	026154	004767	000160		2\$:	JSR	PC,CHKADR		:GO CHECK VALIDITY OF THE ADDR		
6659	026160	000770				BR	1\$		:ERROR BRANCH		
6660											
6661	026162	013701	027774		INPAR1:	MOV	3#DHVEC,R1		:MOVE FIRST DH VECTOR INTO R1.		
6662	026166	032777	000001	152744		BIT	#BIT0,3SWR		:ARE PARAMETERS TO BE INPUT MANUALLY?		
6663	026174	001405				BEQ	2\$		:BRANCH IF NOT.		
6664	026176	104401			1\$:	TYPE			:ASK FOR VECTOR ADDRESS		
6665	026200	034775				INMSG2					
6666	026202	104412				RDOCT			:READ IN WHAT HE TYPES		
6667	026204	012601				MOV	(SP)+,R1		:GET THE ADDRESS		
6668	026206	001403				BEQ	INPAR3		:BR IF DEFAULT		
6669	026210	004767	000240		2\$:	JSR	PC,CHKVCT		:GO CHECK VALIDITY OF VECTOR		
6670	026214	000770				BR	1\$		:ERROR BRANCH		
6671											
6672	026216	013701	030204		INPAR3:	MOV	3#\$DHSEL,R1		:MOVE DEVICE SELECTION PARAMETER INTO R1.		
6673	026222	005700				TST	R0		:DID WE START AT 210?		
6674	026224	100404				BMI	2\$		:BRANCH IF YES.		
6675	026226	032777	000001	152704		BIT	#BIT0,3SWR		:IS PARAMETER TO BE INPUT MANUALLY?		
6676	026234	001405				BEQ	1\$		:BRANCH IF NOT.		
6677	026236	104401			2\$:	TYPE			:ASK FOR DEVICE SELECTION PARAMETER		
6678	026240	035044				INMSG3					
6679	026242	104412				RDOCT			:READ IN WHAT HE TYPES		
6680	026244	012601				MOV	(SP)+,R1		:GET THE SELECT PARAMETER		
6681	026246	001402				BEQ	INPAR4		:BR IF DEFAULT		
6682	026250	010167	000724		1\$:	MOV	R1,DHSEL		:SET UP DH11 SELECTION PARAMETER		
6683											
6684	026254	005700			INPAR4:	TST	R0		:DID WE START AT 210?		
6685	026256	100404				BMI	3\$		:BRANCH IF YES.		
6686	026260	032777	000001	152652		BIT	#BIT0,3SWR		:IS LINE SELECT PARAMETER TO BE INPUT MANUALLY?		
6687	026266	001410				BEQ	1\$		:BRANCH IF NO.		
6688	026270	104401			3\$:	TYPE			:ASK FOR LINE SELECT PARAMETER		
6689	026272	035242				INMSG6					
6690	026274	104412				RDOCT			:GET WHAT HE TYPES		
6691	026276	012601				MOV	(SP)+,R1		:GET PARAMETER		
6692	026300	001403				BEQ	1\$		:BR IF DEFAULT		
6693	026302	010167	000674			MOV	R1,LINSEL		:SET UP LINE SELECT PARAMETER		
6694	026306	000403				BR	2\$		:CONTINUE		
6695	026310	012767	177777	000664	1\$:	MOV	#-1,LINSEL		:SET UP DEFAULT (ALL LINES)		
6696	026316	032777	000400	152614	2\$:	BIT	#BIT0,3SWR		:HALT AFTER SET UP ??		
6697	026324	001403				BEQ	EXPAR		:BR IF NOT		
6698	026326	104401				TYPE			:TYPE CONTINUE MESSAGE PRIOR TO HALTING		
6699	026330	035304				INMSG7					
6700	026332	000000				HALT			:DEPRESS CONTINUE TO RESUME TESTING		
6701	026334	000167	154264		EXPAR:	JMP	START2		:GO START UP THE PROGRAM		

6702						
6703						
6704	026340	020127	160020	CHKADR:	CMP	R1, #160020
6705	026344	002001			BGE	1\$
6706	026346	000437			BR	4\$
6707	026350	020127	160420	1\$:	CMP	R1, #160420
6708	026354	002401			BLT	2\$
6709	026356	000433			BR	4\$
6710	026360	032701	000017	2\$:	BIT	#17, R1
6711	026364	001030			BNE	4\$
6712	026366	062716	000002		ADD	#2, (SP)
6713	026372	012702	027570		MOV	#DHADTB, R2
6714	026376	032777	000001	152534	BIT	#BIT0, #SWR
6715	026404	001011			BNE	3\$
6716	026406	012703	027732		MOV	#DHADRS, R3
6717						
6718	026412	016704	001566		MOV	\$DHSEL, R4
6719	026416	012322		6\$:	MOV	(R3)+, (R2)+
6720						
6721	026420	006204			ASR	R4
6722	026422	005704			TST	R4
6723	026424	001374			BNE	6\$
6724	026426	000411			BR	5\$
6725	026430	010122		3\$:	MOV	R1, (R2)+
6726	026432	062701	000020		ADD	#20, R1
6727	026436	022702	027630		CMP	#DHADTB+40, R2
6728	026442	001372			BNE	3\$
6729	026444	000402			BR	5\$
6730	026446	004401		4\$:	TYPE	
6731	026450	035115			INMSG4	
6732	026452	000207		5\$:	RTS	PC
6733						
6734	026454	020127	000300	CHKVCT:	CMP	R1, #300
6735	026460	002001			BGE	1\$
6736	026462	000436			BR	4\$
6737	026464	020127	001000	1\$:	CMP	R1, #1000
6738	026470	002401			BLT	2\$
6739	026472	000432			BR	4\$
6740	026474	032701	000007	2\$:	BIT	#7, R1
6741	026500	001027			BNE	4\$
6742	026502	062716	000002		ADD	#2, (SP)
6743	026506	012702	027630		MOV	#DHVCTB, R2
6744	026512	032777	000001	152420	BIT	#BIT0, #SWR
6745	026520	001011			BNE	3\$
6746	026522	012703	027774		MOV	#DHVEC, R3
6747						
6748	026526	016704	001452		MOV	\$DHSEL, R4
6749	026532	012322		6\$:	MOV	(R3)+, (R2)+
6750						
6751	026534	006204			ASR	R4
6752	026536	005704			TST	R4
6753	026540	001374			BNE	6\$
6754	026542	000410			BR	5\$
6755	026544	010122		3\$:	MOV	R1, (R2)+
6756	026546	060001			ADD	R0, R1
6757	026550	022702	027670		CMP	#DHVCTB+40, R2

; IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
; BR IF YES
; BR IF NOT
; IS IT BELOW THE HIGH LIMIT?
; BR IF YES
; BR IF NOT
; CORRECT BOUNDARY ?
; BR IF NOT
; MOVE RETURN PC AROUND ERROR BRANCH
; POINT TO BEGIN OF ADDR TABLE
; ARE WE AUTOSIZING?
; BRANCH IF NOT.
; POINT TO BEGINNING OF AUTOSIZER
; DH ADDRESS TABLE.

; MOVE CONTENTS OF AUTOSIZER DH TABLE
; TO THE TABLE USED BY PROGRAM.

; HAVE WE MOVED ALL TABLE ENTRIES?
; BRANCH IF NOT--ONE MORE TIME.
; RETURN TO INPUT ROUTINES.
; SET UP A TABLE ENTRY
; GENERATE NEXT DH11 ADDR
; END OF TABLE ?
; BR IF NOT
; RETURN TO INPUT ROUTINES
; TELL HIM HE GOOFED

; RETURN TO INPUT ROUTINES

; IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
; BR IF YES
; BR IF NOT
; IS IT BELOW THE HIGH LIMIT?
; BR IF YES
; BR IF NOT
; CORRECT BOUNDARY ?
; BR IF NOT
; MOVE RETURN PC AROUND ERROR BRANCH
; POINT TO BEGIN OF VECTOR TABLE
; ARE WE AUTOSIZING?
; BRANCH IF NOT.
; POINT TO BEGINING OF AUTOSIZER
; DH VECTOR TABLE.

; MOVE CONTENTS OF AUTOSIZER VECTOR
; TABLE TO TABLE USED BY PROGRAM.

; HAVE WE MOVED ALL TABLE ENTRIES?
; BRANCH IF NOT--ONE MORE TIME.
; RETURN TO INPUT ROUTINES.
; SET UP A TABLE ENTRY
; GENERATE NEXT DH11 ADDR
; END OF TABLE ?


```

6758 026554 001373      BNE      3$      ;BR IF NOT
6759 026556 000402      BR       5$      ;RETURN TO INPUT ROUTINES
6760 026560 104401      4$:      TYPE      ;TELL HIM HE GOOFED
6761 026562 035166      INMSG5
6762 026564 000207      5$:      RTS       PC      ;RETURN TO INPUT ROUTINES
6763
6764      ;THESE TWO ROUTINES SERVICE UNEXPECTED BUS ERROR AND RSVD INSTP TRAPS
6765
6766 026566 012767 000340 152406 BUSER: MOV      #340,$TMPD ;SAVE THE PSW
6767 026574 010667 152376      MOV      SP,$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      MOV      $TSTNM,R0 ;GET TEST NO.
6771 026610 012706 001100      MOV      #STACK,SP ;RESET THE STACK POINTER
6772 026614 004767 175600      JSR      PC,SUER3 ;GO SET UP ERROR INFO
6773 026620 012767 026630 152262      MOV      #1$,$LPERR ;ALWAYS COME BACK TO 1$
6774 026626 104027      ERROR 27 ;UNEXPECTED BUS ERROR TRAP
6775 026630 000005      1$:      RESET ;PREPARE TO RESTART.
6776 026632 004767 000240      JSR      PC,CHPS1 ;GO CLEAR PSW
6777 026636 000167 154056      JMP      REST1 ;GO RESTART THE PROGRAM
6778
6779 026642 012767 000340 152332 RESERR: MOV      #340,$TMPD ;SAVE THE PSW
6780 026650 010667 152322      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      MOV      $TSTNM,R0 ;GET TEST NO.
6784 026664 012706 001100      MOV      #STACK,SP ;RESET THE STACK POINTER
6785 026670 004767 175524      JSR      PC,SUER3 ;GO SET UP ERROR INFO
6786 026674 012767 026704 152206      MOV      #1$,$LPERR ;ALWAYS COME BACK TO 1$
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      CLR    RBUF(R2)      ;START WITH DATA CHAR OF 000
6829 027036 110364 036112      MOV    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

```

806

MACY11 2707321 23-SEP-76 14:33 PAGE 155
POWER DOWN AND UP ROUTINES

SEC 0273

0071

:THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
 :TEST (ALL 1'S DATA TABLE)

RETAB: 100377 :TEST DATA FOR LINE 00
 100777 :TEST DATA FOR LINE 01
 101377
 101777
 102377
 102777
 103377
 103777
 104377
 104777
 105377
 105777
 106377
 106777
 107377
 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)

RETAB0: 100000 :TEST DATA FOR LINE 00
 100400 :TEST DATA FOR LINE 01
 101000
 101400
 102000
 102400
 103000
 103400
 104000
 104400
 105000
 105400
 106000
 106400
 107000
 107400 :TEST DATA FOR LINE 17

:THIS TABLE USED BY THE AUTO ECHO TEST 2 TO RESET ACTIVE BIT WHEN A
 :LINE IS DONE

LINBIT: BIT00 :DEACTIVATE LINE 00
 BIT01 :DEACTIVATE LINE 01
 BIT02
 BIT03
 BIT04
 BIT05
 BIT06
 BIT07
 BIT08
 BIT09
 BIT10
 BIT11

E06

MAINTEN-11-DZDM-B MACY11 27.732) 23-SEP-76 14:33 PAGE 158
 2204MB.P11 POWER DOWN AND UP ROUTINES

SEQ 0276

027440 010000
 027442 020000
 027444 040000
 027446 100000

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 IN THE MULTI-LINE
 :PARITY TEST TO KEEP TRACK OF TOTAL CHARS RECEIVED

027452 000000

MULPTB: .BLKW 16.

:SIXTEEN WORD COUNTERS TABLE

:THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE BREAK BIT
 :TEST

027512 120000
 027514 120400
 027516 121000
 027518 121400
 027520 122000
 027522 122400
 027524 123000
 027526 123400
 027528 124000
 027530 124400
 027532 125000
 027534 125400
 027536 126000
 027538 126400
 027540 127000
 027542 127400

BRKTAB: 120000
 120400
 121000
 121400
 122000
 122400
 123000
 123400
 124000
 124400
 125000
 125400
 126000
 126400
 127000
 127400

:TEST DATA FOR LINE 00
 :TEST DATA FOR LINE 01

027552 131177
 027554 046600
 027556 177767
 027558 177777
 027560 100077
 027562 042200
 027564 030100

RGMSK1: 131177
 RGMSK2: 46600
 RGMSK3: 177767
 RGMSK4: 177777
 RGMSK5: 100077
 RGMSK6: 42200
 INTMSK: 30100

:MASK TO SPECIFY R/W BITS FOR NORMAL "SCR" REG TEST
 :MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR NORMAL MODE TEST
 :MASK TO SPECIFY R/W BITS IN "LPR"
 :MASK TO SPECIFY R/W BITS IN "BKR"
 :MASK TO SPECIFY R/W BITS IN "SSR"
 :MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR MAINT. MODE TEST
 :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
 027572 160040
 027574 160060
 027576 160100
 027578 160120
 027580 160140
 027582 160160
 027584 160200
 027586 160220
 027588 160240
 027590 160260

CHACTB: 160020
 160040
 160060
 160100
 160120
 160140
 160160
 160200
 160220
 160240
 160260

:ADDRESS OF FIRST DH11
 :ADDRESS OF SECOND DH11

F06

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 159
 DZDMMB.P11 POWER DOWN AND UP ROUTINES

SEG 0277

7039 027616 160300
 7040 027620 160320
 7041 027622 160340
 7042 027624 160360
 7043 027626 160400

160300
 160320
 160340
 160360
 160400

;ADDRESS OF THE LAST DH11

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

DHVCTB: 330

;ADDRESS OF VECTOR FOR FIRST DH11
 ;ADDRESS OF VECTOR FOR SECOND DH11

7044 027630 000330
 7045 027632 000350
 7046 027634 000370
 7047 027636 000410
 7048 027640 000430
 7049 027642 000450
 7050 027644 000470
 7051 027646 000510
 7052 027650 000530
 7053 027652 000550
 7054 027654 000570
 7055 027656 000610
 7056 027660 000630
 7057 027662 000650
 7058 027664 000670
 7059 027666 000710

350
 370
 410
 430
 450
 470
 510
 530
 550
 570
 610
 630
 650
 670
 710

;ADDRESS OF VECTOR FOR LAST DH11

7060 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

BRLVL: 120240

;BRLEVELS FOR FIRST DH11
 ;BR LEVELS FOR SECOND DH11

7061 027672 120240
 7062 027674 120240
 7063 027676 120240
 7064 027700 120240
 7065 027702 120240
 7066 027704 120240
 7067 027706 120240
 7068 027710 120240
 7069 027712 120240
 7070 027714 120240
 7071 027716 120240
 7072 027720 120240
 7073 027722 120240
 7074 027724 120240
 7075 027726 120240
 7076 027730 120240

120240
 120240
 120240
 120240
 120240
 120240
 120240
 120240
 120240
 120240
 120240
 120240
 120240
 120240
 120240
 120240

;BR LEVELS FOR LAST DH11

;THIS DH ADDRESS TABLE IS FILLED BY THE AUTOSIZER.

DHADRS:

7091 027732 000000
 7092 027732 000000
 7093 027734 000000
 7094 027736 000000

.WORD 0
 .WORD 0
 .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

;THIS DH VECTOR TABLE IS FILLED BY THE AUTOSIZER.

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

;THIS DM ADDRESS TABLE IS FILLED BY THE AUTOSIZER.

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				

DMVEC:

[illegible]

SLNSEL:

```

..WORD 1
..WORD 121
..WORD 421
..WORD 100
..WORD 421
..WORD 40
..WORD 100
..WORD 4000
..WORD 400
..WORD 10000
..WORD 40000
..WORD 100000
..WORD 400000
..WORD 1000000
..WORD 4000000
..WORD 10000000

```

ADRVEC: 0

CHMCSR: 0

DHMLSR: 0

SDHSEL: 0

CHRLVL: .BYTE 0

CHTLVL: .BYTE 0

OHNL: 00

LINE: 00
LINEA: 00

LINEA: 0
-ADDRESS: 1

;ADDRESS POINTERS TO SET UP THELES WHEN INPUTTING PARAMETERS

ADPTR: 0

VCPTR: 0

BRPTR: 0

THE FOLLOWING TEN CONSTANTS ARE USED BY THE DM11-BB PORTION OF THIS PROGRAM.

:ADDRESSES BETWEEN VECTORS - FILLED BY THE AUTOSIZER

DM11-BB CONTROL AND STATUS REGISTER.

```

;DMI1-BB LINE STATUS REGISTER.

```

```

;DEVICE SELECT PARAMETER - FILLED BY THE AUTOSIZER.

```

;BR LEVEL FOR RCVR

BR LEVEL FOR XMITTER

CONTAINS NUMBER OF THE DATA UNDER TEST

;CONTINUES NUMBER OF THE LINE UNDER TEST
LOCATION TO SAME LINE NUMBER

```

;LOCATION TO SAVE LINE NUMBER
;UP TOPLES WHEN INPUTTING DATA

```

UP THELES WHEN INPUTTING PARAMETERS

:POINTS TO ADDRESS TABLE

:POINTS TO VECTOR TABLE

POINTS TO BR LEVEL TABLE

TANTS ARE USED BY THE DM11-BB PORTION OF THIS PROGRAM.

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 162
 DZDMMB.P11 POWER DOWN AND UP ROUTINES

SEQ 0280

7207	000400	STEP=400
7208	000001	LINENA=1
7209	000002	TRMRDY=2
7210	000004	RS=4
7211	000010	SECTX=10
7212	000020	SECRX=20
7213	000040	CS=40
7214	000100	CO=100
7215	000200	RING=200
7216	002000	CLRMUX=2000
7217		
7218	030224 000000	TDATA1: 0
7219	030226 000037	TDATA2: 37
7220	030230 000077	77
7221	030232 000177	177
7222	030234 000377	377
7223	030236 000000	TITLEG: 0
7224	030240 000000	TIMEA: 0
7225	030242 000000	TIMEB: 0
7226	030244 000000	TIMEC: 0
7227	030246 000000	TINCL: 0
7228		
7229		
7230		

:DATA BUFFER FOR BASIC DATA TEST
 :TEST DATA FOR FIVE BIT CHAR
 :TEST DATA FOR SIX BIT CHAR
 :TEST DATA FOR SEVEN BIT CHAR
 :TEST DATA FOR EIGHT BIT CHAR
 :FLAG TO ALLOW PRINTING TITLE ONLY ONCE
 :GENERAL PURPOSE TIMERS
 :TIMER FOR TIMING TESTS
 :CONTAINS TWO NULL CHARS USED BY BREAK TEST

```

7231 ::*****
7232 :ERROR MESSAGE INFORMATION - MESSAGE BUFFERS AND POINTERS
7233 :*****
7234 :
7235 :INFORMATION FOR MESSAGE 1
7236
7237 030250 044104 030461 051040 EM1: .ASCIZ 'DH11 REGISTER REFERENCE CAUSED TIMEOUT'
7238 030256 043505 051511 042524
7239 030264 020122 042522 042506
7240 030272 042522 041516 020105
7241 030300 040503 051525 042105
7242 030306 052040 046511 047505
7243 030314 052125 000
7244 030317 040 050050 024503 DH1: .ASCIZ ' (PC) (PS) (SP) TEST DEVADR REGADR'
7245 030324 020040 020040 050050
7246 030332 024523 020040 020040
7247 030340 051450 024520 020040
7248 030346 020040 042524 052123
7249 030354 020040 042040 053105
7250 030362 042101 020122 051040
7251 030370 043505 042101 000122
7252 .EVEN
7253 030376 001116 001202 001176 DT1: .WORD $ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,0
7254 030404 001162 001164 001166
7255 030412 000000
7256
7257 :INFORMATION FOR MESSAGE 2
7258
7259 030414 054523 052123 046505 EM2: .ASCIZ 'SYSTEM CONTROL REGISTER ERROR'
7260 030422 041440 047117 051124
7261 030430 046117 051040 043505
7262 030436 051511 042524 020122
7263 030444 051105 047522 000122
7264 030452 024040 041520 020051 DH2: .ASCIZ ' (PC) (PS) (SP) TEST DEVADR REGADR WAS S B'
7265 030460 020040 024040 051520
7266 030466 020051 020040 024040
7267 030474 050123 020051 020040
7268 030502 052040 051505 020124
7269 030510 020040 042504 040526
7270 030516 051104 020040 042522
7271 030524 040507 051104 020040
7272 030532 053440 051501 020040
7273 030540 020040 051440 041057
7274 030546 000
7275 .EVEN
7276 030550 030550
7277 030550 001116 001202 001176 DT2: .WORD $ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4,0
7278 030556 001162 001164 001166
7279 030564 001170 001172 000000
7280
7281 :INFORMATION FOR MESSAGE 3
7282
7283 030572 044104 030461 046440 EM3: .ASCIZ 'DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG'
7284 030600 051501 042524 020122
7285 030606 046103 040505 020122
7286 030614 040506 046111 042105
7287 030622 052040 020117 046103

```

K06

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 164
 DZDMMB.P11 POWER DOWN AND UP ROUTINES

SEQ 0282

7287 030630 020122 050123 041505
 7288 030636 043111 042511 020104
 7289 030644 042522 000107

; INFORMATION FOR MESSAGE 4

7293 030650 044514 042516 050040
 7294 030656 051101 046501 052105
 7295 030664 051105 051040 043505
 7296 030672 051511 042524 020122
 7297 030700 051105 047522 000122

EM4: .ASCIZ 'LINE PARAMETER REGISTER ERROR'

; INFORMATION FOR MESSAGE 5

7301 030706 051102 040505 020113
 7302 030714 047503 052116 047522
 7303 030722 020114 042522 044507
 7304 030730 052123 051105 042440
 7305 030736 051122 051117 000

EM5: .ASCIZ 'BREAK CONTROL REGISTER ERROR'

; INFORMATION FOR MESSAGE 6

7309 030743 123 046111 020117
 7310 030750 052123 052101 051525
 7311 030756 051040 043505 051511
 7312 030764 042524 020122 051105
 7313 030772 047522 000122

EM6: .ASCIZ 'SILO STATUS REGISTER ERROR'

; INFORMATION FOR MESSAGE 7

7317 030776 052503 051122 047105
 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

EM7: .ASCIZ 'CURRENT ADDRESS REGISTER ERROR - LINE #XX'

; INFORMATION FOR MESSAGE 10

7327 031050 054502 042524 041440
 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

EM10: .ASCIZ 'BYTE COUNTER REGISTER ERROR - LINE #XX'

; INFORMATION FOR MESSAGE 11

7337 031117 125 042516 050130
 7338 031124 041505 042524 020104
 7339 031132 044104 030461 051040
 7340 031140 053103 020122 047111
 7341 031146 042524 051122 050125
 7342 031154 000124

EM11: .ASCIZ 'UNEXPECTED DM11 RCVR INTERRUPT'

L06

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 165
 DZDMMB.P11 POWER DOWN AND UP ROUTINES

SEQ 0293

7343
7344
7345
7346
7347
7348
7349
7350
7351
7352
7353
7354
7355
7356
7357
7358
7359
7360
7361
7362
7363
7364
7365
7366
7367
7368
7369
7370
7371
7372
7373
7374
7375
7376
7377
7378
7379
7380
7381
7382
7383
7384
7385
7386
7387
7388
7389
7390
7391
7392
7393
7394
7395
7396
7397
7398

; INFORMATION FOR MESSAGE 12

EM12: .ASCIZ 'UNEXPECTED DH11 XMITTR INTERRUPT'

; INFORMATION FOR MESSAGE 13

EM13: .ASCIZ 'CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT'

; INFORMATION FOR MESSAGE 14

EM14: .ASCIZ 'TRANSMITTER NPR LOGIC ERROR - LINE # '

; INFORMATION FOR MESSAGE 15

EM15: .ASCIZ 'XMITTR FAILED TO INTERRUPT - LINE # '

; INFORMATION FOR MESSAGE 16

EM16: .ASCIZ 'RCVR FAILED TO INTERRUPT'

; INFORMATION FOR MESSAGE 17

EM17: .ASCIZ 'TRANSMITTER TIMING ERROR - LINE # '

M06

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 166
 DZDMMB.P11 POWER DOWN AND UP ROUTINES

SEQ 0284

7399	031476	026440	046040	047111
7400	031504	020105	020043	000040
7401	031512	024040	041520	020051
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		

; INFORMATION FOR MESSAGE 20

7415	031610	042522	042503	053111
7416	031616	051105	052040	046511
7417	031624	047111	020107	051105
7418	031632	047522	020122	020055
7419	031640	044514	042516	021440
7420	031646	020040	000	

; INFORMATION FOR MESSAGE 21

7425	031651	122	053103	020122
7426	031656	040506	046111	042105
7427	031664	052040	020117	047111
7428	031672	042524	051122	050125
7429	031700	020124	020055	044514
7430	031706	042516	021440	020040
7431	031714	000		

; INFORMATION FOR MESSAGE 22

7434	031715	103	040510	020122
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	

; INFORMATION FOR MESSAGE 23

7445	031771	102	051501	041511
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
7452	032040	020040	024040	051520
7453	032046	020051	020040	024040
7454	032054	050123	020051	020040

DH7: .ASCIZ ' (PC) (PS) (SP) TEST DEVADR CHRLNG JAS S B'

DH6: .ASCIZ ' (PC) (PS) (SP) TEST DEVADR SPEED TIMEB TIMEC'

EM20: .ASCIZ 'RECEIVER TIMING ERROR - LINE # '

EM21: .ASCIZ 'RCVR FAILED TO INTERRUPT - LINE # '

EM22: .ASCIZ 'CHAR AVAIL FAILED TO SET ON TIME - LINE # '

EM23: .ASCIZ 'BASIC DATA TEST ERROR - LINE # '

N06

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 167
DZDMMB.P11 POWER DOWN AND UP ROUTINES

SEQ 0285

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		

; INFORMATION FOR MESSAGE 24

7465	032127	101	052125	020117
7466	032134	041505	047510	052040
7467	032142	051505	020124	051105
7468	032150	047522	020122	020055
7469	032156	044514	042516	021440
7470	032164	020040	000	

EM24: .ASCIZ 'AUTO ECHO TEST ERROR - LINE # '

7471

; INFORMATION FOR MESSAGE 25

7474	032167	102	042522	045501
7475	032174	041040	052111	052040
7476	032202	051505	020124	051105
7477	032210	047522	020122	020055
7478	032216	044514	042516	021440
7479	032224	020040	000	

EM25: .ASCIZ 'BREAK BIT TEST ERROR - LINE # '

7480

; INFORMATION FOR MESSAGE 26

7483	032227	110	046101	026506
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

EM26: .ASCIZ 'HALF-DUPLEX TEST ERROR - LINE # '

7489

; INFORMATION FOR MESSAGE 27

7492	032271	125	042516	050130
7493	032276	041505	042524	020104
7494	032304	052502	020123	051105
7495	032312	047522	020122	051124
7496	032320	050101	000	
7497	032323	040	050050	024503
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

EM27: .ASCIZ 'UNEXPECTED BUS ERROR TRAP'

7505				
7506	032402	001116	001202	001176
7507	032410	001162	001164	001166
7508	032416	000000		

.EVEN

CT3: .WORD \$ERRPC,\$TMFC,\$REG6,\$REG0,\$REG1,\$REG2,0

7509

; INFORMATION FOR MESSAGE 30

7510

: INFORMATION FOR MESSAGE 31

EN31: .ASCIZ 'AUTO ECHO DATA COMPARE ERROR - LINE # '

CH4: .ASCIZ ' (PC) (PS) SF TEST WALKER BIRTH DATE

: INFORMATION FOR MESSAGE 32

FM32: .ASCII 'AUTO ECHO TEST TIMEOUT - LINE # '

DHS: .ASCIIZ * (PC) (LPRG) TEST

.EVEN
 DT4: .WORD \$ERRPC,\$TMP0,\$TMP2,0

: INFORMATION FOR MESSAGE 33

EM33: .ASCIZ 'PARITY LOGIC TEST ERROR - LINE # '

: INFORMATION FOR MESSAGE 34

EM34: .ASCII 'MULTI-LINE PARITY DATA TEST ERROR - LINE # - SUBTEST # '

C07

MAINDEC-11-DDMM-S MACY11 27(732) 23-SEP-76 14:33 PAGE 169
 DDMMB.P11 POWER DOWN AND UP ROUTINES

SEG 0287

0332770	046055	047111	020105
0332776	040520	044522	054524
0333004	042040	052101	020101
0333012	042524	052123	042440
0333020	051122	051111	026440
0333026	046040	047111	020105
0333034	020043	020040	020055
0333042	052523	052103	051505
0333050	020124	020043	020040

; INFORMATION FOR MESSAGE 35

EM35: .ASCIZ 'MULTI-LINE PARITY DATA TEST TIMEOUT'

0333056	052515	052114	026511
0333064	044514	042516	050340
0333072	051101	052111	020131
0333100	040504	040524	052040
0333106	051505	020124	044524
0333114	042515	052517	000124
0333122	024040	041520	020051
0333130	020040	046050	051120
0333136	024507	020040	041501
0333144	046124	047111	000
0333152	033152		
0333158	001116	001202	001210
0333160	000000		

.EVEN
 DT6: .WORD SERRPC, STMP0, STMP3, 0

; INFORMATION FOR MESSAGE 36

EM36: .ASCIZ 'CHAR AVAILABLE TIMEOUT'

0333162	044103	051101	040440
0333170	040526	046111	041101
0333176	042514	052040	046511
0333204	047505	052125	000

; INFORMATION FOR MESSAGE 37

EM37: .ASCIZ 'DATA COMPARE ERROR - LINE # '

0333211	047504	052101	020101
0333216	047503	050115	051101
0333224	020105	051105	047522
0333232	020122	020055	044514
0333240	042516	021440	020040
0333246	000		

; INFORMATION FOR MESSAGE 40

EM40: .ASCIZ 'BUFFER ACTIVE REG ERROR - LINE # '

0333247	020102	043125	042506
0333254	020122	041501	044524
0333262	042526	051040	043505
0333270	042440	051122	051117
0333276	026440	046040	047111
0333304	020105	020043	000040

; INFORMATION FOR MESSAGE 41

EM41: .ASCIZ 'RCVR FALSE INTERRUPT'

0333312	041522	051126	043040
0333320	046101	042523	044440
0333326	052116	051105	052522

007

MAINDEC-11-DZDM-B MAC11 27-732 23-SEP-76 14:33 PAGE 170
 DZDM-B.P11 POWER DOWN AND LP ROUTINES

SEC 0298

033334 052120 000

: INFORMATION FOR MESSAGE 42

033337 053123 046111 020117
 033344 053117 051105 046105
 033352 053517 042440 051122
 033360 051117 000

EM42: .ASCIZ 'SILO OVERFLOW ERROR'

: INFORMATION FOR MESSAGE 43

033353 053123 046111 020117
 033373 053117 051105 046105
 033376 053517 043040 044501
 033404 042514 020104 047524
 033412 042440 047105 051105
 033420 052101 020105 047522
 033426 051126 044440 052116
 033434 051125 052522 052120
 033442 000

EM43: .ASCIZ 'SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT'

: INFORMATION FOR MESSAGE 44

033443 052116 047117 042440
 033450 020130 042515 047515
 033456 054522 043040 044501
 033464 042514 020104 047524
 033472 043440 047105 051105
 033500 052101 020105 046530
 033506 052111 051124 044440
 033514 052116 051105 052522
 033522 052120 000

EM44: .ASCIZ 'NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT'

: INFORMATION FOR MESSAGE 45

033500 054513 044515 020124
 033506 047105 042516 042040
 033514 044501 042514 020104
 033522 047524 043440 047105
 033530 051105 052101 020105
 033536 046530 052111 051124
 033544 044440 052116 051105
 033576 052522 052120 000

EM45: .ASCIZ 'XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT'

: INFORMATION FOR MESSAGE 46

033603 052103 051125 042522
 033610 052116 040440 042104
 033616 042522 051523 046440
 033624 046505 051117 020131
 033632 043520 052124 051105
 033640 051516 052040 051505
 033646 020124 051105 047522
 033654 020122 020055 044514
 033662 042516 021440 020040
 033670 000

EM46: .ASCIZ 'CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE # '

E07

MACY11 27-7321 23-SEP-76 14:33 PAGE 171
POWER DOWN AND CP ROUTINES

SEQ 0289

CH10: .ASCIZ ' (PC) LINEWR PATRN TEST DEVADR REGADR WAS S/B'

023371	020040	050050	024503
023376	020040	046040	047111
023374	050050	020122	050040
023371	050050	051124	020116
023372	020040	042524	052122
023376	020040	042040	053105
023374	042101	020122	051040
023371	042505	042101	020122
023372	020040	040527	020122
023376	020040	020040	021523
023374	000102		

.EVEN

DT5: .WORD \$ERRPC,\$TMP0,\$TMP1,\$REG0,\$REG1,\$REG2,\$REG3,\$REG4,0

023376	001116	001202	001204
023374	001162	001164	001166
023372	001170	001172	000000

; INFORMATION FOR MESSAGE 47

0234010	054532	042524	041440
0234016	052517	052116	046440
0234024	046505	051117	020131
0234032	046505	052124	051105
0234040	051516	052040	051505
0234046	020122	051105	047522
0234054	020122	020055	044514
0234070	000	021440	020040

EM47: .ASCIZ 'BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE # '

; INFORMATION FOR MESSAGE 50

0234080	021440	051505	020124
0234086	021440	042505	052517
0234094	021440	040527	052111
0234102	021440	020107	047506
0234110	021440	046530	052111
0234118	021440	047117	020105
0234126	021440	044514	042516
0234134	021440	020040	000

EM50: .ASCIZ 'TEST TIMEOUT WAITING FOR XMIT DONE - LINE # '

; INFORMATION FOR MESSAGE 51

0234147	043517	051120	046040
0234154	051505	041511	052040
0234162	051105	020124	020062
0234170	024040	047522	000122
0234176	020040	041520	020051
0234184	052103	044514	040516
0234192	041516	020040	044514
0234200	052040	051505	020040
0234208	020040	042504	040526
0234216	020040	020040	042522
0234224	020040	051104	020040
0234232	020040	051501	020040
0234240	000	051440	041057

EM51: .ASCIZ 'NPR LOGIC TEST 2 ERROR'

CH11: .ASCIZ ' (PC) LINACT LINCHK TEST DEVADR REGADR WAS S B'

F07

MAINCEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 172
DZDMMB.P11 POWER DOWN AND UP ROUTINES

SEQ 0290

; INFORMATION FOR MESSAGE 52

EM52: .ASCIZ 'BASIC DATA COMPARE ERROR'

; INFORMATION FOR MESSAGE 53

DH12: .ASCIZ ' (PC) SPEED (SP) TEST DEVAOR REGAOR WAS S/B'

; INFORMATION FOR MESSAGE 55

DH13: .ASCIZ ' (PC) (PS) (SP) TEST DEVAOR CHALNG SCRNAS SCRS/B'

; INFORMATION FOR MESSAGE 56

EM56: .ASCIZ 'OVERRUN BIT FAILED TO SET - LINE # '

; INFORMATION FOR MESSAGE 57

EM57: .ASCIZ 'STORAGE OVERFLOW BIT FAILED - LINE # '

.EVEN

G07

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 173
 DZDMMB.P11 POWER DOWN AND UP ROUTINES

SEQ 0291

: MISCELLANEOUS MESSAGES

TITLE: .ASCIZ <15><12>'MAINDEC-11-DZDMM-B DH11 DIAGNOSTIC'<15><12>

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

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

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

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

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

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

INMSG6: .ASCIZ <15><12>'TYPE LINE SELECTION PARAMETER'<15><12>

INMSG7: .ASCIZ <15><12>'DEPRESS "CONTINUE" TO START TESTING'<15><12>

H07

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 174
 DZDMMB.P11 POWER DOWN AND UP ROUTINES

SEQ 0292

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	VOWC: .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 DM11'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 DM 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	025461	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>'DM11, 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>'DM11 DM11 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 DZDMM DIAGNOSTIC'<15><12>

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 175
 DZDMMB.P11 POWER DOWN AND UP ROUTINES

SEQ 0293

```

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
7910 036032 046504 030461 041055
7911 036040 020102 047506 047125
7912 036046 006504 000012
7913
7914
7915
7916 036052 000020
7917
7918
7919
7920 036112 000400
7921
7922
7923
7924
7925
7926 037112 000400
7927
7928 000001
  
```

MSG5: .ASCIZ <15><12>/NO DM11-BB FOUND/<15><12>

.EVEN
 ;SIXTEEN CHAR COUNTERS USED BY THE AUTO ECHO TEST #3

RCNT: .BLKW 16.

;256. WORD RECEIVER INPUT BUFFER

RBUF: .BLKW 256.

;256(10) BYTE TRANSMITTER OUTPUT DATA BUFFER

.EVEN
 TBUF: .BLKB 256.

.END

J07

SE0 0294

BASE	=	000000	299						
PCW1	=	000000	301						
PCW2	=	000000	302						
PCPJOP	=	000000	303						
PCW3	=	000000	304						
PCW4	=	000000	305						
PCW5	=	000000	306						
PCW6	=	000000	307						
PCW7	=	000000	308						
PCW8	=	000000	309						
PCW9	=	000000	310						
PCW10	=	000000	311						
PCW11	=	000000	312						
PCW12	=	000000	264						
PCW13	=	000000	300						
PCW14	=	000000	752*	767	5456*	7201*			
PCW15	=	000000	741	7190*					
PCW16	=	000000	269						
PCW17	=	000000	270						
PCW18	=	000000	4639	4767	4807	4811	4812	6931*	
PCW19	=	000000	6951*						
PCW20	=	000000	261						
PCW21	=	000000	296						
PCW22	=	000000	290						
PCW23	=	000000	293						
PCW24	=	000000	296						
PCW25	=	000000	280						
PCW26	=	000000	288						
PCW27	=	000000	291						
PCW28	=	000000	294						
PCW29	=	000000	266						
PCW30	=	000000	267						
PCW31	=	000000	260						
PCW32	=	000000	281						
PCW33	=	000000	289						
PCW34	=	000000	292						
PCW35	=	000000	295						
PCW36	=	000000	263						
PCW37	=	000000	5945*						
PCW38	=	000000	5933	5901	5943*				
PCW39	=	000000	5942*						
PCW40	=	000000	5903	5944*					
PCW41	=	000000	271						
PCW42	=	000000	262						
PCW43	=	000000	265						
PCW44	=	000000	272						
PCW45	=	000000	6428*						
PCW46	=	000000	297						

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 178
DZDMMB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

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*
BCR	= 000010	6879*												
		1895	2051	2215	23	3063*	3119	3124	3227	3238	3583*	3776*	3891*	4011*
		4131*	4202	4277*	4577*	4660*	4770*	4906*	4937*	5056*	5551*	6381*	6393*	6409*
		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	5543
		5607	6980											
BIT1	= 000002	178*	6590											
BIT10	= 002000	159*	2853	5222	6981									
BIT11	= 004000	158*	903	2603	2698	2768	2848	2926	3061	3100	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*	6980*				
BPTVEC	= 000014	186*				</								

[illegible]

[illegible]

SEQ 0298

[illegible]

№	Տվյալներ	Ընդամենը	Տվյալներ	Ընդամենը
1	...	...	...	...
2	...	...	...	...
3	...	...	...	...
4	...	...	...	...
5	...	...	...	...
6	...	...	...	...
7	...	...	...	...
8	...	...	...	...
9	...	...	...	...
10	...	...	...	...

SECRET

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

F08

MACV-11-0224-B MACV 11 27,732 23-SEP-76 14:33 PAGE 186
 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0303

		6613*	6632	6656	6667	6680	6691	6712*	6742*	6767	6768	6769	6771*	6780
		6614*	6633	6657	6668	6681	6692	6713*	6743*	6768	6769	6770	6772*	6781
		6615*	6634	6658	6669	6682	6693	6714*	6744*	6769	6770	6771	6773*	6782
STAC =	001100	6616*	6635	6659	6670	6683	6694	6715*	6745*	6770	6771	6772	6774*	6783
STAC =	001100	6617*	6636	6660	6671	6684	6695	6716*	6746*	6771	6772	6773	6775*	6784
STAC =	001100	6618*	6637	6661	6672	6685	6696	6717*	6747*	6772	6773	6774	6776*	6785
STAC =	001100	6619*	6638	6662	6673	6686	6697	6718*	6748*	6773	6774	6775	6777*	6786
STAC =	001100	6620*	6639	6663	6674	6687	6698	6719*	6749*	6774	6775	6776	6778*	6787
STAC =	001100	6621*	6640	6664	6675	6688	6699	6720*	6750*	6775	6776	6777	6779*	6788
STAC =	001100	6622*	6641	6665	6676	6689	6700	6721*	6751*	6776	6777	6778	6780*	6789
STAC =	001100	6623*	6642	6666	6677	6690	6701	6722*	6752*	6777	6778	6779	6781*	6790
STAC =	001100	6624*	6643	6667	6678	6691	6702	6723*	6753*	6778	6779	6780	6782*	6791
STAC =	001100	6625*	6644	6668	6679	6692	6703	6724*	6754*	6779	6780	6781	6783*	6792
STAC =	001100	6626*	6645	6669	6680	6693	6704	6725*	6755*	6780	6781	6782	6784*	6793
STAC =	001100	6627*	6646	6670	6681	6694	6705	6726*	6756*	6781	6782	6783	6785*	6794
STAC =	001100	6628*	6647	6671	6682	6695	6706	6727*	6757*	6782	6783	6784	6786*	6795
STAC =	001100	6629*	6648	6672	6683	6696	6707	6728*	6758*	6783	6784	6785	6787*	6796
STAC =	001100	6630*	6649	6673	6684	6697	6708	6729*	6759*	6784	6785	6786	6788*	6797
STAC =	001100	6631*	6650	6674	6685	6698	6709	6730*	6760*	6785	6786	6787	6789*	6798
STAC =	001100	6632*	6651	6675	6686	6699	6710	6731*	6761*	6786	6787	6788	6790*	6799
STAC =	001100	6633*	6652	6676	6687	6700	6711	6732*	6762*	6787	6788	6789	6791*	6800
STAC =	001100	6634*	6653	6677	6688	6701	6712	6733*	6763*	6788	6789	6790	6792*	6801
STAC =	001100	6635*	6654	6678	6689	6702	6713	6734*	6764*	6789	6790	6791	6793*	6802
STAC =	001100	6636*	6655	6679	6690	6703	6714	6735*	6765*	6790	6791	6792	6794*	6803
STAC =	001100	6637*	6656	6680	6691	6704	6715	6736*	6766*	6791	6792	6793	6795*	6804
STAC =	001100	6638*	6657	6681	6692	6705	6716	6737*	6767*	6792	6793	6794	6796*	6805
STAC =	001100	6639*	6658	6682	6693	6706	6717	6738*	6768*	6793	6794	6795	6797*	6806
STAC =	001100	6640*	6659	6683	6694	6707	6718	6739*	6769*	6794	6795	6796	6798*	6807
STAC =	001100	6641*	6660	6684	6695	6708	6719	6740*	6770*	6795	6796	6797	6799*	6808
STAC =	001100	6642*	6661	6685	6696	6709	6720	6741*	6771*	6796	6797	6798	6800*	6809
STAC =	001100	6643*	6662	6686	6697	6710	6721	6742*	6772*	6797	6798	6799	6801*	6810
STAC =	001100	6644*	6663	6687	6698	6711	6722	6743*	6773*	6798	6799	6800	6802*	6811
STAC =	001100	6645*	6664	6688	6699	6712	6723	6744*	6774*	6799	6800	6801	6803*	6812
STAC =	001100	6646*	6665	6689	6700	6713	6724	6745*	6775*	6800	6801	6802	6804*	6813
STAC =	001100	6647*	6666	6690	6701	6714	6725	6746*	6776*	6801	6802	6803	6805*	6814
STAC =	001100	6648*	6667	6691	6702	6715	6726	6747*	6777*	6802	6803	6804	6806*	6815
STAC =	001100	6649*	6668	6692	6703	6716	6727	6748*	6778*	6803	6804	6805	6807*	6816
STAC =	001100	6650*	6669	6693	6704	6717	6728	6749*	6779*	6804	6805	6806	6808*	6817
STAC =	001100	6651*	6670	6694	6705	6718	6729	6750*	6780*	6805	6806	6807	6809*	6818
STAC =	001100	6652*	6671	6695	6706	6719	6730	6751*	6781*	6806	6807	6808	6810*	6819
STAC =	001100	6653*	6672	6696	6707	6720	6731	6752*	6782*	6807	6808	6809	6811*	6820
STAC =	001100	6654*	6673	6697	6708	6721	6732	6753*	6783*	6808	6809	6810	6812*	6821
STAC =	001100	6655*	6674	6698	6709	6722	6733	6754*	6784*	6809	6810	6811	6813*	6822
STAC =	001100	6656*	6675	6699	6710	6723	6734	6755*	6785*	6810	6811	6812	6814*	6823
STAC =	001100	6657*	6676	6700	6711	6724	6735	6756*	6786*	6811	6812	6813	6815*	6824
STAC =	001100	6658*	6677	6701	6712	6725	6736	6757*	6787*	6812	6813	6814	6816*	6825
STAC =	001100	6659*	6678	6702	6713	6726	6737	6758*	6788*	6813	6814	6815	6817*	6826
STAC =	001100	6660*	6679	6703	6714	6727	6738	6759*	6789*	6814	6815	6816	6818*	6827
STAC =	001100	6661*	6680	6704	6715	6728	6739	6760*	6790*	6815	6816	6817	6819*	6828
STAC =	001100	6662*	6681	6705	6716	6729	6740	6761*	6791*	6816	6817	6818	6820*	6829
STAC =	001100	6663*	6682	6706	6717	6730	6741	6762*	6792*	6817	6818	6819	6821*	6830
STAC =	001100	6664*	6683	6707	6718	6731	6742	6763*	6793*	6818	6819	6820	6822*	6831
STAC =	001100	6665*	6684	6708	6719	6732	6743	6764*	6794*	6819	6820	6821	6823*	6832
STAC =	001100	6666*	6685	6709	6720	6733	6744	6765*	6795*	6820	6821	6822	6824*	6833
STAC =	001100	6667*	6686	6710	6721	6734	6745	6766*	6796*	6821	6822	6823	6825*	6834
STAC =	001100	6668*	6687	6711	6722	6735	6746	6767*	6797*	6822	6823	6824	6826*	6835
STAC =	001100	6669*	6688	6712	6723	6736	6747	6768*	6798*	6823	6824	6825	6827*	6836
STAC =	001100	6670*	6689	6713	6724	6737	6748	6769*	6799*	6824	6825	6826	6828*	6837
STAC =	001100	6671*	6690	6714	6725	6738	6749	6770*	6800*	6825	6826	6827	6829*	6838
STAC =	001100	6672*	6691	6715	6726	6739	6750	6771*	6801*	6826	6827	6828	6830*	6839
STAC =	001100	6673*	6692	6716	6727	6740	6751	6772*	6802*	6827	6828	6829	6831*	6840
STAC =	001100	6674*	6693	6717	6728	6741	6752	6773*	6803*	6828	6829	6830	6832*	6841
STAC =	001100	6675*	6694	6718	6729	6742	6753	6774*	6804*	6829	6830	6831	6833*	6842
STAC =	001100	6676*	6695	6719	6730	6743	6754	6775*	6805*	6830	6831	6832	6834*	6843
STAC =	001100	6677*	6696	6720	6731	6744	6755	6776*	6806*	6831	6832	6833	6835*	6844
STAC =	001100	6678*	6697	6721	6732	6745	6756	6777*	6807*	6832	6833	6834	6836*	6845
STAC =	001100	6679*	6698	6722	6733	6746	6757	6778*	6808*	6833	6834	6835	6837*	6846
STAC =	001100	6680*	6699	6723	6734	6747	6758	6779*	6809*	6834	6835	6836	6838*	6847
STAC =	001100	6681*	6700	6724	6735	6748	6759	6780*	6810*	6835	6836	6837	6839*	6848
STAC =	001100	6682*	6701	6725	6736	6749	6760	6781*	6811*	6836	6837	6838	6840*	6849
STAC =	001100	6683*	6702	6726	6737	6750	6761	6782*	6812*	6837	6838	6839	6841*	6850
STAC =	001100	6684*	6703	6727	6738	6751	6762	6783*	6813*	6838	6839	6840	6842*	6851
STAC =	001100	6685*	6704	6728	6739	6752	6763	6784*	6814*	6839	6840	6841	6843*	6852
STAC =	001100	6686*	6705	6729	6740	6753	6764	6785*	6815*	6840	6841	6842	6844*	6853
STAC =	001100	6687*	6706	6730	6741	6754	6765	6786*	6816*	6841	6842	6843	6845*	6854
STAC =	001100	6688*	6707	6731	6742	6755	6766	6787*	6817*	6842	6843	6844	6846*	6855
STAC =	001100	6689*	6708	6732	6743	6756	6767	6788*	6818*	6843	6844	6845	6847*	6856
STAC =	001100	6690*	6709	6733	6744	6757	6768	6789*	6819*	6844	6845	6846	6848*	6857
STAC =	001100	6691*	6710	6734	6745	6758	6769	6790*	6820*	6845	6846	6847	6849*	6858
STAC =	001100	6692*	6711	6735	6746	6759	6770	6791*	6821*	6846	6847	6848	6850*	6859
STAC =	001100	6693*	6712	6736	6747	6760	6771	6792*	6822*	6847	6848	6849	6851*	6860
STAC =	001100	6694*	6713	6737	6748	6761	6772	6793*	6823*	6848	6849	6850	6852*	6861
STAC =	001100	6695*	6714	6738	6749	6762	6773	6794*	6824*	6849	6850	6851	6853*	6862
STAC =	001100	6696*	6715	6739	6750	6763	6774	6795*	6825*	6850	6851	6852	6854*	6863
STAC =	001100	6697*	6716	6740	6751	6764	6775	6796*	6826*	6851	6852	6853	6855*	6864
STAC =	001100	6698*	6717	6741	6752	6765	6776	6797*	6827*	6852	6853	6854	6856*	6865
STAC =</														

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 198
DZDMMB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

MAINDEC-11-DZDMM-B MACY11 27(732) 23-SEP-76 14:33 PAGE 189
DZDHMB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0306

SDDW12	001346	315#																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	</

SEQ 0307

		2760*	2840*	2918*	3047*	3194*	3322*	3439*	3511*	3568*	3676*	3769*	3884*	4003*
		4122*	4267*	4349*	4529*	4638*	4764*	4886*	5049*	5149*	5545*	5562*	5568*	5603*
		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	001256	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#	3949#	4055#	4232#	4315#	4463#
		4605#	4732#	4827#	5000#	5094#	5196#	5278#	5336#	5394#				
\$JCNT	022000	5698#	5727*	5740#										
\$JMODE	022002	5693*	5697*	5702	5705*	5716*	5742#							
\$OVER	021220	5522	5538	5546	5556	5565#								
\$PASS	001240	263#	709*	5477*	5478*	5486	5499	5552	5569					
\$PASTH	000222	79#												
\$POWER	024256	6244	6251#											
\$PWROD	024244	6246#												
\$PWROD	024104	685	6213#	6241										
\$PWARMG	024240	6244#												
\$PWRLP	024156	6223	6229#											
\$QUES	001226	251#	5619	5889	6014	6090	6107	6163	6166					
\$RDCHE	023242	6027#	6206											
\$RDDEC=	***** U	6209												
\$RDLIN	023362	6055#	6207											
\$RDOCT	023662	6127#	6208											
\$RDSZ =	000010	6048#												
\$REGAD	001160	231#												
\$REGD	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*	6289*	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*	6279*	6291*	6292*	6767*	6780*	7253	7276	7506			
\$REG7	001200	240#												
\$RTNAC	020724	5498#												
\$R2A =	***** U	6209												

K08

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0308

```

SSAVRE= ***** U
SSAVR6  024254
SSCOPE  020746
SSETUP= 000137

```

SSUP = 177777
SSVLAD 021164
SSVPC = 000214
SSWR = 165400

```

SSWREG      001254
SSWRMK=     000000
$TESTN     001236
$TIMES      001222
$TKB        001146
$TKS        001144
$TMPO       001202

```

STMP1 001204

STMP2 001206

STMP3 001210

STMP4 001212

STMP5 001214

STMP6 001216

STMP7 001220

SYN = 000061

STPB	001152
STPFLG	001157
STPS	001150

6209	6230	6231*	6232*	6250#								
6232*	5517*	6571										
6759	678	679	681	683	685	687	688	689	691	714	5475	5518
671*	5606	5614	5951	6113								
5583												
671*	5559*											
5530	59											
54*	25	30	31	32	33	34	35	249	250	251	688	689
8*	692	782	843	915	1002	1058	1165	1248	1332	1421	1499	1577
691	1798	1940	2012	2084	2172	2260	2344	2428	2541	2630	2720	2800
1657	2959	3148	3266	3378	3450	3523	3624	3741	3834	3952	4059	4236
2880	4467	4609	4736	4831	5004	5098	5200	5282	5340	5398	5470	5476
4319	5497	5499	5509	5510	5511	5512	5513	5521	5533	5535	5536	5539
5491	5541	5548	5549	5550	5562	5565	5568	5575	5576	5577	5578	5587
5540	5603	5607	5619	6247								
5591	712											
271*	36	5513	5514	5537								
35	5560*											
262*	688*	5476*	5548*	5555	5558*	5568						
249*	5949	5960	5977	6031	6037							
224*	5949	5958	5974	5998*	6029	6035						
223*	3234*	3235*	3250*	3251*	3696*	3697*	3698*	3699*	3700*	3789*	3790*	3791*
241*	3793*	4363*	4675*	4791*	5226*	5227*	5229	6306*	6307*	6766*	6779*	6868*
3792*	7276	7506	7552	7590	7691							
7253	2129*	2144	2147	2217*	2232	2235	2302*	2386*	2492*	2508	3236*	3252*
242*	4930	4951*	4960	4968*	4979	5175*	5181*	7691				
4919*	2146*	2148	2151	2159	2164*	2165	2234*	2236	2239	2247	2252*	2253
243*	2320	2323	2331	2336*	2337	2402*	2404	2407	2415	2420*	2421	2493*
2318*	4372*	4373*	4662*	4663*	4777*	4778*	7552					
2513	2135*	2136	2138*	2139	2223*	2224	2226*	2227	2307*	2308	2310*	2311
244*	2392	2394*	2395	2496*	2498	2501*	2502	4364*	4445*	4459*	4530*	4539*
2391*												
7590	2512*	2515	2520	2528	2533*	2534	4384*	4404*	4405	4409*	4410*	4421
245*	4801*	4802	6402*	6416*								
4784*	4359*	4696*	4702	6409								
246*	4358*	4551*	4577*	4799*	4816	4823	6408	6419	6820*	6834*		
247*	1754*	1793	1896*	1935	3224*	3225	3228	3236	3244	3252	3258*	3259
248*	3421	3442*	3443	3482*	3485	3495	3514*	3515	3571*	3573	3605	3609
3419*	3617*	3618	3679*	3685	3696	3720	3732*	3733	3772*	3779	3789	3912
3612	3825	3914*	3915*	4008*	4012	4041	4351*	4353*	4360*	4424	4550*	4593*
3824*	4708	4712	4713	6378*	6379	6382*	6383	6391*	6392	6394*	6395	
4646*	7											

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000

100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000

SR

SR

4948	4949	4950	4951	4952	4953	4954	4955	4956	4957	4958	4959	4960	4961	4962	4963	4964	4965	4966	4967	4968	4969	4970	4971	4972	4973	4974	4975	4976	4977	4978	4979	4980	4981	4982	4983	4984	4985	4986	4987	4988	4989	4990	4991	4992	4993	4994	4995	4996	4997	4998	4999	5000	5001	5002	5003	5004	5005	5006	5007	5008	5009	5010	5011	5012	5013	5014	5015	5016	5017	5018	5019	5020	5021	5022	5023	5024	5025	5026	5027	5028	5029	5030	5031	5032	5033	5034	5035	5036	5037	5038	5039	5040	5041	5042	5043	5044	5045	5046	5047	5048	5049	5050	5051	5052	5053	5054	5055	5056	5057	5058	5059	5060	5061	5062	5063	5064	5065	5066	5067	5068	5069	5070	5071	5072	5073	5074	5075	5076	5077	5078	5079	5080	5081	5082	5083	5084	5085	5086	5087	5088	5089	5090	5091	5092	5093	5094	5095	5096	5097	5098	5099	5100	5101	5102	5103	5104	5105	5106	5107	5108	5109	5110	5111	5112	5113	5114	5115	5116	5117	5118	5119	5120	5121	5122	5123	5124	5125	5126	5127	5128	5129	5130	5131	5132	5133	5134	5135	5136	5137	5138	5139	5140	5141	5142	5143	5144	5145	5146	5147	5148	5149	5150	5151	5152	5153	5154	5155	5156	5157	5158	5159	5160	5161	5162	5163	5164	5165	5166	5167	5168	5169	5170	5171	5172	5173	5174	5175	5176	5177	5178	5179	5180	5181	5182	5183	5184	5185	5186	5187	5188	5189	5190	5191	5192	5193	5194	5195	5196	5197	5198	5199	5200	5201	5202	5203	5204	5205	5206	5207	5208	5209	5210	5211	5212	5213	5214	5215	5216	5217	5218	5219	5220	5221	5222	5223	5224	5225	5226	5227	5228	5229	5230	5231	5232	5233	5234	5235	5236	5237	5238	5239	5240	5241	5242	5243	5244	5245	5246	5247	5248	5249	5250	5251	5252	5253	5254	5255	5256	5257	5258	5259	5260	5261	5262	5263	5264	5265	5266	5267	5268	5269	5270	5271	5272	5273	5274	5275	5276	5277	5278	5279	5280	5281	5282	5283	5284	5285	5286	5287	5288	5289	5290	5291	5292	5293	5294	5295	5296	5297	5298	5299	5300	5301	5302	5303	5304	5305	5306	5307	5308	5309	5310	5311	5312	5313	5314	5315	5316	5317	5318	5319	5320	5321	5322	5323	5324	5325	5326	5327	5328	5329	5330	5331	5332	5333	5334	5335	5336	5337	5338	5339	5340	5341	5342	5343	5344	5345	5346	5347	5348	5349	5350	5351	5352	5353	5354	5355	5356	5357	5358	5359	5360	5361	5362	5363	5364	5365	5366	5367	5368	5369	5370	5371	5372	5373	5374	5375	5376	5377	5378	5379	5380	5381	5382	5383	5384	5385	5386	5387	5388	5389	5390	5391	5392	5393	5394	5395	5396	5397	5398	5399	5400	5401	5402	5403	5404	5405	5406	5407	5408	5409	5410	5411	5412	5413	5414	5415	5416	5417	5418	5419	5420	5421	5422	5423	5424	5425	5426	5427	5428	5429	5430	5431	5432	5433	5434	5435	5436	5437	5438	5439	5440	5441	5442	5443	5444	5445	5446	5447	5448	5449	5450	5451	5452	5453	5454	5455	5456	5457	5458	5459	5460	5461	5462	5463	5464	5465	5466	5467	5468	5469	5470	5471	5472	5473	5474	5475	5476	5477	5478	5479	5480	5481	5482	5483	5484	5485	5486	5487	5488	5489	5490	5491	5492	5493	5494	5495	5496	5497	5498	5499	5500	5501	5502	5503	5504	5505	5506	5507	5508	5509	5510	5511	5512	5513	5514	5515	5516	5517	5518	5519	5520	5521	5522	5523	5524	5525	5526	5527	5528	5529	5530	5531	5532	5533	5534	5535	5536	5537	5538	5539	5540	5541	5542	5543	5544	5545	5546	5547	5548	5549	5550	5551	5552	5553	5554	5555	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566	5567	5568	5569	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581	5582	5583	5584	5585	5586	5587	5588	5589	5590	5591	5592	5593	5594	5595	5596	5597	5598	5599	5600	5601	5602	5603	5604	5605	5606	5607	5608	5609	5610	5611	5612	5613	5614	5615	5616	5617	5618	5619	5620	5621	5622	5623	5624	5625	5626	5627	5628	5629	5630	5631	5632	5633	5634	5635	5636	5637	5638	5639	5640	5641	5642	5643	5644	5645	5646	5647	5648	5649	5650	5651	5652	5653	5654	5655	5656	5657	5658	5659	5660	5661	5662	5663	5664	5665	5666	5667	5668	5669	5670	5671	5672	5673	5674	5675	5676	5677	5678	5679	5680	5681	5682	5683	5684	5685	5686	5687	5688	5689	5690	5691	5692	5693	5694	5695	5696	5697	5698	5699	5700	5701	5702	5703	5704	5705	5706	5707	5708	5709	5710	5711	5712	5713	5714	5715	5716	5717	5718	5719	5720	5721	5722	5723	5724	5725	5726	5727	5728	5729	5730	5731	5732	5733	5734	5735	5736	5737	5738	5739	5740	5741	5742	5743	5744	5745	5746	5747	5748	5749	5750	5751	5752	5753	5754	5755	5756	5757	5758	5759	5760	5761	5762	5763	5764	5765	5766	5767	5768	5769	5770	5771	5772	5773	5774	5775	5776	5777	5778	5779	5780	5781	5782	5783	5784	5785	5786	5787	5788	5789	5790	5791	5792	5793	5794	5795	5796	5797	5798	5799	5800	5801	5802	5803	5804	5805	5806	5807	5808	5809	5810	5811	5812	5813	5814	5815	5816	5817	5818	5819	5820	5821	5822	5823	5824	5825	5826	5827	5828	5829	5830	5831	5832	5833	5834	5835	5836	5837	5838	5839	5840	5841	5842	5843	5844	5845	5846	5847	5848	5849	5850	5851	5852	5853	5854	5855	5856	5857	5858	5859	5860	5861	5862	5863	5864	5865	5866	5867	5868	5869	5870	5871	5872	5873	5874	5875	5876	5877	5878	5879	5880	5881	5882	5883	5884	5885	5886	5887	5888	5889	5890	5891	5892	5893	5894	5895	5896	5897	5898	5899	5900	5901	5902	5903	5904	5905	5906	5907	5908	5909	5910	5911	5912	5913	5914	5915	5916	5917	5918	5919	5920	5921	5922	5923	5924	5925	5926	5927	5928	5929	5930	5931	5932	5933	5934	5935	5936	5937	5938	5939	5940	5941	5942	5943	5944	5945	5946	5947	5948	5949	5950	5951	5952	5953	5954	5955	5956	5957	5958	5959	5960	5961	5962	5963	5964	5965	5966	5967	5968	5969	5970	5971	5972	5973	5974	5975	5976	5977	5978	5979	5980	5981	5982	5983	5984	5985	5986	5987	5988	5989	5990	5991	5992	5993	5994	5995	5996	5997	5998	5999	6000	6001	6002	6003	6004	6005	6006	6007	6008	6009	6010	6011	6012	6013	6014	6015	6016	6017	6018	6019	6020	6021	6022	6023	6024	6025	6026	6027	6028	6029	6030	6031	6032	6033	6034	6035	6036	6037	6038	6039	6040	6041	6042	6043	6044	6045	6046	6047	6048	6049	6050	6051	6052	6053	6054	6055	6056	6057	6058	6059	6060	6061	6062	6063	6064	6065	6066	6067	6068	6069	6070	6071	6072	6073	6074	6075	6076	6077	6078	6079	6080	6081	6082	6083	6084	6085	6086	6087	6088	6089	6090	6091	6092	6093	6094	6095	6096	6097	6098	6099	6100	6101	6102	6103	6104	6105	6106	6107	6108	6109	6110	6111	6112	6113	6114	6115	6116	6117	6118	6119	6120	6121	6122	6123	6124	6125	6126	6127	6128	6129	6130	6131	6132	6133	6134	6135	6136	6137	6138	6139	6140	6141	6142	6143	6144	6145	6146	6147	6148	6149	6150	6151	6152	6153	6154	6155	6156	6157	6158	6159	6160	6161	6162	6163	6164	6165	6166	6167	6168	6169	6170	6171	6172	6173	6174	6175	6176	6177	6178	6179	6180	6181	6182	6183	6184	6185	6186	6187	6188	6189	6190	6191	6192	6193	6194	6195	6196	6197	6198	6199	6200	6201	6202	6203	6204	6205	6206	6207	6208	6209	6210	6211	6212	6213	6214	6215	6216	6217	6218	6219	6220	6221	6222	6223	6224	6225	6226	6227	6228	6229	6230	6231	6232	6233	6234	6235	6236	6237	6238	6239	6240	6241	6242	6243	6244	6245	6246	6247	6248	6249	6250	6251	6252	6253	6254	6255	6256	6257	6258	6259	6260	6261	6262	6263	6264	6265	6266	6267	6268	6269	6270	6271	6272	6273	6274	6275	6276	6277	6278	6279	6280	6281	6282	6283	6284	6285	6286	6287	6288	6289	6290	6291	6292	6293	6294</

4949	4950	4951	4952	4953	4954	4955	4956	4957	4958	4959	4960	4961	4962	4963	4964	4965	4966	4967	4968	4969	4970	4971	4972	4973	4974	4975	4976	4977	4978	4979	4980	4981	4982	4983	4984	4985	4986	4987	4988	4989	4990	4991	4992	4993	4994	4995	4996	4997	4998	4999	5000	5001	5002	5003	5004	5005	5006	5007	5008	5009	5010	5011	5012	5013	5014	5015	5016	5017	5018	5019	5020	5021	5022	5023	5024	5025	5026	5027	5028	5029	5030	5031	5032	5033	5034	5035	5036	5037	5038	5039	5040	5041	5042	5043	5044	5045	5046	5047	5048	5049	5050	5051	5052	5053	5054	5055	5056	5057	5058	5059	5060	5061	5062	5063	5064	5065	5066	5067	5068	5069	5070	5071	5072	5073	5074	5075	5076	5077	5078	5079	5080	5081	5082	5083	5084	5085	5086	5087	5088	5089	5090	5091	5092	5093	5094	5095	5096	5097	5098	5099	5100	5101	5102	5103	5104	5105	5106	5107	5108	5109	5110	5111	5112	5113	5114	5115	5116	5117	5118	5119	5120	5121	5122	5123	5124	5125	5126	5127	5128	5129	5130	5131	5132	5133	5134	5135	5136	5137	5138	5139	5140	5141	5142	5143	5144	5145	5146	5147	5148	5149	5150	5151	5152	5153	5154	5155	5156	5157	5158	5159	5160	5161	5162	5163	5164	5165	5166	5167	5168	5169	5170	5171	5172	5173	5174	5175	5176	5177	5178	5179	5180	5181	5182	5183	5184	5185	5186	5187	5188	5189	5190	5191	5192	5193	5194	5195	5196	5197	5198	5199	5200	5201	5202	5203	5204	5205	5206	5207	5208	5209	5210	5211	5212	5213	5214	5215	5216	5217	5218	5219	5220	5221	5222	5223	5224	5225	5226	5227	5228	5229	5230	5231	5232	5233	5234	5235	5236	5237	5238	5239	5240	5241	5242	5243	5244	5245	5246	5247	5248	5249	5250	5251	5252	5253	5254	5255	5256	5257	5258	5259	5260	5261	5262	5263	5264	5265	5266	5267	5268	5269	5270	5271	5272	5273	5274	5275	5276	5277	5278	5279	5280	5281	5282	5283	5284	5285	5286	5287	5288	5289	5290	5291	5292	5293	5294	5295	5296	5297	5298	5299	5300	5301	5302	5303	5304	5305	5306	5307	5308	5309	5310	5311	5312	5313	5314	5315	5316	5317	5318	5319	5320	5321	5322	5323	5324	5325	5326	5327	5328	5329	5330	5331	5332	5333	5334	5335	5336	5337	5338	5339	5340	5341	5342	5343	5344	5345	5346	5347	5348	5349	5350	5351	5352	5353	5354	5355	5356	5357	5358	5359	5360	5361	5362	5363	5364	5365	5366	5367	5368	5369	5370	5371	5372	5373	5374	5375	5376	5377	5378	5379	5380	5381	5382	5383	5384	5385	5386	5387	5388	5389	5390	5391	5392	5393	5394	5395	5396	5397	5398	5399	5400	5401	5402	5403	5404	5405	5406	5407	5408	5409	5410	5411	5412	5413	5414	5415	5416	5417	5418	5419	5420	5421	5422	5423	5424	5425	5426	5427	5428	5429	5430	5431	5432	5433	5434	5435	5436	5437	5438	5439	5440	5441	5442	5443	5444	5445	5446	5447	5448	5449	5450	5451	5452	5453	5454	5455	5456	5457	5458	5459	5460	5461	5462	5463	5464	5465	5466	5467	5468	5469	5470	5471	5472	5473	5474	5475	5476	5477	5478	5479	5480	5481	5482	5483	5484	5485	5486	5487	5488	5489	5490	5491	5492	5493	5494	5495	5496	5497	5498	5499	5500	5501	5502	5503	5504	5505	5506	5507	5508	5509	5510	5511	5512	5513	5514	5515	5516	5517	5518	5519	5520	5521	5522	5523	5524	5525	5526	5527	5528	5529	5530	5531	5532	5533	5534	5535	5536	5537	5538	5539	5540	5541	5542	5543	5544	5545	5546	5547	5548	5549	5550	5551	5552	5553	5554	5555	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566	5567	5568	5569	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581	5582	5583	5584	5585	5586	5587	5588	5589	5590	5591	5592	5593	5594	5595	5596	5597	5598	5599	5600	5601	5602	5603	5604	5605	5606	5607	5608	5609	5610	5611	5612	5613	5614	5615	5616	5617	5618	5619	5620	5621	5622	5623	5624	5625	5626	5627	5628	5629	5630	5631	5632	5633	5634	5635	5636	5637	5638	5639	5640	5641	5642	5643	5644	5645	5646	5647	5648	5649	5650	5651	5652	5653	5654	5655	5656	5657	5658	5659	5660	5661	5662	5663	5664	5665	5666	5667	5668	5669	5670	5671	5672	5673	5674	5675	5676	5677	5678	5679	5680	5681	5682	5683	5684	5685	5686	5687	5688	5689	5690	5691	5692	5693	5694	5695	5696	5697	5698	5699	5700	5701	5702	5703	5704	5705	5706	5707	5708	5709	5710	5711	5712	5713	5714	5715	5716	5717	5718	5719	5720	5721	5722	5723	5724	5725	5726	5727	5728	5729	5730	5731	5732	5733	5734	5735	5736	5737	5738	5739	5740	5741	5742	5743	5744	5745	5746	5747	5748	5749	5750	5751	5752	5753	5754	5755	5756	5757	5758	5759	5760	5761	5762	5763	5764	5765	5766	5767	5768	5769	5770	5771	5772	5773	5774	5775	5776	5777	5778	5779	5780	5781	5782	5783	5784	5785	5786	5787	5788	5789	5790	5791	5792	5793	5794	5795	5796	5797	5798	5799	5800	5801	5802	5803	5804	5805	5806	5807	5808	5809	5810	5811	5812	5813	5814	5815	5816	5817	5818	5819	5820	5821	5822	5823	5824	5825	5826	5827	5828	5829	5830	5831	5832	5833	5834	5835	5836	5837	5838	5839	5840	5841	5842	5843	5844	5845	5846	5847	5848	5849	5850	5851	5852	5853	5854	5855	5856	5857	5858	5859	5860	5861	5862	5863	5864	5865	5866	5867	5868	5869	5870	5871	5872	5873	5874	5875	5876	5877	5878	5879	5880	5881	5882	5883	5884	5885	5886	5887	5888	5889	5890	5891	5892	5893	5894	5895	5896	5897	5898	5899	5900	5901	5902	5903	5904	5905	5906	5907	5908	5909	5910	5911	5912	5913	5914	5915	5916	5917	5918	5919	5920	5921	5922	5923	5924	5925	5926	5927	5928	5929	5930	5931	5932	5933	5934	5935	5936	5937	5938	5939	5940	5941	5942	5943	5944	5945	5946	5947	5948	5949	5950	5951	5952	5953	5954	5955	5956	5957	5958	5959	5960	5961	5962	5963	5964	5965	5966	5967	5968	5969	5970	5971	5972	5973	5974	5975	5976	5977	5978	5979	5980	5981	5982	5983	5984	5985	5986	5987	5988	5989	5990	5991	5992	5993	5994	5995	5996	5997	5998	5999	6000	6001	6002	6003	6004	6005	6006	6007	6008	6009	6010	6011	6012	6013	6014	6015	6016	6017	6018	6019	6020	6021	6022	6023	6024	6025	6026	6027	6028	6029	6030	6031	6032	6033	6034	6035	6036	6037	6038	6039	6040	6041	6042	6043	6044	6045	6046	6047	6048	6049	6050	6051	6052	6053	6054	6055	6056	6057	6058	6059	6060	6061	6062	6063	6064	6065	6066	6067	6068	6069	6070	6071	6072	6073	6074	6075	6076	6077	6078	6079	6080	6081	6082	6083	6084	6085	6086	6087	6088	6089	6090	6091	6092	6093	6094	6095	6096	6097	6098	6099	6100	6101	6102	6103	6104	6105	6106	6107	6108	6109	6110	6111	6112	6113	6114	6115	6116	6117	6118	6119	6120	6121	6122	6123	6124	6125	6126	6127	6128	6129	6130	6131	6132	6133	6134	6135	6136	6137	6138	6139	6140	6141	6142	6143	6144	6145	6146	6147	6148	6149	6150	6151	6152	6153	6154	6155	6156	6157	6158	6159	6160	6161	6162	6163	6164	6165	6166	6167	6168	6169	6170	6171	6172	6173	6174	6175	6176	6177	6178	6179	6180	6181	6182	6183	6184	6185	6186	6187	6188	6189	6190	6191	6192	6193	6194	6195	6196	6197	6198	6199	6200	6201	6202	6203	6204	6205	6206	6207	6208	6209	6210	6211	6212	6213	6214	6215	6216	6217	6218	6219	6220	6221	6222	6223	6224	6225	6226	6227	6228	6229	6230	6231	6232	6233	6234	6235	6236	6237	6238	6239	6240	6241	6242	6243	6244	6245	6246	6247	6248	6249	6250	6251	6252	6253	6254	6255	6256	6257	6258	6259	6260	6261	6262	6263	6264	6265	6266	6267	6268	6269	6270	6271	6272	6273	6274	6275	6276	6277	6278	6279	6280	6281	6282	6283	6284	6285	6286	6287	6288	6289	6290	6291	6292	6293	6294	6295	6296	6297	6298	6299	6300	6301	6302	6303	6304	63

	42336	42559	4315	4317	4319	4355	4380	4463	4465	4467	4544	4595	4600	4605	4607
	4609	4651	4732	4734	4736	4733	4819	4827	4823	4831	4898	5000	5002	5004	5051
	5094	5096	5098	5147	5196	5198	5200	5238	5278	5280	5282	5298	5336	5338	5340
	5356	5394	5396	5398	5467	5469	5469	5470	5471	5472	5474	5430	5483	5485	5489
	5481	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
	6066	6107	6113	6115	6118	6124	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	233	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
	2149	2257	2258	2259	2260	2305	2341	2342	2343	2344	2389	2425	2426	2427	2428
	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363
	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378
	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393
	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408
	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2423
	2424	2425	2426	2427	2428	2429	2430	2431	2432	2433	2434	2435	2436	2437	2438
	2439	2440	2441	2442	2443	2444	2445	2446	2447	2448	2449	2450	2451	2452	2453
	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463	2464	2465	2466	2467	2468
	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479	2480	2481	2482	2483
	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495	2496	2497	2498
	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511	2512	2513
	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527	2528
	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558
	2559	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573
	2574	2575	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588
	2589	2590	2591	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603
	2604	2605	2606	2607	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618
	2619	2620	2621	2622	2623	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633
	2634	2635	2636	2637	2638	2639	2640	2641	2642	2643	2644	2645	2646	2647	2648
	2649	2650	2651	2652	2653	2654	2655	2656	2657	2658	2659	2660	2661	2662	2663
	2664	2665	2666	2667	2668	2669	2670	2671	2672	2673	2674	2675	2676	2677	2678
	2679	2680	2681	2682	2683	2684	2685	2686	2687	2688	2689	2690	2691	2692	2693
	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703	2704	2705	2706	2707	2708
	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719	2720	2721	2722	2723
	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735	2736	2737	2738
	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751	2752	2753
	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767	2768
	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798
	2799	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813
	2814	2815	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828
	2829	2830	2831	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843
	2844	2845	2846	2847	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858
	2859	2860	2861	2862	2863	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873
	2874	2875	2876	2877	2878	2879	2880	2881	2882	2883	2884	2885	2886	2887	2888
	2889	2890	2891	2892	2893	2894	2895	2896	2897	2898	2899	2900	2901	2902	2903
	2904	2905	2906	2907	2908	2909	2910	2911	2912	2913	2914	2915	2916	2917	2918
	2919	2920	2921	2922	2923	2924	2925	2926	2927	2928	2929	2930	2931	2932	2933
	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943	2944	2945	2946	2947	2948
	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959	2960	2961	2962	2963
	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975	2976	2977	2978
	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991	2992	2993
	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007	3008
	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038
	3039	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053
	3054	3055	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068
	3069	3070	3071	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083
	3084	3085	3086	3087	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098
	3099	3100	3101	3102	3103	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113
	3114	3115	3116	3117	3118	3119	3120	3121	3122	3123	3124	3125	3126	3127	3128
	3129	3130	3131	3132	3133	3134	3135	3136	3137	3138	3139	3140	3141	3142	3143
	3144	3145	3146	3147	3148	3149	3150	3151	3152	3153	3154	3155	3156	3157	3158
	3159	3160	3161	3162	3163	3164	3165	3166	3167	3168	3169	3170	3171	3172	3173
	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183	3184	3185	3186	3187	3188
	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199	3200	3201	3202	3203
	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215	3216	3217	3218
	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231	3232	3233
	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247	3248
	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278
	3279	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293
	3294	3295	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308
	3309	3310	3311	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323
	3324	3325	3326	3327	3328	3329	3330	3331	3332						

MAINTDEC-11-DZDHMB-8 MACY11 27(732) 23-SEP-76 14:33 PAGE 20-
 DZDHMB.F11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

SEQ C319

.LIST	1	6	35	42	194	231	233	234	235	236	237	238	239	240	241
	241	243	244	245	246	247	248	249	255	258	671	693	778	782	839
	911	915	998	1002	1054	1058	1161	1165	1244	1248	1328	1332	1417	1421	1421
	1499	1573	1577	1653	1657	1794	1799	1926	1940	2008	2012	2090	2094	2168	2168
	2256	2260	2340	2344	2424	2428	2537	2541	2626	2630	2716	2720	2796	2800	2800
	2808	2954	2958	3144	3148	3262	3266	3374	3378	3446	3450	3519	3523	3620	3620
	3624	3741	3830	3834	3948	3952	4055	4059	4232	4236	4315	4319	4463	4467	4467
	4605	4609	4732	4736	4827	4831	5000	5004	5098	5196	5200	5278	5282	5336	5336
	5340	5398	5475	5479	5491	5513	5614	6048	6189	6197	6198	6199	6200	6201	6202
.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	2875	2954
	3144	3262	3374	3446	3519	3620	3737	3830	3948	4055	4232	4315	4463	4605	4732
.REM	4827	5000	5094	5278	5336	5394	6254	7231							
	782	843	915	1002	1058	1165	1248	1332	1421	1499	1577	1657	1798	1940	2012
	2084	2172	2260	2344	2428	2541	2630	2720	2900	2980	2958	3148	3266	3379	3450
	3523	3624	3741	3834	3952	4059	4236	4319	4467	4609	4736	4831	5004	5098	5200
.REPT	5282	5340	5398												
.SEPTL	42	233	241	7092	7113	7133	7154								
	26	36	45	50	60	84	194	255	323	671	714	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	5465
	5503	5559	5619	5666	5743	5810	5899	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	249	250	251
	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)

Spooler runtime 47 Seconds, 1380 KCS, 1505 disk reads, 4 disk writes, 319 pages

~~CONFIDENTIAL~~ - SECURITY INFORMATION Date 14-Oct-76 18:01:17 Monitor IPC-9 8078 (183) ENDJOB

[illegible]