

DH11

DH11 MEMORY TEST
CZDHBC0

AH-FG21C-MC
1 OF 1 OCT 1985
COPYRIGHT© 1972-85

digital
MADE IN USA

The image displays a grid of 60 small plots, organized into 10 rows and 6 columns. Each plot contains a different visualization of data, likely from a memory test. The plots show various patterns, including vertical lines, horizontal lines, and complex waveforms. Some plots appear to be histograms or bar charts, while others show more intricate data structures. The overall appearance is that of a diagnostic or test results page from a computer system.

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

.REM †

IDENTIFICATION

PRODUCT CODE: AC-8448C-MC
PRODUCT NAME: CZDHBCO DH11 MEMORY TEST
DATE: 12-JUN-1985
MAINTAINER: NAC SOFTWARE ENGINEERING
AUTHOR: MICHAEL DAVIS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES

COPYRIGHT (C) 1972, 1976, 1985 BY DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.

1
2
3
4
5
6
7

1. ABSTRACT

THE DH11 MEMORY TEST IS A TEST OF THE BYTE COUNT AND
BUS ADDRESS MEMORIES OF THE DH11. EACH MEMORY IS
TESTED FOR ADDRESSABILITY AND DATA READ/WRITE
CAPABILITY

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

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 FAMILY STANDARD COMPUTER WITH 4KW OF MEMORY
ASR-33 TELETYPE OR EQUIVALENT
DH11 ASYNCHRONOUS MULTIPLEXER
DM11 MAINTENANCE CARD INSTALLED

2.2 STORAGE

THE PROGRAM LOADS INTO 4KW OF MEMORY

3. LOADING PROCEDURE

THE STANDART PROCEDURE FOR LOADING ABSOLUTE BINARY TAPES
IS TO BE USED

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

4.1.1 AFTER PROGRAM LOAD (INITIAL PROGRAM START)

ALL CONSOLE SWITCHES DOWN

4.1.2 TO MODIFY DEVICE VECTOR AND CONTROL REGISTER ADDRESSES
AFTER PROGRAM RESTART

SW00=1

4.1.3 TO START PROGRAM AT SELECTED TEST AFTER PROGRAM RESTART

SW01=1

4.2 STARTING ADDRESS

THE STARTING ADDRESS FOR ALL TESTS IS 000200

THE RESTART ADDRESS FOR ALL TESTS I 0002000

THE STARTING ADDRESS TO ENTER A SELECTED TEST IS 000200

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 INITIAL PROGRAM START

4.3.1.1 LOAD PROGRAM INTO MEMORY

4.3.1.2 LOAD ADDRESS 000200

4.3.1.3 CLEAR CONSOLE SWITCHES

4.3.1.4 PRESS START

4.3.1.5 THE PROGRAM WILL TYPE "DH11 MEMORY TEST"
AND WILL THEN TYPE "VECTOR ADDRESS-" AND WAIT FOR AN
INPUT FROM THE TELETYPE KEYBOARD.

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

4.3 (CONT'D)

4.3.1.6 TYPE IN THE ADDRESS OF THE RECEIVER INTERRUPT VECTOR FOR THE DH11 TO BE TESTED FOLLOWED BY <CARRIAGE RETURN>

NOTE: WORDS IN ANGLE BRACKETS, I.E. <CARRIAGE RETURN> MEAN THAT THE TELETYPE KEY WITH THE NAMED FUNCTION SHOULD BE STRUCK

IF AN INCORRECT ADDRESS IS ENTERED, THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE SECOND MESSAGE OF 4.3.1.5
4.3.1.7 THE PROGRAM WILL TYPE "CONTROL REGISTER ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.1.8 TYPE IN THE ADDRESS OF THE SYSTEM CONTROL REGISTER OF THE DH11 TO BE TESTED FOLLOWED BY <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED, THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.1.7
4.3.1.9 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT IS ABOUT TO START TESTING, AND THEN TESTING WILL BEGIN

4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN

4.3.2.1 PERFORM 4.3.1.2 TO 4.3.1.5

4.3.2.2 THE PROGRAM WILL TYPE "DH11 MEMORY TEST" AND WILL THEN CONTINUE AS DESCRIBED IN 4.3.1.9

4.3.3 PROGRAM RESTART WITH SW00=1

4.3.3.1 LOAD ADDRESS 000200

4.3.3.2 SET SW01=1

4.3.3.3 PRESS START

4.3.3.4 THE PROGRAM WILL PERFORM AS DESCRIBED IN 4.3.1.5 TO 4.3.1.9

4.3.4 PROGRAM RESTART WITH SW01=1

4.3.4.1 LOAD ADDRESS 000200

4.3.4.2 SET SW01=1

4.3.4.3 PRESS START

4.3.4.4 THE PROGRAM WILL TYPE "DH11 MEMORY TEST" AND WILL THEN TYPE "TEST PC-" AND WILL WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.4.5 TYPE IN THE ADDRESS OF THE TEST AT WHICH THE PROGRAM IS TO BE STARTED FOLLOWED BY <CARRIAGE RETURN>

4.3.4.6 THE PROGRAM WILL TYPE R TO INDICATE THAT IT HAS STARTED AND WILL START TESTING AT THE SELECTED TEST.

NOTE: CARE MUST BE TAKEN WHEN THIS FEATURE IS USED, SINCE THERE IS NO PROTECTION AGAINST SELECTING AN ADDRESS THAT IS IN THE MIDDLE OF A TEST

NOTE: IF IT IS DESIRED TO LOOP ON THE TEST THAT IS SELECTED SET SW14=1 BEFORE ENTERING THE TEST ADDRESS

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

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

SW15=1, HALT ON ERROR
 SW14=1, LOOP ON CURRENT TEST
 SW13=1, SUPPRESS ERROR TYPEOUT
 SW11=1, INHIBIT ITERATIONS
 SW10=1, ESCAPE TO NEXT TEST ON ERROR
 SW09=1, FREEZE VARIABLE PARAMETER IN CURRENT TEST
 SW01=1, START PROGRAM AT SELECTED TEST
 SW00=1, CHANGE PARAMETERS AT PROGRAM RESTART

5.2 SUBROUTINE ABSTRACTS

5.2.1 TRAPCATCHER (LOCATIONS 000000-000776)

THIS ROUTINE IS USED TO INTERCEPT UNEXPECTED INTERRUPTS AND TRAPS. THE AREA FROM 000000-000776 IS LOADED WITH THE FOLLOWING SEQUENCE

2
0
4
0
:
:
:
772
0
776
0

IF AN UNEXPECTED INTERRUPT OR TRAP OCCURS, THE PROGRAM WILL HALT WITH THE PC 2 GREATER THAN THE ADDRESS TO WHICH THE PROGRAM TRAPPED. THE PROCESSOR STACK MAY BE EXAMINED TO DETERMINE WHERE THE PROGRAM WAS WHEN THE TRAP OR INTERRUPT OCCURED.

5.2.2 START (PROGRAM INITIALIZATION)

THIS ROUTINE INITIALIZES ALL PROGRAM FLAGS AND COUNTERS, TYPES THE PROGRAM TITLE MESSAGE, AND INPUTS THE VECTOR AND CONTROL REGISTER ADDRESSES OF THE DH11 TO BE TESTED.

5.2.3 BEGIN (PROGRAM START AND RESTART)

THIS ROUTINE IS ENTERED IMMEDIATELY AFTER "START" AND EACH TIME A PROGRAM PASS HAS BEEN COMPLETED. THE ROUTINE SETS UP THE PROCESSOR STACK AND STATUS WORD AND THEN TRANSFERS CONTROL TO THE TEST AT WHICH TESTING WILL BEGIN. IF SW01=0 WHEN THIS ROUTINE IS ENTERED TESTING WILL START AT T1 (TEST 1). IF SW01=1 WHEN THIS ROUTINE IS ENTERED, TESTING WILL START AT THE PC ENTERED FROM THE TELETYPE KEYBOARD.

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

5.2.4 EOP (END OF PASS)

THIS ROUTINE IS ENTERED ONCE PER PASS AFTER ALL TESTS HAVE BEEN COMPLETED. THIS ROUTINE TYPES THE MAINDEC IDENTIFICATION CODE OF THE PROGRAM, CLEARS ERROR FLAGS AND UPDATES THE PASS COUNT. IF THE PROGRAM WAS LOADED UNDER ACT11 OR DDP, THE ROUTINE CHECKS FOR RETURN TO THE ACT11 OR DDP MONITOR. IF THE PROGRAM IS NOT UNDER MONITOR CONTROL, THE ROUTINE TRANSFERS TO BEGIN.

5.2.5 SCOPER (SCOPE LOOP AND ITERATION HANDLER)

THIS ROUTINE IS ENTERED EACH TIME A TEST IS COMPLETED. THE ROUTINE CHECKS FOR THE FOLLOWING UPON ENTRY
A) IF SW10=1, THE ROUTINE WILL TRANSFER TO THE NEXT TEST IN SEQUENCE, AFTER CLEARING ERROR FLAGS.
B) IF SW11=1, THE ROUTINE WILL TRANSFER TO THE NEXT TEST SEQUENCE, AFTER CLEARING ERROR FLAGS.
C) IF SW14=1, THE ROUTINE WILL LOOP ON THE CURRENT TEST REGARDLESS OF THE ITERATION COUNT.

IF NONE OF THE ABOVE IS TRUE, THE ROUTINE WILL ADD 1 TO THE COUNT OF TEST ITERATIONS, AND COMPARE THIS VALUE TO THE NUMBER OF ITERATIONS THAT SHOULD BE PERFORMED. IF THESE NUMBERS ARE EQUAL, THE ROUTINE WILL TRANSFER TO THE NEXT TEST IN SEQUENCE. IF THE NUMBERS ARE NOT EQUAL, THE TEST CURRENTLY IN PROGRESS WILL BE REPEATED.

5.2.6 SCOP1R (FREEZE ON CURRENT DATA)

THE CALL TO THIS ROUTINE FOLLOWS IMMEDIATELY AFTER THE CALL TO THE ERROR HANDLER IN THOSE TESTS THAT HAVE VARIABLE PARAMETERS. THIS ROUTINE IS ALWAYS ENTERED IN THOSE TESTS, WHETHER OR NOT AN ERROR OCCURS. IF SW09=1, THE ROUTINE WILL TRANSFER CONTROL BACK TO THE TEST AT A POINT WHICH WILL ALLOW REPEATING THE FUNCTION UNDER TEST CONTINUOUSLY WITH THE SAME DATA. IF THIS OPTION IS SELECTED, THE ROUTINE "SCOPER" IS NEVER ENTERED AND ITERATION COUNTS WILL NOT BE UPDATED.

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

5.2.7 ERRORS (ERROR HANDLER)

THIS ROUTINE IS ENTERED UPON ERROR DETECTION ONLY.
WITH ALL CONSOLE SWITCHES DOWN, THE ROUTINE PROCEEDS AS FOLLOWS:

- A) THE PC OF THE INSTRUCTION THAT CALLED THE ERROR HANDLER IS ACCESSED THRU THE STACK, AND THEN THE EMT INSTRUCTION ITSELF IS FETCHED. THE 8 LSB OF THE EMT INSTRUCTION ARE THE ERROR CODE. THIS CODE IS USED TO ACCESS A TABLE OF ERROR MESSAGES AND ERROR DATA STORAGE LOCATIONS.
- B) IF THE TEST THAT FAILED DID NOT FAIL PREVIOUSLY DURING THIS PASS, A COMPLETE ERROR REPORT IS MADE IF THE TEST THAT FAILED FAILED MOR THAT ONCE DURING THE CURRENT PASS, ONLY THE DATA RELATING TO THE FAILUER IS TYPED. IF SW13=1, NO ERROR TYPEOUT IS MADE.
- C) THE ROUTINE NOW CHECKS FOR HALT ON ERROR. IF SW15=1 THE PROGRAM WILL HALT WITH THE PC OF THE CALL TO THE ERROR ROUTINE IN R0. IF SW15=0, THE PROGRAM WILL NOT HALT, BUT WILL CHECK FOR ESCAPE TO NEXT TEST.
- D) IF SW10=0, THE ROUTINE WILL RETURN TO THE TEST IN PROGRESS. IF SW10=1, THE ROUTINE WILL ABORT THE CURRENT TEST, AND TRANSFER TO THE NEXT TEST IN SEQUENCE, THRU THE ROUTINE "SCOPER".

5.2.8 TRPSRV (TRAP DECODE AND DISPATCH)

THIS ROUTINE DECODES THE 8 LSB OF THE TRAP INSTRUCTION THAT CAUSED TH PROGRAM INTERRUPT, AND TRANSFERS CONTROL TO THE ROUTINE THRU THE TABLE "TRPTAB" USING THE 8 LSB OF THE TRAP INSTRUCTION AS AN OFFSET TO THE POINTER TO THE ROUTINE TO BE ENTERED.

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

5.3 PROGRAM AND OR OPERATOR ACTION

5.3.1 PROGRAM START WITH ALL SWITCHES DOWN

5.3.1.1 REFER TO SECTIONS 4.3.1 AND 4.3.2 FOR INITIAL PROGRAM BEHAVIOR.

5.3.1.2 AFTER "R" HAS BEEN TYPED BY THE PROGRAM, TEST EXECUTION WILL BEGIN. EACH TEST WILL BE REPEATED A SELECTED NUMBER OF ITERATIONS (SEE LISTING FOR EXACT NUMBER FOR EACH TEST) AND THEN THE PROGRAM WILL PROCEED TO THE NEXT TEST.

5.3.1.3 WHEN ALL ITERATIONS HAVE BEEN COMPLETED, THE PROGRAM WILL TYPE "CZDHB-C-C" AND THEN RESTART TESTING AT TEST 1 (LOCATION T1 IN THE PROGRAM).

5.3.1.4 IF AN ERROR OCCURS, THE PROGRAM WILL TYPE AN APPROPRIATE ERROR MESSAGE, AND THEN CONTINUE THE TEST IN PROGRESS.

5.3.2 PROGRAM START WITH SW00=1

THE PROGRAM WILL PERFORM AS DESCRIBED IN 4.3.1 AND 5.3.1

5.3.3 PROGRAM START WITH SW01=1

5.3.3.1 REFER TO SECTION 4.3.4 FOR INITIAL PROGRAM BEHAVIOR

5.3.3.2 TEST EXECUTION WILL START AT THE ADDRESS SPECIFIED AND WILL CONTINUE AS DESCRIBED IN 5.3.1.2

5.3.3.3 AFTER "CZDHB-C" HAS BEEN TYPED, THE PROGRAM WILL RESUME TESTING AT TEST 1

5.3.4 PROGRAM OPERATION WITH SW15=1

SAME AS 5.3.1, EXCEPT THAT IN THE CASE OF AN ERROR, THE PROGRAM WILL HALT AFTER THE ERROR TYPEOUT, AND THE PC+2 OF THE CALL TO THE ERROR ROUTINE WILL BE DISPLAYED IN RO.

5.3.5 PROGRAM OPERATION WITH SW13=1

SAME AS 5.3.1 EXCEPT THAT NO ERROR TYPEOUTS WILL OCCUR

5.3.6 PROGRAM OPERATION WITH SW11=1

SAME AS 5.3.1 EXCEPT THAT EACH TEST WILL BE REPEATED ONCE ONLY

5.3.7 PROGRAM OPERATION WITH SW10=1

SAME AS 5.3.1, EXCEPT THAT IN THE CASE OF AN ERROR THE CURRENT TEST WILL BE ABORTED, AND THE PROGRAM WILL PROCEED TO THE NEXT TEST IN SEQUENCE.

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

5. (CONT'D)

5.3.8 PROGRAM OPERATION WITH SW14=1, OR SW09=1

THESE FUNCTIONS ARE NORMALLY USED FOR TROUBLE SHOOTING.
SEE SECTION 6.3 FOR THEIR USE.

6. ERRORS

6.1 ERROR HALTS

THE ERROR MESSAGE FORMAT FOR ALL ERROR TYPEOUTS
IS AS FOLLOWS

PC+2 MESSAGE
HEADER (IF APPLICABLE)
DATA (IF APPLICABLE)

WHERE

PC+2 IS THE ADDRESS OF THE CALL TO THE ERROR HANDLER + 2
MESSAGE IS AN ASCII MESSAGE DESCRIBING (BRIEFLY) THE FAILURE
HEADER IS A DESCRIPTION OF THE DATA TO FOLLOW
DATA IS OCTAL INFORMATION RELATING TO THE CAUSE OF THE FAILURE
IF THE SAME ERROR OCCURS IN A GIVEN TEST ON THE SAME
PASS, AND IF DATA IS ASSOCIATED WITH THAT ERROR, ONLY
DATA IS TYPED ON SUCCEEDING ERROR TYPEOUTS

IF NO DATA IS ASSOCIATED WITH THE ERROR
THE COMPLETE ERROR MESSAGE IS TYPED.

6.1.1 ERROR DESCRIPTIONS

SEE LISTING FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

6.2.1 SW15=0

IF THE PROGRAM IS RUN WITH SW15=0, NO OPERATOR ACTION IS
REQUIRED TO CONTINUE TESTING

6.2.2 SW15=1

IF THE PROGRAM IS RUN WITH SW15=1, TO CONTINUE TESTING
AFTER THE PROGRAM HAS HALTED, PRESS THE PROCESSOR
CONSOLE CONTINUE SWITCH

6.3 SCOPE LOOPING

6.3.1 TO SCOPE ON A SPECIFIC TEST, SET SW14=1 AND SW13=1
THIS WILL CAUSE THE PROGRAM TO CONTINUOUSLY LOOP ON THE
SAME TEST, AND WILL CAUSE ALL ERROR TYPEOUTS TO BE INHIBITED

6.3.2 TO SCOPE ON A SPECIFIC VALUE OF A PARAMETER WITHIN
A TEST, SET SW09=1 TO FREEZE THE DATA
(SEE LISTING FOR THOSE TESTS THAT INCORPORATE THIS FEATURE)

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

6. (CONT'D)
6.3.3 PROGRAM START TO SCOPE LOOP ON SELECTED TEST
PERFORM SECTION 4.3.4 WITH SW14=1

7. RESTRICTIONS
7.1 STARTING
THE DH11 TEST CARD MUST BE INSTALLED

7.2 RUNNING
NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME
THE TIME FOR ONE PASS OF THE PROGRAM (END OF
TYPEOUT OF CZDHB-C TO END OF TYPEOUT OF CZDHB-C)
IS GIVEN FOR VARIOUS PROCESSORS IN THE TABLE BELOW

PROCESSOR	TIME
PDP-11/04	05
	10
	15
	20
	24
	34
	35
	40
	44
	45
	55
	60
	70
	74
	84

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

9. PROGRAM DESCRIPTION

THE PROGRAM FIRST TESTS THE BUS ADDRESS AND BYTE COUNT MEMORIES FOR ADDRESSABILITY. THE TEST IS PERFORMED IN THE FOLLOWING MANNER:

A) EACH LOCATION OF THE MEMORY TO BE TESTED IS LOADED WITH ITS ADDRESS, DUPLICATED EVERY 4 BITS. THE BINARY CONTENTS OF EACH LOCATION IS SHOWN BELOW

LOCATION	CONTENTS
00	0000 0000 0000 0000
01	0001 0001 0001 0001
02	0010 0010 0010 0010
03	0011 0011 0011 0011
...
16	1110 1110 1110 1110
17	1111 1111 1111 1111

THE ABOVE PATTERN WAS CHOSEN SINCE THE MEMORY IS COMPOSED OF FOUR (4) CHIPS EACH HAVING A CAPACITY OF 16 WORDS BY FOUR (4) BITS. IF ANY OF THE FOUR CHIPS IS ADDRESSED INCORRECTLY, THE CONTENTS OF THAT CHIP WILL BE INCORRECT AND WILL INDICATION WHAT LOCATION WAS ACTUALLY ADDRESSED.

AFTER THE ABOVE TESTS HAVE BEEN COMPLETED, EACH LOCATION IN BOTH THE BUS ADDRESS AND BYTE COUNT MEMORIES ARE TESTED TO VERIFY THAT ALL BITS CAN BE SET TO 1S AND CLEARED TO 0S

THE NEXT GROUP OF TESTS VERIFY THAT A SELECTED ADDRESS IN EITHER THE BYTE COUNT OR BUS ADDRESS MEMORY CAN BE SET TO A SELECTED VALUE WITHOUT CHANGING THE CONTENTS OF ANY OTHER LOCATION IN THAT MEMORY.

THE NEXT GROUP OF TESTS SETS ALL LOCATIONS IN EITHER THE BYTE COUNT OR BUS ADDRESS MEMORY TO 1S, CLEARS A SELECTED LOCATION TO 0S, AND VERIFY THAT ONLY THE SELECTED LOCATION WAS AFFECTED.

THE FINAL GROUP OF TESTS VERIFIES THAT THE MEMORY EXTENTION BITS OF THE BUS ADDRESS MEMORY CAN BE SET AND CLEARED.

10. LISTING

†

1 ; DHMAC-A - DH11 MACRO LIBRARY
 2 ; COPYRIGHT 1985, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
 3
 4
 5
 6
 7

.LIST ME
 .NLIST MC,MD,CND

104
 119
 131
 148
 158
 167
 303
 339
 373
 520
 563
 595
 607
 652
 664
 691
 712
 743
 744
 745
 746
 747
 748

TRAPS."

749
 750
 CLEAN END OF PASS
 MESSAGE.

751
 752
 753
 TCH REGISTER"
 754

; CMS REPLACEMENT HISTORY

; *9 SKONETSKI 26-APR-1985 16:23:08 "FIXED TYPO CAUSING ASSEMBLY ERRORS"
 ; *8 SKONETSKI 22-APR-1985 16:48:03 "TYPO ERROR IN VECTOR CHANGE CODE SOURCE FIXED"
 ; *7 SKONETSKI 22-APR-1985 16:26:04 "ADDED CODE TO SET VECTORS FOR PWR FAIL, ERRORS, AND EMT
 ; *6 SKONETSKI 22-APR-1985 14:22:35 "FIXED BRANCH ERROR IN END OF PASS ROUTINE"
 ; *5 SKONETSKI 22-APR-1985 08:28:54 "FIXED BUG (AN OCTASC MACRO CALL WAS WRONG) AND ADDED A
 ; *4 SKONETSKI 18-APR-1985 14:20:15 "ADDED SOFTWARE SWITCH REG SUPPORT, BUT UNTESTED"
 ; *3 SKONETSKI 12-APR-1985 10:34:52 "FIXED PROBLEMS WITH SPURIOUS CR/LFS"
 ; *2 SKONETSKI 11-APR-1985 16:00:24 "ADDED MACRO FROM SYSMAC.SML THAT SIZES FOR SOFTWARE SWI
 ; *1 SKONETSKI 11-APR-1985 15:49:05 "LIBRARY FOR DH11 DIAGNOSTICS"

; 3

2
3
5 000000

.LIST ME
.NLIST MC,MD,CND
.HEADER †/1976,1985/,†/DH11 MEMORY TEST/,†/CZDHB-CO/

;STARTING PROCEDURE
;LOAD PROGRAM
;LOAD ADDRESS 000200
;PRESS START
;PROGRAM WILL TYPE DH11 MEMORY TEST
;PROGRAM WILL TYPE "VECTOR ADDRESS-"
;TYPE IN THE ADDRESS OF THE RECEIVER INTERRUPT VECTOR
;FOR THE DH11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
;PROGRAM WILL TYPE "CONTROL REGISTER ADDRESS-"
;TYPE IN THE ADDRESS OF THE SYSTEM CONTROL REGISTER
;FOR THE DH11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
;AT THE END OF A PASS, PROGRAM WILL TYPE " CZDHB-CO "
;AND THEN RESUM TESTING

; 3

000000

.TITLE CZDHB-CO
.ENABLE ABS
.NLIST MC,MD,CND
.LIST ME
.SYMBOLS

6 000000

;SWITCH REGISTER OPTIONS

100000
040000
020000
010000
004000
002000
001000
000400
000100
000040
000020
000010
000004
000002
000001

SW15=100000 ;-1,HALT ON ERROR
SW14=40000 ;-1,LOOP ON CURRENT TEST
SW13=20000 ;-1,INHIBIT ERROR TIMEOUT
SW12=10000
SW11=4000 ;-1,INHIBIT ITERATIONS
SW10=2000 ;-1,ESCAPE TO NEXT TEST ON ERROR
SW09=1000 ;-1,LOOP WITH CURRENT DATA
SW08=400
SW06=100
SW05=40
SW04=20
SW03=10
SW02=4
SW01=2
SW00=1

; 3

;RESTART PROGRAM AT SELECTED TEST
;RESELECT VECTOR AND CONTROL REGISTER
;ADDRESS AFTER PROGRAM RESTART

0

;REGISTER DEFINITIONS

```

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

```

;LOCATION EQUIVALENCIES

```

;SWR=177570 ;CONSOLE SWITCH REGISTER ; 3
;LIGHTS=177570 ;PDP-11/45 DISPLAY REGISTER ; 4
177776      PS=177776 ;PROCESSOR STATUS WORD ; 4
012774      STACK=ENDCOD+200 ;START OF PROCESSOR STACK ; 3

```

;INSTRUCTION DEFINITIONS

```

005746      PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD
005726      POP1SP=5726 ;INCREMENT PROCESSOR STACK 1 WORD
010046      PUSHRO=10046 ;SAVE R0 ON STACK
012600      POPRO=12600 ;RESTORE R0 FROM STACK
024646      PUSH2SP=24646 ;DECREMENT STACK TWICE
022626      POP2SP=22626 ;INCREMENT STACK TWICE

```

```

;
; .MACRO HLT ;A
; EMT ;A
; .ENDM HLT
;
;

```

```

100000      BIT15=100000
040000      BIT14=40000 ; 3
020000      BIT13=20000
010000      BIT12=10000
004000      BIT11=4000
002000      BIT10=2000
001000      BIT09=1000
000400      BIT08=400
000200      BIT07=200
000100      BIT06=100
000040      BIT05=40
000020      BIT04=20
000010      BIT03=10
000004      BIT02=4
000002      BIT01=2
000001      BIT00=1
1 000000    .CATCH

```


000146	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000150	000152	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000152	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000154	000156	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000156	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000160	000162	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000162	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000164	000166	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000166	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000170	000172	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000172	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000174	000176	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000176	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000200	000202	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000202	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000204	000206	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000206	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000210	000212	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000212	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000214	000216	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000216	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000220	000222	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000222	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000224	000226	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000226	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000230	000232	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000232	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000234	000236	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000236	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000240	000242	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000242	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000244	000246	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000246	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000250	000252	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000252	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000254	000256	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000256	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000260	000262	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000262	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000264	000266	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000266	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000270	000272	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000272	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000274	000276	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000276	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000300	000302	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000302	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000304	000306	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000306	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000310	000312	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000312	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000314	000316	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000316	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000320	000322	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000322	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000324	000326	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000326	000000	HALT	;EXAMINE STACK TO FIND CAUSE

000330	000332	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000332	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000334	000336	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000336	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000340	000342	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000342	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000344	000346	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000346	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000350	000352	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000352	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000354	000356	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000356	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000360	000362	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000362	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000364	000366	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000366	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000370	000372	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000372	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000374	000376	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000376	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000400	000402	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000402	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000404	000406	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000406	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000410	000412	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000412	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000414	000416	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000416	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000420	000422	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000422	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000424	000426	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000426	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000430	000432	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000432	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000434	000436	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000436	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000440	000442	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000442	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000444	000446	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000446	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000450	000452	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000452	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000454	000456	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000456	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000460	000462	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000462	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000464	000466	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000466	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000470	000472	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000472	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000474	000476	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000476	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000500	000502	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000502	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000504	000506	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000506	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000510	000512	.+2	;UNEXPECTED TRAP TO THIS LOCATION

000512	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000514	000516	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000516	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000520	000522	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000522	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000524	000526	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000526	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000530	000532	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000532	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000534	000536	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000536	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000540	000542	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000542	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000544	000546	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000546	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000550	000552	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000552	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000554	000556	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000556	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000560	000562	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000562	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000564	000566	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000566	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000570	000572	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000572	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000574	000576	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000576	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000600	000602	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000602	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000604	000606	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000606	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000610	000612	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000612	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000614	000616	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000616	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000620	000622	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000622	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000624	000626	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000626	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000630	000632	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000632	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000634	000636	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000636	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000640	000642	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000642	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000644	000646	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000646	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000650	000652	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000652	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000654	000656	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000656	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000660	000662	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000662	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000664	000666	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000666	000000	HALT	;EXAMINE STACK TO FIND CAUSE
000670	000672	.+2	;UNEXPECTED TRAP TO THIS LOCATION
000672	000000	HALT	;EXAMINE STACK TO FIND CAUSE

```
000674 000676      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000676 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000700 000702      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000702 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000704 000706      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000706 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000710 000712      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000712 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000714 000716      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000716 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000720 000722      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000722 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000724 000726      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000726 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000730 000732      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000732 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000734 000736      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000736 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000740 000742      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000742 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000744 000746      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000746 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000750 000752      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000752 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000754 000756      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000756 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000760 000762      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000762 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000764 000766      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000766 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000770 000772      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000772 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
000774 000776      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
000776 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
1 001000      .SETVEC
```

```

0          000200      000167 000600      .-200      ;STANDARD INTERRUPT VECTORS
          000200      JMP      START          ;GO TO START OF PROGRAM

1 000204          .TRPDEF

          ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
          ;POINTERS TO SUBROUTINES CAN BE FOUND STARTING
          ;AT LOCATION "TRPTAB"

000204          TRPDEF SCOPE,+/SCOPE LOOP AND ITERATION HANDLER/
          104400      SCOPE=TRAP+Y          ;SCOPE LOOP AND ITERATION HANDLER
          000001      Y=Y+1

000204          TRPDEF TYPE,+/TELETYPE OUTPUT ROUTINE/
          104401      TYPE=TRAP+Y          ;TELETYPE OUTPUT ROUTINE
          000002      Y=Y+1

000204          TRPDEF OCTASC,+/OCTAL TO ASCII CONVERSION/
          104402      OCTASC=TRAP+Y        ;OCTAL TO ASCII CONVERSION
          000003      Y=Y+1

000204          TRPDEF INSTR,+/INPUT ASCII STRING/
          104403      INSTR=TRAP+Y        ;INPUT ASCII STRING
          000004      Y=Y+1

000204          TRPDEF INSTER,+/STRING INPUT ERROR/
          104404      INSTER=TRAP+Y       ;STRING INPUT ERROR
          000005      Y=Y+1

000204          TRPDEF PARAM,+/CONVERT STRING TO OCTAL, CHECK LIMITS/
          104405      PARAM=TRAP+Y        ;CONVERT STRING TO OCTAL, CHECK LIMITS
          000006      Y=Y+1

000204          TRPDEF SAVOSP,+/SAVE R0-R5, PC/
          104406      SAVOSP=TRAP+Y       ;SAVE R0-R5, PC
          000007      Y=Y+1

000204          TRPDEF RESO5,+/RESTORE R0-R5/
          104407      RESO5=TRAP+Y        ;RESTORE R0-R5
          000010      Y=Y+1

000204          TRPDEF SCOPE1,+/CHECK FOR FREEZE ON CURRENT DATA/
          104410      SCOPE1=TRAP+Y       ;CHECK FOR FREEZE ON CURRENT DATA
          000011      Y=Y+1

2          .MACRO CODEM1
3          MOV      DHSSR,DHSLR          ;SET UP ADDRESS OF SILO
4          INC      DHSLR                ;STATUS REGISTER HIGH BYTE
5          .ENDM CODEM1
6 000204          .START DHRVEC,3,4,DHSCR,0,177776,7,10,...1

```



```

.ENDC
.IF NB <>
TRACER: MOV #1$, @#10 ;SET UP ILLEGAL INSTRUCTION TRAP RETURN
SXT R0 ;DO 11/40, 11/45 INSTRUCTION
MOV #RTT, TRTRET ;11/40,45 RTT RETURN FROM TRACE TRAP
BR 2$
1$: MOV #RTI, TRTRET ;1105,10,20 RTI RETURN FROM TRACE TRAP
MOV #12, @#10 ;RESTORE TRAPCATCHER
MOV #TRTRET, @#16 ;SET UP TRACE TRAP VECTOR

.ENDC
.IF NB <DHRVEC>
.IF B <> ; 3
BR VEC2
.IFF
TST INIFLG ;IF INITIALIZE FLAG=0
BEQ VEC2 ;GET VECTOR AND CSR ADDRESS

.ENDC
VEC1: BIT #SW00, @SWR ;IF SW00=1, GET NEW VECTOR ; 4
BEQ BEGIN ;AND CSR ; 4
VEC2: MOV #300, R1 ; 4
MOV #302, R2 ; 4
MOV #4, R3
1$: MOV R2, (R1) ;RESTORE TRAPCATCHER
CLR (R2) ;IN FLOATING VECTOR AREA
ADD R3, R1
ADD R3, R2
CMP R1, #1000
BNE 1$
INSTR ;INPUT ADDRESS OF DEVICE VECTOR
MVECTOR ;MESSAGE "VECTOR ADDRESS-"
PARAM ;CONVERT STRING TO OCTAL
300 ;LOW LIMIT
770 ;HIGH LIMIT
DHRVEC ;LOCATIONS TO BE FILLED ; 3
3 ;NUMBER OF LOCATIONS
4 ;LSB MASK
INSTR ;INPUT ADDRESS OF DEVICE CSR
MREGAD ;MESSAGE "CONTROL REGISTER ADDRESS-"
PARAM ;CONVERT STRING TO OCTAL
0 ;LOW LIMIT
177776 ;HIGH LIMIT
DHSCR ;LOCATIONS TO BE FILLED
7 ;NUMBER OF LOCATIONS
10 ;LSB MASK

.ENDC
.IF NB <1>
CODEM1
MOV DHSSR, DHSLR ;SET UP ADDRESS OF SILO
INC DHSLR ;STATUS REGISTER HIGH BYTE

.ENDC
TST INIFLG ;IF INITIALIZATION FLAG
BNE BEGIN ;IS CLEARED
COM INIFLG ;SET IT

;PROGRAM START
;CHECK FOR PROGRAM START AT SELECTED ADDRESS ; 3

```

```

001306 012767 000340 176462 BEGIN: MOV #340,PS ;LOCK OUT INTERRUPTS
001314 012706 012774 MOV #STACK,SP ;SET UP PROCESSOR STACK
001320 032777 000002 177452 BIT #SW01,@SWR ;IF SW01=1 ; 4
001326 001410 BEQ 1$ ;GET PC FOR PROGRAM START
001330 104403 INSTR ;GET PC
001332 012044 MTSTPC ;MESSAGE "TEST PC"
001334 104405 PARAM ;CONVERT STRING TO OCTAL
001336 000000 0
001340 017500 17500
001342 011374 RETRN
001344 001 .BYTE 1
001345 001 .BYTE 1
001346 000410 BR 2$
001350 012767 001400 010016 1$: MOV #T1,RETRN ;NORMAL START, TEST 1
001356 005767 010046 TST STFLG ;IF LOOPING, BYPASS TYPEOUT
001362 001004 BNE 3$
001364 005167 010040 COM STFLG
001370 104401 012040 2$: TYPE ,MR ;TYPE "R" TO INDICATE START
001374 000177 007774 3$: JMP @RETRN ;START TESTING ; 3

```

```

2 001400          MEMENT1  †/BA/,†/BUS ADDRESS/

;BUS ADDRESS MEMORY ADDRESSING TEST
;LOAD EACH LOCATION IN THE BUS ADDRESS MEMORY
;WITH THE ADDRESS OF THAT LOCATION.
;THE ADDRESS IS REPEATED EVERY 4 BITS
;VERIFY THAT EACH LOCATION IN THE BUS ADDRESS MEMORY
;WAS ADDRESSED.

001400          TS \XN,100,4‡
001400 012767 000340 176370 T1:  MOV    †340,PS          ;DISABLE ALL INTERRUPTS
001406 012767 000100 007766      MOV    †100,ICOUNT      ;SET UP FOR 100 ITERATIONS
001414 012767 001514 007754      MOV    †4‡,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

          .IF NB  <>
          MOV    †,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
          .ENDC
          XN=XN+1
001422 000002 012700 000020      MOV    †20,R0          ;SET UP TO ADDRESS 20 (OCTAL)
;LOCATIONS IN THE BUS ADDRESS MEMORY
;START AT ADDRESS 0
001426 005004          CLR    R4
001430 005002          CLR    R2
001432 010477 007676      1‡:  MOV    R4,‡DHSCR      ;SELECT ADDRESS IN BUS ADDRESS
;MEMORY TO BE ADDRESSED
;LOAD MEMORY LOCATION
;WITH ITS ADDRESS
;ADVANCE TO NEXT ADDRESS
001436 010277 007700      MOV    R2,‡DHBA
001442 062702 010421      ADD    †10421,R2
001446 005204          INC    R4
001450 005300          DEC    R0
001452 001367          BNE   1‡
001454 012700 000020      MOV    †20,R0          ;CONTINUE IF NOT DONE
;SET UP TO CHECK
;EACH MEMORY ADDRESS
;START AT ADDRESS 0
001460 005004          CLR    R4
001462 005002          CLR    R2
001464 010477 007644      2‡:  MOV    R4,‡DHSCR      ;ADDRESS MEMORY LOCATION
;READ CONTENTS OF MEMORY
;WAS MEMORY LOCATION LOADED
;WITH ITS ADDRESS
001470 017703 007646      MOV    ‡DHBA,R3
001474 020203          CMP    R2,R3
001476 001401          BEQ   3‡
001500          .IIF IDN <BA>,<BA>,      HLT    1          ;BUS ADDRESS MEMORY ERROR
001500 104001          EMT    1
001502 005204          .IIF IDN <BA>,<BC>,      HLT    2          ;BYTE COUNT MEMORY ERROR
001504 062702 010421      3‡:  INC    R4          ;ADVANCE TO NEXT LOCATION
001510 005300          ADD    †10421,R2
001512 001364          DEC    R0
001514 104400          BNE   2‡          ;CONTINUE IF NOT DONE
3 001516          4‡:  SCOPE
MEMENT1 †/BC/,†/BYTE COUNT/

;BYTE COUNT MEMORY ADDRESSING TEST
;LOAD EACH LOCATION IN THE BYTE COUNT MEMORY
;WITH THE ADDRESS OF THAT LOCATION.
;THE ADDRESS IS REPEATED EVERY 4 BITS
;VERIFY THAT EACH LOCATION IN THE BYTE COUNT MEMORY
;WAS ADDRESSED.

001516          TS \XN,100,4‡
001516 012767 000340 176252 T2:  MOV    †340,PS          ;DISABLE ALL INTERRUPTS

```

```

001524 012767 000100 007650      MOV      #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
001532 012767 001632 007636      MOV      #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
001540 012700 000020      MOV      #20,R0          ;SET UP TO ADDRESS 20 (OCTAL)
                                ;LOCATIONS IN THE BYTE COUNT MEMORY
001544 005004      CLR      R4              ;START AT ADDRESS 0
001546 005002      CLR      R2
001550 010477 007560      1$:  MOV      R4,@DHSCR      ;SELECT ADDRESS IN BYTE COUNT
                                ;MEMORY TO BE ADDRESSED
                                MOV      R2,@DHBC      ;LOAD MEMORY LOCATION
                                ADD      #10421,R2      ;WITH ITS ADDRESS
                                INC      R4              ;ADVANCE TO NEXT ADDRESS
                                DEC      R0
001554 010277 007564      MOV      R2,@DHBC
001560 062702 010421      ADD      #10421,R2
001564 005204      INC      R4
001566 005300      DEC      R0
001570 001367      BNE      1$
001572 012700 000020      MOV      #20,R0          ;CONTINUE IF NOT DONE
                                ;SET UP TO CHECK
                                CLR      R4              ;EACH MEMORY ADDRESS
                                CLR      R2              ;START AT ADDRESS 0
001576 005004      CLR      R4
001600 005002      CLR      R2
001602 010477 007526      2$:  MOV      R4,@DHSCR      ;ADDRESS MEMORY LOCATION
001606 017703 007532      MOV      @DHBC,R3        ;READ CONTENTS OF MEMORY
001612 020203      CMP      R2,R3          ;WAS MEMORY LOCATION LOADED
                                ;WITH ITS ADDRESS
001614 001401      BEQ      3$
                                .IIF IDN <BC>,<BA>      HLT      1
                                .IIF IDN <BC>,<BC>      HLT      2      ;BUS ADDRESS MEMORY ERROR
                                ;BYTE COUNT MEMORY ERROR
001616 104002      EMT      2
001620 005204      3$:  INC      R4              ;ADVANCE TO NEXT LOCATION
001622 062702 010421      ADD      #10421,R2
001626 005300      DEC      R0
001630 001364      BNE      2$
001632 104400      4$:  SCOPE
                                XADRS=0
                                XCADRS=0
                                ADRS=XADRS
                                CADRS=XCADRS
                                .REPT 16.
001634      MENT2  ↑/BA/,↑/BUS ADDRESS/,\XADRS,\XCADRS,177777,↑/177777/
                                .NLIST
                                XADRS=XADRS+1
                                XCADRS=XCADRS+1
                                .LIST
                                .ENDR
                                MENT2  ↑/BA/,↑/BUS ADDRESS/,\XADRS,\XCADRS,177777,↑/177777/

                                ;BUS ADDRESS MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 0 OF BUS ADDRESS MEMORY
                                ;CAN BE SET TO 177777 AND CLEARED TO 0

001634      TS \XN,100,2$
001634 012767 000340 176134      T3:  MOV      #340,PS      ;DISABLE ALL INTERRUPTS
001642 012767 000100 007532      MOV      #100,ICOUNT    ;SET UP FOR 100 ITERATIONS
001650 012767 001730 007520      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3

```



```

002026 104400      2: SCOPE
000002      XADRS=XADRS+1
000002      XCADRS=XCADRS+1
002030      MEMENT2 ↑/BA/,↑/BUS ADDRESS/,\XADRS,\XCADRS,177777,↑/177777/

;BUS ADDRESS MEMORY DATA TEST
;VERIFY THAT ADDRESS 2 OF BUS ADDRESS MEMORY
;CAN BE SET TO 177777 AND CLEARED TO 0

002030      TS \XN,100,2:
002030 012767 000340 175740 T5: MOV #340,PS ;DISABLE ALL INTERRUPTS
002036 012767 000100 007336 MOV #100,ICOUNT ;SET UP FOR 100 ITERATIONS
002044 012767 002124 007324 MOV #2,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1

002052 000006
002052 012705 177777 MOV #177777,R5 ;EXPECTED RESULT=177777
002056 012777 000002 007250 MOV #2,@DHSCR ;SELECT LOCATION 2
;OF BUS ADDRESS MEMORY
002064 012777 177777 007250 MOV #177777,@DHBA ;WRITE 177777 INTO MEMORY
002072 017704 007244 MOV @DHBA,R4 ;READ CONTENTS OF MEMORY LOCATION
002076 020504 CMP R5,R4 ;COMPARE EXPECTED AND
002100 001401 BEQ # ;RECEIVED MEMORY CONTENTS
002102 .IIF IDN <BA>,<BA>, HLT 3 ;BUS ADDRESS MEMORY ERROR
002102 104003 EMT 3
.IIF IDN <BA>,<BC>, HLT 4 ;BYTE COUNT MEMORY ERROR
1: CLR R5 ;EXPECTED RESULT AFTER CLEAR=0
002104 005005 BIC #177777,@DHBA ;CLEAR MEMORY LOCATION
002106 042777 177777 007226 MOV @DHBA,R4 ;READ CONTENTS OF BUS ADDRESS
002114 017704 007222 ;MEMORY ADDRESS 2
002120 001401 BEQ #2:
002122 .IIF IDN <BA>,<BA>, HLT 3 ;BUS ADDRESS MEMORY ERROR
002122 104003 EMT 3
.IIF IDN <BA>,<BC>, HLT 4 ;BYTE COUNT MEMORY ERROR
;ADDRESS 2 NOT 0. ERROR

002124 104400      2: SCOPE
000003      XADRS=XADRS+1
000003      XCADRS=XCADRS+1
002126      MEMENT2 ↑/BA/,↑/BUS ADDRESS/,\XADRS,\XCADRS,177777,↑/177777/

;BUS ADDRESS MEMORY DATA TEST
;VERIFY THAT ADDRESS 3 OF BUS ADDRESS MEMORY
;CAN BE SET TO 177777 AND CLEARED TO 0

002126      TS \XN,100,2:
002126 012767 000340 175642 T6: MOV #340,PS ;DISABLE ALL INTERRUPTS
002134 012767 000100 007240 MOV #100,ICOUNT ;SET UP FOR 100 ITERATIONS
002142 012767 002222 007226 MOV #2,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1

002150 000007
002150 012705 177777 MOV #177777,R5 ;EXPECTED RESULT=177777
002154 012777 000003 007152 MOV #3,@DHSCR ;SELECT LOCATION 3
;OF BUS ADDRESS MEMORY
002162 012777 177777 007152 MOV #177777,@DHBA ;WRITE 177777 INTO MEMORY

```

```

002170 017704 007146      MOV      @DHBA,R4      ;READ CONTENTS OF MEMORY LOCATION
002174 020504              CMP      R5,R4        ;COMPARE EXPECTED AND
002176 001401              BEQ      1$           ;RECEIVED MEMORY CONTENTS
002200 104003      .IIF IDN <BA>,<BA>,    HLT      3           ;BUS ADDRESS MEMORY ERROR
002200 104003      EMT      3
002202 005005      .IIF IDN <BA>,<BC>,    HLT      4           ;BYTE COUNT MEMORY ERROR
002204 042777 177777 007130 1$: CLR      R5        ;EXPECTED RESULT AFTER CLEAR=0
002212 017704 007124      BIC      @177777,@DHBA ;CLEAR MEMORY LOCATION
002212 017704 007124      MOV      @DHBA,R4     ;READ CONTENTS OF BUS ADDRESS
002216 001401              BEQ      2$           ;MEMORY ADDRESS 3
002220 104003      .IIF IDN <BA>,<BA>,    HLT      3           ;BUS ADDRESS MEMORY ERROR
002220 104003      EMT      3
002222 104400      .IIF IDN <BA>,<BC>,    HLT      4           ;BYTE COUNT MEMORY ERROR
002222 104400      2$: SCOPE              ;ADDRESS 3 NOT 0, ERROR
000004 XADRS=XADRS+1
000004 XCADRS=XCADRS+1
002224 MEMENT2  ↑/BA/,↑/BUS ADDRESS/,\XADRS,\XCADRS,177777,↑/177777/

;BUS ADDRESS MEMORY DATA TEST
;VERIFY THAT ADDRESS 4 OF BUS ADDRESS MEMORY
;CAN BE SET TO 177777 AND CLEARED TO 0

002224 TS \XN,100,2$
002224 012767 000340 175544 T7: MOV      @340,PS      ;DISABLE ALL INTERRUPTS
002232 012767 000100 007142      MOV      @100,ICOUNT   ;SET UP FOR 100 ITERATIONS
002240 012767 002320 007130      MOV      @2$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
.IF NB <>
MOV      @,FREEZ1       ;SET UP TO LOOP WITH DATA      : 3
.ENDC
XN=XN+1
002246 000010      MOV      @177777,R5    ;EXPECTED RESULT=177777
002252 012777 000004 007054      MOV      @4,@DHSCR     ;SELECT LOCATION 4
002260 012777 177777 007054      MOV      @177777,@DHBA ;OF BUS ADDRESS MEMORY
002266 017704 007050      MOV      @DHBA,R4     ;WRITE 177777 INTO MEMORY
002272 020504      CMP      R5,R4        ;READ CONTENTS OF MEMORY LOCATION
002274 001401      BEQ      1$           ;COMPARE EXPECTED AND
002276 104003      .IIF IDN <BA>,<BA>,    HLT      3           ;RECEIVED MEMORY CONTENTS
002276 104003      EMT      3           ;BUS ADDRESS MEMORY ERROR
002300 005005      .IIF IDN <BA>,<BC>,    HLT      4           ;BYTE COUNT MEMORY ERROR
002302 042777 177777 007032 1$: CLR      R5        ;EXPECTED RESULT AFTER CLEAR=0
002310 017704 007026      BIC      @177777,@DHBA ;CLEAR MEMORY LOCATION
002310 017704 007026      MOV      @DHBA,R4     ;READ CONTENTS OF BUS ADDRESS
002314 001401              BEQ      2$           ;MEMORY ADDRESS 4
002316 104003      .IIF IDN <BA>,<BA>,    HLT      3           ;BUS ADDRESS MEMORY ERROR
002316 104003      EMT      3
002320 104400      .IIF IDN <BA>,<BC>,    HLT      4           ;BYTE COUNT MEMORY ERROR
002320 104400      2$: SCOPE              ;ADDRESS 4 NOT 0, ERROR
000005 XADRS=XADRS+1
000005 XCADRS=XCADRS+1
002322 MEMENT2  ↑/BA/,↑/BUS ADDRESS/,\XADRS,\XCADRS,177777,↑/177777/

;BUS ADDRESS MEMORY DATA TEST

```

;VERIFY THAT ADDRESS 5 OF BUS ADDRESS MEMORY
;CAN BE SET TO 177777 AND CLEARED TO 0

```

002322      000340 175446 TS \XN,100,2$
002322 012767 000340 175446 T10:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
002330 012767 000100 007044      MOV    #100,ICOUNT       ;SET UP FOR 100 ITERATIONS
002336 012767 002416 007032      MOV    #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
002344 000011      177777      MOV    #177777,R5       ;EXPECTED RESULT=177777
002350 012777 000005 006756      MOV    #5,@DHSCR       ;SELECT LOCATION 5
                                ;OF BUS ADDRESS MEMORY
002356 012777 177777 006756      MOV    #177777,@DHBA   ;WRITE 177777 INTO MEMORY
002364 017704 006752      MOV    @DHBA,R4        ;READ CONTENTS OF MEMORY LOCATION
002370 020504      CMP    R5,R4          ;COMPARE EXPECTED AND
002372 001401      BEQ    1$            ;RECEIVED MEMORY CONTENTS
002374 104003      .IIF IDN <BA>,<BA>,    HLT    3           ;BUS ADDRESS MEMORY ERROR
                                EMT    3
002376 005005      .IIF IDN <BA>,<BC>,    HLT    4           ;BYTE COUNT MEMORY ERROR
002400 042777 177777 006734 1$:   CLR    R5            ;EXPECTED RESULT AFTER CLEAR=0
002406 017704 006730      BIC    #177777,@DHBA  ;CLEAR MEMORY LOCATION
                                MOV    @DHBA,R4        ;READ CONTENTS OF BUS ADDRESS
                                ;MEMORY ADDRESS 5
002412 001401      BEQ    2$            ;EXPECTED RESULT=177777
002414 104003      .IIF IDN <BA>,<BA>,    HLT    3           ;BUS ADDRESS MEMORY ERROR
                                EMT    3
002416 104400      .IIF IDN <BA>,<BC>,    HLT    4           ;BYTE COUNT MEMORY ERROR
                                ;ADDRESS 5 NOT 0, ERROR
                                2$:   SCOPE
002420 000006      XADRS=XADRS+1
                                000006      XCADRS=XCADRS+1
                                MENT2  +/BA/,+/BUS ADDRESS/,\XADRS,\XCADRS,177777,+/177777/

```

;BUS ADDRESS MEMORY DATA TEST
;VERIFY THAT ADDRESS 6 OF BUS ADDRESS MEMORY
;CAN BE SET TO 177777 AND CLEARED TO 0

```

002420      000340 175350 TS \XN,100,2$
002420 012767 000340 175350 T11:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
002426 012767 000100 006746      MOV    #100,ICOUNT       ;SET UP FOR 100 ITERATIONS
002434 012767 002514 006734      MOV    #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
002442 000012      177777      MOV    #177777,R5       ;EXPECTED RESULT=177777
002446 012777 000006 006660      MOV    #6,@DHSCR       ;SELECT LOCATION 6
                                ;OF BUS ADDRESS MEMORY
002454 012777 177777 006660      MOV    #177777,@DHBA   ;WRITE 177777 INTO MEMORY
002462 017704 006654      MOV    @DHBA,R4        ;READ CONTENTS OF MEMORY LOCATION
002466 020504      CMP    R5,R4          ;COMPARE EXPECTED AND
002470 001401      BEQ    1$            ;RECEIVED MEMORY CONTENTS
002472 104003      .IIF IDN <BA>,<BA>,    HLT    3           ;BUS ADDRESS MEMORY ERROR
                                EMT    3
002472 104003      .IIF IDN <BA>,<BC>,    HLT    4           ;BYTE COUNT MEMORY ERROR

```

```

002474 005005      1$:   CLR      R5                ;EXPECTED RESULT AFTER CLEAR=0
002476 042777 177777 006636   BIC      #177777,@DHBA        ;CLEAR MEMORY LOCATION
002504 017704 006632      MOV      @DHBA,R4            ;READ CONTENTS OF BUS ADDRESS
                                ;MEMORY ADDRESS 6
002510 001401      BEQ      2$
002512      .IIF IDN <BA>,<BA>,   HLT      3                ;BUS ADDRESS MEMORY ERROR
002512 104003      EMT      3
                                .IIF IDN <BA>,<BC>,   HLT      4                ;BYTE COUNT MEMORY ERROR
                                ;ADDRESS 6 NOT 0, ERROR
002514 104400      2$:   SCOPE
000007 XADRS=XADRS+1
000007 XCADRS=XCADRS+1
002516 MEMENT2  ↑/BA/,↑/BUS ADDRESS/,\XADRS,\XCADRS,177777,↑/177777/

                                ;BUS ADDRESS MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 7 OF BUS ADDRESS MEMORY
                                ;CAN BE SET TO 177777 AND CLEARED TO 0

002516      TS \XN,100,2$
002516 012767 000340 175252 T12:   MOV      #340,PS        ;DISABLE ALL INTERRUPTS
002524 012767 000100 006650   MOV      #100,ICOUNT       ;SET UP FOR 100 ITERATIONS
002532 012767 002612 006636   MOV      #2$,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #,FREEZ1        ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
002540 000013      MOV      #177777,R5        ;EXPECTED RESULT=177777
002544 012777 000007 006562   MOV      #7,@DHSCR         ;SELECT LOCATION 7
                                ;OF BUS ADDRESS MEMORY
002552 012777 177777 006562   MOV      #177777,@DHBA     ;WRITE 177777 INTO MEMORY
002560 017704 006556      MOV      @DHBA,R4         ;READ CONTENTS OF MEMORY LOCATION
002564 020504      CMP      R5,R4           ;COMPARE EXPECTED AND
002566 001401      BEQ      1$              ;RECEIVED MEMORY CONTENTS
002570      .IIF IDN <BA>,<BA>,   HLT      3                ;BUS ADDRESS MEMORY ERROR
002570 104003      EMT      3
                                .IIF IDN <BA>,<BC>,   HLT      4                ;BYTE COUNT MEMORY ERROR
002572 005005      1$:   CLR      R5                ;EXPECTED RESULT AFTER CLEAR=0
002574 042777 177777 006540   BIC      #177777,@DHBA     ;CLEAR MEMORY LOCATION
002602 017704 006534      MOV      @DHBA,R4         ;READ CONTENTS OF BUS ADDRESS
                                ;MEMORY ADDRESS 7
002606 001401      BEQ      2$
002610      .IIF IDN <BA>,<BA>,   HLT      3                ;BUS ADDRESS MEMORY ERROR
002610 104003      EMT      3
                                .IIF IDN <BA>,<BC>,   HLT      4                ;BYTE COUNT MEMORY ERROR
                                ;ADDRESS 7 NOT 0, ERROR
002612 104400      2$:   SCOPE
000010 XADRS=XADRS+1
000010 XCADRS=XCADRS+1
002614 MEMENT2  ↑/BA/,↑/BUS ADDRESS/,\XADRS,\XCADRS,177777,↑/177777/

                                ;BUS ADDRESS MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 10 OF BUS ADDRESS MEMORY
                                ;CAN BE SET TO 177777 AND CLEARED TO 0

002614      TS \XN,100,2$
002614 012767 000340 175154 T13:   MOV      #340,PS        ;DISABLE ALL INTERRUPTS
002622 012767 000100 006552   MOV      #100,ICOUNT       ;SET UP FOR 100 ITERATIONS

```

```

002630 012767 002710 006540      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
002636 012705 177777      MOV      #177777,R5      ;EXPECTED RESULT=177777
002642 012777 000010 006464      MOV      #10,@DHSCR      ;SELECT LOCATION 10
      ;OF BUS ADDRESS MEMORY
002650 012777 177777 006464      MOV      #177777,@DHBA      ;WRITE 177777 INTO MEMORY
002656 017704 006460      MOV      @DHBA,R4      ;READ CONTENTS OF MEMORY LOCATION
002662 020504      CMP      R5,R4      ;COMPARE EXPECTED AND
002664 001401      BEQ      1$      ;RECEIVED MEMORY CONTENTS
002666      .IIF IDN <BA>,<BA>,      HLT      3      ;BUS ADDRESS MEMORY ERROR
002666 104003      EMT      3
      .IIF IDN <BA>,<BC>,      HLT      4      ;BYTE COUNT MEMORY ERROR
002670 005005      1$:      CLR      R5      ;EXPECTED RESULT AFTER CLEAR=0
002672 042777 177777 006442      BIC      #177777,@DHBA      ;CLEAR MEMORY LOCATION
002700 017704 006436      MOV      @DHBA,R4      ;READ CONTENTS OF BUS ADDRESS
      ;MEMORY ADDRESS 10
002704 001401      BEQ      2$
002706      .IIF IDN <BA>,<BA>,      HLT      3      ;BUS ADDRESS MEMORY ERROR
002706 104003      EMT      3
      .IIF IDN <BA>,<BC>,      HLT      4      ;BYTE COUNT MEMORY ERROR
      ;ADDRESS 10 NOT 0, ERROR
002710 104400      2$:      SCOPE
000011      XADRS=XADRS+1
000011      XCADRS=XCADRS+1
002712      MENT2  ↑/BA/,↑/BUS ADDRESS/,\XADRS,\XCADRS,177777,↑/177777/

      ;BUS ADDRESS MEMORY DATA TEST
      ;VERIFY THAT ADDRESS 11 OF BUS ADDRESS MEMORY
      ;CAN BE SET TO 177777 AND CLEARED TO 0

002712      TS \XN,100,2$
002712 012767 000340 175056      T14:      MOV      #340,PS      ;DISABLE ALL INTERRUPTS
002720 012767 000100 006454      MOV      #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
002726 012767 003006 006442      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
002734 012705 177777      MOV      #177777,R5      ;EXPECTED RESULT=177777
002740 012777 000011 006366      MOV      #11,@DHSCR      ;SELECT LOCATION 11
      ;OF BUS ADDRESS MEMORY
002746 012777 177777 006366      MOV      #177777,@DHBA      ;WRITE 177777 INTO MEMORY
002754 017704 006362      MOV      @DHBA,R4      ;READ CONTENTS OF MEMORY LOCATION
002760 020504      CMP      R5,R4      ;COMPARE EXPECTED AND
002762 001401      BEQ      1$      ;RECEIVED MEMORY CONTENTS
002764      .IIF IDN <BA>,<BA>,      HLT      3      ;BUS ADDRESS MEMORY ERROR
002764 104003      EMT      3
      .IIF IDN <BA>,<BC>,      HLT      4      ;BYTE COUNT MEMORY ERROR
002766 005005      1$:      CLR      R5      ;EXPECTED RESULT AFTER CLEAR=0
002770 042777 177777 006344      BIC      #177777,@DHBA      ;CLEAR MEMORY LOCATION
002776 017704 006340      MOV      @DHBA,R4      ;READ CONTENTS OF BUS ADDRESS
      ;MEMORY ADDRESS 11
003002 001401      BEQ      2$
003004      .IIF IDN <BA>,<BA>,      HLT      3      ;BUS ADDRESS MEMORY ERROR

```

```

003004 104003          EMT      3
                   .IIF IDN    <BA>,<BC>,    HLT      4          ;BYTE COUNT MEMORY ERROR
                                     ;ADDRESS 11 NOT 0, ERROR
003006 104400          2$:      SCOPE
000012              XADRS=XADRS+1
000012              XCADRS=XCADRS+1
003010              MEMT2  ↑/BA/,↑/BUS ADDRESS/,\XADRS,\XCADRS,17777,↑/17777/

                   ;BUS ADDRESS MEMORY DATA TEST
                   ;VERIFY THAT ADDRESS 12 OF BUS ADDRESS MEMORY
                   ;CAN BE SET TO 17777 AND CLEARED TO 0

003010              TS \XN,100,2$
003010 012767 000340 174760 T15:   MOV      #340,PS          ;DISABLE ALL INTERRUPTS
003016 012767 000100 006356      MOV      #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
003024 012767 003104 006344      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                   .IF NB    <>
                   MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                   .ENDC
                   XN=XN+1
003032 000016
003032 012705 177777          MOV      #177777,R5      ;EXPECTED RESULT=177777
003036 012777 000012 006270      MOV      #12,@DHSCR      ;SELECT LOCATION 12
                                     ;OF BUS ADDRESS MEMORY
003044 012777 177777 006270      MOV      #177777,@DHBA  ;WRITE 177777 INTO MEMORY
003052 017704 006264          MOV      @DHBA,R4        ;READ CONTENTS OF MEMORY LOCATION
003056 020504          CMP      R5,R4          ;COMPARE EXPECTED AND
003060 001401          BEQ      1$          ;RECEIVED MEMORY CONTENTS
003062 104003          .IIF IDN    <BA>,<BA>,    HLT      3          ;BUS ADDRESS MEMORY ERROR
                                     EMT      3
003064 005005          .IIF IDN    <BA>,<BC>,    HLT      4          ;BYTE COUNT MEMORY ERROR
003066 042777 177777 006246      1$:     CLR      R5          ;EXPECTED RESULT AFTER CLEAR=0
003074 017704 006242          BIC      #177777,@DHBA  ;CLEAR MEMORY LOCATION
                                     MOV      @DHBA,R4        ;READ CONTENTS OF BUS ADDRESS
                                     ;MEMORY ADDRESS 12
003100 001401          BEQ      2$
003102 104003          .IIF IDN    <BA>,<BA>,    HLT      3          ;BUS ADDRESS MEMORY ERROR
                                     CHT      3
003102 104003          .IIF IDN    <BA>,<BC>,    HLT      4          ;BYTE COUNT MEMORY ERROR
                                     ;ADDRESS 12 NOT 0, ERROR
003104 104400          2$:      SCOPE
000013              XADRS=XADRS+1
000013              XCADRS=XCADRS+1
003106              MEMT2  ↑/BA/,↑/BUS ADDRESS/,\XADRS,\XCADRS,17777,↑/17777/

                   ;BUS ADDRESS MEMORY DATA TEST
                   ;VERIFY THAT ADDRESS 13 OF BUS ADDRESS MEMORY
                   ;CAN BE SET TO 17777 AND CLEARED TO 0

003106              TS \XN,100,2$
003106 012767 000340 174662 T16:   MOV      #340,PS          ;DISABLE ALL INTERRUPTS
003114 012767 000100 006260      MOV      #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
003122 012767 003202 006246      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                   .IF NB    <>
                   MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                   .ENDC
                   XN=XN+1
003130 000017
003130 012705 177777          MOV      #177777,R5      ;EXPECTED RESULT=177777

```

```

003134 012777 000013 006172      MOV      #13, @DHSCR      ;SELECT LOCATION 13
003142 012777 177777 006172      MOV      #177777, @DHBA  ;OF BUS ADDRESS MEMORY
003150 017704 006166                MOV      @DHBA, R4       ;WRITE 177777 INTO MEMORY
003154 020504                CMP      R5, R4         ;READ CONTENTS OF MEMORY LOCATION
003156 001401                BEQ      1$             ;COMPARE EXPECTED AND
003160                .IIF IDN <BA>, <BA>, HLT 3      ;RECEIVED MEMORY CONTENTS
003160 104003                EMT      3             ;BUS ADDRESS MEMORY ERROR
003162 005005                .IIF IDN <BA>, <BC>, HLT 4      ;BYTE COUNT MEMORY ERROR
003164 042777 177777 006150 1$: CLR      R5           ;EXPECTED RESULT AFTER CLEAR=0
003172 017704 006144                BIC      #177777, @DHBA ;CLEAR MEMORY LOCATION
003176 001401                MOV      @DHBA, R4     ;READ CONTENTS OF BUS ADDRESS
003200                BEQ      2$             ;MEMORY ADDRESS 13
003200 104003                .IIF IDN <BA>, <BA>, HLT 3      ;BUS ADDRESS MEMORY ERROR
003200                EMT      3             ;
003202 104400                .IIF IDN <BA>, <BC>, HLT 4      ;BYTE COUNT MEMORY ERROR
000014                2$: SCOPE              ;ADDRESS 13 NOT 0, ERROR
000014                XADRS=XADRS+1
003204 000014                XCADRS=XCADRS+1
003204                MEMENT2 ↑/BA/,↑/BUS ADDRESS/, \XADRS, \XCADRS, 177777, ↑/177777/

;BUS ADDRESS MEMORY DATA TEST
;VERIFY THAT ADDRESS 14 OF BUS ADDRESS MEMORY
;CAN BE SET TO 177777 AND CLEARED TO 0

003204                TS \XN, 100, 2$
003204 012767 000340 174564 T17: MOV      #340, PS     ;DISABLE ALL INTERRUPTS
003212 012767 000100 006162      MOV      #100, ICOUNT   ;SET UP FOR 100 ITERATIONS
003220 012767 003300 006150      MOV      #2$, ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
003220                .IF NB <>
003220                MOV      #, FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
003220                .ENDC
003220                XN=XN+1
003226 012705 177777                MOV      #177777, R5    ;EXPECTED RESULT=177777
003232 012777 000014 006074      MOV      #14, @DHSCR   ;SELECT LOCATION 14
003240 012777 177777 006074      MOV      #177777, @DHBA ;OF BUS ADDRESS MEMORY
003246 017704 006070      MOV      @DHBA, R4     ;WRITE 177777 INTO MEMORY
003252 020504                CMP      R5, R4         ;READ CONTENTS OF MEMORY LOCATION
003254 001401                BEQ      1$             ;COMPARE EXPECTED AND
003256                .IIF IDN <BA>, <BA>, HLT 3      ;RECEIVED MEMORY CONTENTS
003256 104003                EMT      3             ;BUS ADDRESS MEMORY ERROR
003260 005005                .IIF IDN <BA>, <BC>, HLT 4      ;BYTE COUNT MEMORY ERROR
003262 042777 177777 006052 1$: CLR      R5           ;EXPECTED RESULT AFTER CLEAR=0
003270 017704 006046                BIC      #177777, @DHBA ;CLEAR MEMORY LOCATION
003274 001401                MOV      @DHBA, R4     ;READ CONTENTS OF BUS ADDRESS
003276                BEQ      2$             ;MEMORY ADDRESS 14
003276 104003                .IIF IDN <BA>, <BA>, HLT 3      ;BUS ADDRESS MEMORY ERROR
003276                EMT      3             ;
003300                .IIF IDN <BA>, <BC>, HLT 4      ;BYTE COUNT MEMORY ERROR
003300 104400                2$: SCOPE              ;ADDRESS 14 NOT 0, ERROR
000015                XADRS=XADRS+1
000015                XCADRS=XCADRS+1

```

003302

MEMT2 †/BA/,†/BUS ADDRESS/, \XADRS, \XCADRS, 177777, †/177777/

;BUS ADDRESS MEMORY DATA TEST
 ;VERIFY THAT ADDRESS 15 OF BUS ADDRESS MEMORY
 ;CAN BE SET TO 177777 AND CLEARED TO 0

003302

003302 012767 000340 174466
 003310 012767 000100 006064
 003316 012767 003376 006052

TS \XN,100,2‡

T20:

MOV #340,PS
 MOV #100,ICOUNT
 MOV #2‡,ESCAPE
 .IF NB <>
 MOV †,FREEZ1

;DISABLE ALL INTERRUPTS
 ;SET UP FOR 100 ITERATIONS
 ;SET UP TO ESCAPE TO NEXT TEST
 ;SET UP TO LOOP WITH DATA ; 3

.ENDC

XN=XN+1

003324 000021
 003330 012705 177777 000015 005776

MOV #177777,R5
 MOV #15,@DHSCR

;EXPECTED RESULT=177777
 ;SELECT LOCATION 15
 ;OF BUS ADDRESS MEMORY
 ;WRITE 177777 INTO MEMORY
 ;READ CONTENTS OF MEMORY LOCATION
 ;COMPARE EXPECTED AND
 ;RECEIVED MEMORY CONTENTS

003336 012777 177777 005776
 003344 017704 005772
 003350 020504
 003352 001401

MOV #177777,@DHBA
 MOV @DHBA,R4
 CMP R5,R4
 BEQ 1‡

003354
 003354 104003

.IIF IDN <BA>,<BA>,
 EMT 3

HLT

3 ;BUS ADDRESS MEMORY ERROR

003356 005005
 003360 042777 177777 005754
 003366 017704 005750

.IIF IDN <BA>,<BC>,
 1‡: CLR R5
 BIC #177777,@DHBA
 MOV @DHBA,R4

HLT

4 ;BYTE COUNT MEMORY ERROR
 ;EXPECTED RESULT AFTER CLEAR=0
 ;CLEAR MEMORY LOCATION
 ;READ CONTENTS OF BUS ADDRESS
 ;MEMORY ADDRESS 15

003372 001401
 003374
 003374 104003

BEQ 2‡
 .IIF IDN <BA>,<BA>,
 EMT 3

HLT

3 ;BUS ADDRESS MEMORY ERROR

.IIF IDN <BA>,<BC>,
 EMT 4

HLT

4 ;BYTE COUNT MEMORY ERROR
 ;ADDRESS 15 NOT 0, ERROR

003376 104400
 000016
 000016

2‡: SCOPE
 XADRS=XADRS+1
 XCADRS=XCADRS+1

003400

MEMT2 †/BA/,†/BUS ADDRESS/, \XADRS, \XCADRS, 177777, †/177777/

;BUS ADDRESS MEMORY DATA TEST
 ;VERIFY THAT ADDRESS 16 OF BUS ADDRESS MEMORY
 ;CAN BE SET TO 177777 AND CLEARED TO 0

003400

003400 012767 000340 174370
 003406 012767 000100 005766
 003414 012767 003474 005754

TS \XN,100,2‡

T21:

MOV #340,PS
 MOV #100,ICOUNT
 MOV #2‡,ESCAPE
 .IF NB <>
 MOV †,FREEZ1

;DISABLE ALL INTERRUPTS
 ;SET UP FOR 100 ITERATIONS
 ;SET UP TO ESCAPE TO NEXT TEST
 ;SET UP TO LOOP WITH DATA ; 3

.ENDC

XN=XN+1

003422 000022
 003426 012705 177777 000016 005700

MOV #177777,R5
 MOV #16,@DHSCR

;EXPECTED RESULT=177777
 ;SELECT LOCATION 16
 ;OF BUS ADDRESS MEMORY
 ;WRITE 177777 INTO MEMORY
 ;READ CONTENTS OF MEMORY LOCATION
 ;COMPARE EXPECTED AND
 ;RECEIVED MEMORY CONTENTS

003434 012777 177777 005700
 003442 017704 005674
 003446 020504
 003450 001401

MOV #177777,@DHBA
 MOV @DHBA,R4
 CMP R5,R4
 BEQ 1‡

```

003452      .IIF IDN      <BA>,<BA>,      HLT      3      ;BUS ADDRESS MEMORY ERROR
003452 104003      EMT      3
003454 005005      .IIF IDN      <BA>,<BC>,      HLT      4      ;BYTE COUNT MEMORY ERROR
003456 042777 177777 005656 1$: CLR      R5      ;EXPECTED RESULT AFTER CLEAR=0
003464 017704 005652      BIC      #177777,@DHBA ;CLEAR MEMORY LOCATION
                                MOV      @DHBA,R4      ;READ CONTENTS OF BUS ADDRESS
                                ;MEMORY ADDRESS 16
003470 001401      BEQ      2$
003472 104003      .IIF IDN      <BA>,<BA>,      HLT      3      ;BUS ADDRESS MEMORY ERROR
                                EMT      3
                                .IIF IDN      <BA>,<BC>,      HLT      4      ;BYTE COUNT MEMORY ERROR
                                ;ADDRESS 16 NOT 0, ERROR
003474 104400      2$: SCOPE
                                XADRS=XADRS+1
                                XCADRS=XCADRS+1
003476      MENT2  +/BA/,+/BUS ADDRESS/,\XADRS,\XCADRS,177777,+/177777/

                                ;BUS ADDRESS MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 17 OF BUS ADDRESS MEMORY
                                ;CAN BE SET TO 177777 AND CLEARED TO 0

003476      TS \XN,100,2$
003476 012767 000340 174272 T22: MOV      #340,PS      ;DISABLE ALL INTERRUPTS
0C3504 012767 000100 005670      MOV      #100,ICOUNT ;SET UP FOR 100 ITERATIONS
003512 012767 003572 005656      MOV      #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB      <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      : 3
                                .ENDC
                                XN=XN+1
003520 012705 177777      MOV      #177777,R5      ;EXPECTED RESULT=177777
003524 012777 000017 005602      MOV      #17,@DHSCR ;SELECT LOCATION 17
                                ;OF BUS ADDRESS MEMORY
                                ;WRITE 177777 INTO MEMORY
003532 012777 177777 005602      MOV      #177777,@DHBA ;READ CONTENTS OF MEMORY LOCATION
003540 017704 005576      MOV      @DHBA,R4      ;COMPARE EXPECTED AND
003544 020504      CMP      R5,R4      ;RECEIVED MEMORY CONTENTS
003546 001401      BEQ      1$
003550      .IIF IDN      <BA>,<BA>,      HLT      3      ;BUS ADDRESS MEMORY ERROR
003550 104003      EMT      3
003552 005005      .IIF IDN      <BA>,<BC>,      HLT      4      ;BYTE COUNT MEMORY ERROR
003554 042777 177777 005560 1$: CLR      R5      ;EXPECTED RESULT AFTER CLEAR=0
003562 017704 005554      BIC      #177777,@DHBA ;CLEAR MEMORY LOCATION
                                MOV      @DHBA,R4      ;READ CONTENTS OF BUS ADDRESS
                                ;MEMORY ADDRESS 17
003566 001401      BEQ      2$
003570      .IIF IDN      <BA>,<BA>,      HLT      3      ;BUS ADDRESS MEMORY ERROR
003570 104003      EMT      3
                                .IIF IDN      <BA>,<BC>,      HLT      4      ;BYTE COUNT MEMORY ERROR
                                ;ADDRESS 17 NOT 0, ERROR
003572 104400      2$: SCOPE
                                XADRS=XADRS+1
                                XCADRS=XCADRS+1
18      XADRS=0
19      XCADRS=0
20      ADRS=XADRS
21      CADRS=XCADRS
23      .REPT 16.
24      MENT2  +/BC/,+/BYTE COUNT/,\XADRS,\XCADRS,177777,+/177777/

```

```

25
26
27
28
29
003574      .NLIST
             XADRS=XADRS+1
             XCADRS=XCADRS+1
             .LIST
             .ENDR
             MEMENT2  ↑/BC/,↑/BYTE COUNT/,\XADRS,\XCADRS,177777,↑/177777/

             ;BYTE COUNT MEMORY DATA TEST
             ;VERIFY THAT ADDRESS 0 OF BYTE COUNT MEMORY
             ;CAN BE SET TO 177777 AND CLEARED TO 0

003574      TS \XN,100,2‡
003574 012767 000340 174174 T23:  MOV  #340,PS           ;DISABLE ALL INTERRUPTS
003602 012767 000100 005572      MOV  #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
003610 012767 003670 005560      MOV  #2‡,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
             .IF NB  <>
             MOV  #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
             .ENDC
             XN=XN+1
003616 000024
003616 012705 177777
003622 012777 000000 005504      MOV  #177777,R5     ;EXPECTED RESULT=177777
             MOV  #0,@DHSCR          ;SELECT LOCATION 0
             MOV  #177777,@DHBC     ;OF BYTE COUNT MEMORY
             MOV  @DHBC,R4          ;WRITE 177777 INTO MEMORY
             CMP  R5,R4             ;READ CONTENTS OF MEMORY LOCATION
             BEQ  1‡                ;COMPARE EXPECTED AND
             .IIF IDN <BC>,<BA>,    HLT  3           ;RECEIVED MEMORY CONTENTS
             .IIF IDN <BC>,<BC>,    HLT  4           ;BUS ADDRESS MEMORY ERROR
003646 104004
003646 005005
003650 005005
003652 042777 177777 005464      1‡:  CLR  R5           ;EXPECTED RESULT AFTER CLEAR=0
             BIC  #177777,@DHBC     ;CLEAR MEMORY LOCATION
             MOV  @DHBC,R4          ;READ CONTENTS OF BYTE COUNT
             BEQ  2‡                ;MEMORY ADDRESS 0
003664 001401
             .IIF IDN <BC>,<BA>,    HLT  3           ;BUS ADDRESS MEMORY ERROR
             .IIF IDN <BC>,<BC>,    HLT  4           ;BYTE COUNT MEMORY ERROR
003666 104004
             EMT  4
             ;ADDRESS 0 NOT 0, ERROR

003670 104400
000001
000001
003672      2‡:  SCOPE
             XADRS=XADRS+1
             XCADRS=XCADRS+1
             MEMENT2  ↑/BC/,↑/BYTE COUNT/,\XADRS,\XCADRS,177777,↑/177777/

             ;BYTE COUNT MEMORY DATA TEST
             ;VERIFY THAT ADDRESS 1 OF BYTE COUNT MEMORY
             ;CAN BE SET TO 177777 AND CLEARED TO 0

003672      TS \XN,100,2‡
003672 012767 000340 174076 T24:  MOV  #340,PS           ;DISABLE ALL INTERRUPTS
003700 012767 000100 005474      MOV  #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
003706 012767 003766 005462      MOV  #2‡,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
             .IF NB  <>
             MOV  #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
             .ENDC
             XN=XN+1
003714 000025
003714 012705 177777
003720 012777 000001 005406      MOV  #177777,R5     ;EXPECTED RESULT=177777
             MOV  #1,@DHSCR          ;SELECT LOCATION 1

```

```

003726 012777 177777 005410      MOV      #177777,@DHBC      ;OF BYTE COUNT MEMORY
003734 017704 005404              MOV      @DHBC,R4          ;WRITE 177777 INTO MEMORY
003740 020504              CMP      R5,R4            ;READ CONTENTS OF MEMORY LOCATION
003742 001401              BEQ      1$               ;COMPARE EXPECTED AND
                                ;RECEIVED MEMORY CONTENTS
003744              .IIF IDN      <BC>,<BA>,      HLT      3              ;BUS ADDRESS MEMORY ERROR
                                .IIF IDN      <BC>,<BC>,      HLT      4              ;BYTE COUNT MEMORY ERROR
003744 104004              EMT      4
003746 005005      1$:      CLR      R5          ;EXPECTED RESULT AFTER CLEAR=0
003750 042777 177777 005366      BIC      @177777,@DHBC    ;CLEAR MEMORY LOCATION
003756 017704 005362              MOV      @DHBC,R4        ;READ CONTENTS OF BYTE COUNT
                                ;MEMORY ADDRESS 1
003762 001401              BEQ      2$
003764              .IIF IDN      <BC>,<BA>,      HLT      3              ;BUS ADDRESS MEMORY ERROR
003764 104004              .IIF IDN      <BC>,<BC>,      HLT      4              ;BYTE COUNT MEMORY ERROR
                                EMT      4
003766 104400      2$:      SCOPE
                                XADRS=XADRS+1
                                XCADRS=XCADRS+1
                                MENT2  +/BC/,+/BYTE COUNT/,\XADRS,\XCADRS,177777,+/177777/
                                ;ADDRESS 1 NOT 0, ERROR
003770              ;BYTE COUNT MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 2 OF BYTE COUNT MEMORY
                                ;CAN BE SET TO 177777 AND CLEARED TO 0

003770              TS \XN,100,2$
003770 012767 000340 174000      T25:     MOV      #340,PS      ;DISABLE ALL INTERRUPTS
003776 012767 000100 005376      MOV      #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
004004 012767 004064 005364      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB      <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
004012 000026              MOV      @177777,R5      ;EXPECTED RESULT=177777
004016 012777 000002 005310      MOV      @2,@DHSCR      ;SELECT LOCATION 2
                                ;OF BYTE COUNT MEMORY
004024 012777 177777 005312      MOV      @177777,@DHBC    ;WRITE 177777 INTO MEMORY
004032 017704 005306              MOV      @DHBC,R4        ;READ CONTENTS OF MEMORY LOCATION
004036 020504              CMP      R5,R4            ;COMPARE EXPECTED AND
004040 001401              BEQ      1$               ;RECEIVED MEMORY CONTENTS
                                .IIF IDN      <BC>,<BA>,      HLT      3              ;BUS ADDRESS MEMORY ERROR
                                .IIF IDN      <BC>,<BC>,      HLT      4              ;BYTE COUNT MEMORY ERROR
004042 104004              EMT      4
004044 005005      1$:      CLR      R5          ;EXPECTED RESULT AFTER CLEAR=0
004046 042777 177777 005270      BIC      @177777,@DHBC    ;CLEAR MEMORY LOCATION
004054 017704 005264              MOV      @DHBC,R4        ;READ CONTENTS OF BYTE COUNT
                                ;MEMORY ADDRESS 2
004060 001401              BEQ      2$
004062              .IIF IDN      <BC>,<BA>,      HLT      3              ;BUS ADDRESS MEMORY ERROR
004062 104004              .IIF IDN      <BC>,<BC>,      HLT      4              ;BYTE COUNT MEMORY ERROR
                                EMT      4
004064 104400      2$:      SCOPE
                                XADRS=XADRS+1
                                XCADRS=XCADRS+1
                                MENT2  +/BC/,+/BYTE COUNT/,\XADRS,\XCADRS,177777,+/177777/
004066

```

```

;BYTE COUNT MEMORY DATA TEST
;VERIFY THAT ADDRESS 3 OF BYTE COUNT MEMORY
;CAN BE SET TO 177777 AND CLEARED TO 0

```

```

004066      000027      TS \XN,100,2#
004066 012767 000340 173702 T26:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
004074 012767 000100 005300      MOV    #100,ICOUNT       ;SET UP FOR 100 ITERATIONS
004102 012767 004162 005266      MOV    #2#,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
004110      000027      MOV    #177777,R5       ;EXPECTED RESULT=177777
004114 012777 000003 005212      MOV    #3,@DHSCR        ;SELECT LOCATION 3
                                ;OF BYTE COUNT MEMORY
004122 012777 177777 005214      MOV    #177777,@DHBC    ;WRITE 177777 INTO MEMORY
004130 017704 005210      MOV    @DHBC,R4         ;READ CONTENTS OF MEMORY LOCATION
004134 020504      CMP    R5,R4            ;COMPARE EXPECTED AND
004136 001401      BEQ    1#               ;RECEIVED MEMORY CONTENTS
                                ;BUS ADDRESS MEMORY ERROR
004140      .IIF IDN    <BC>,<BA>,    HLT    3
004140 104004      .IIF IDN    <BC>,<BC>,    HLT    4           ;BYTE COUNT MEMORY ERROR
004142 005005      EMT    4
004144 042777 177777 005172 1#:   CLR    R5             ;EXPECTED RESULT AFTER CLEAR=0
004152 017704 005166      BIC    #177777,@DHBC   ;CLEAR MEMORY LOCATION
                                MOV    @DHBC,R4         ;READ CONTENTS OF BYTE COUNT
                                ;MEMORY ADDRESS 3
004156 001401      BEQ    2#
                                ;BUS ADDRESS MEMORY ERROR
004160      .IIF IDN    <BC>,<BA>,    HLT    3
004160 104004      .IIF IDN    <BC>,<BC>,    HLT    4           ;BYTE COUNT MEMORY ERROR
                                EMT    4
                                ;ADDRESS 3 NOT 0, ERROR
004162 104400      2#:   SCOPE
000004      XADRS=XADRS+1
000004      XCADRS=XCADRS+1
004164      MENT2  ↑/BC/,↑/BYTE COUNT/,\XADRS,\XCADRS,177777,↑/177777/

```

```

;BYTE COUNT MEMORY DATA TEST
;VERIFY THAT ADDRESS 4 OF BYTE COUNT MEMORY
;CAN BE SET TO 177777 AND CLEARED TO 0

```

```

004164      000030      TS \XN,100,2#
004164 012767 000340 173604 T27:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
004172 012767 000100 005202      MOV    #100,ICOUNT       ;SET UP FOR 100 ITERATIONS
004200 012767 004260 005170      MOV    #2#,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
004206      000030      MOV    #177777,R5       ;EXPECTED RESULT=177777
004212 012777 000004 005114      MOV    #4,@DHSCR        ;SELECT LOCATION 4
                                ;OF BYTE COUNT MEMORY
004220 012777 177777 005116      MOV    #177777,@DHBC    ;WRITE 177777 INTO MEMORY
004226 017704 005112      MOV    @DHBC,R4         ;READ CONTENTS OF MEMORY LOCATION
004232 020504      CMP    R5,R4            ;COMPARE EXPECTED AND
004234 001401      BEQ    1#               ;RECEIVED MEMORY CONTENTS
                                ;BUS ADDRESS MEMORY ERROR
                                .IIF IDN    <BC>,<BA>,    HLT    3

```

```

004236      .IIF IDN      <BC>,<BC>      HLT      4      ;BYTE COUNT MEMORY ERROR
004236      104004      EMT      4
004240      005005      1$:      CLR      R5      ;EXPECTED RESULT AFTER CLEAR=0
004242      042777      177777      ^05074      BIC      #177777,@DHBC      ;CLEAR MEMORY LOCATION
004250      017704      005070      MOV      @DHBC,R4      ;READ CONTENTS OF BYTE COUNT
                                ;MEMORY ADDRESS 4
004254      001401      BEQ      2$
004256      .IIF IDN      <BC>,<BA>      HLT      3      ;BUS ADDRESS MEMORY ERROR
004256      104004      .IIF IDN      <BC>,<BC>      HLT      4      ;BYTE COUNT MEMORY ERROR
                                EMT      4
                                ;ADDRESS 4 NOT 0. ERROR
004260      104400      2$:      SCOPE
000005      XADRS=XADRS+1
000005      XCADRS=XCADRS+1
004262      MENT2      †/BC/,†/BYTE COUNT/,\XADRS,\XCADRS,177777,†/177777/
                                ;BYTE COUNT MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 5 OF BYTE COUNT MEMORY
                                ;CAN BE SET TO 177777 AND CLEARED TO 0

004262      012767      000340      173506      TS \XN,100,2$
004270      012767      000100      005104      T30:      MOV      #340,PS      ;DISABLE ALL INTERRUPTS
004276      012767      004356      005072      MOV      #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
                                MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB      <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
004304      012705      177777      MOV      #177777,R5      ;EXPECTED RESULT=177777
004310      012777      000005      005016      MOV      #5,@DHSCR      ;SELECT LOCATION 5
                                ;OF BYTE COUNT MEMORY
004316      012777      177777      005020      MOV      #177777,@DHBC      ;WRITE 177777 INTO MEMORY
004324      017704      005014      MOV      @DHBC,R4      ;READ CONTENTS OF MEMORY LOCATION
004330      020504      CMP      R5,R4      ;COMPARE EXPECTED AND
004332      001401      BEQ      1$      ;RECEIVED MEMORY CONTENTS
004334      .IIF IDN      <BC>,<BA>      HLT      3      ;BUS ADDRESS MEMORY ERROR
004334      104004      .IIF IDN      <BC>,<BC>      HLT      4      ;BYTE COUNT MEMORY ERROR
004336      005005      1$:      CLR      R5      ;EXPECTED RESULT AFTER CLEAR=0
004340      042777      177777      004776      BIC      #177777,@DHBC      ;CLEAR MEMORY LOCATION
004346      017704      004772      MOV      @DHBC,R4      ;READ CONTENTS OF BYTE COUNT
                                ;MEMORY ADDRESS 5
004352      001401      BEQ      2$
004354      .IIF IDN      <BC>,<BA>      HLT      3      ;BUS ADDRESS MEMORY ERROR
004354      104004      .IIF IDN      <BC>,<BC>      HLT      4      ;BYTE COUNT MEMORY ERROR
                                EMT      4
                                ;ADDRESS 5 NOT 0. ERROR
004356      104400      2$:      SCOPE
000006      XADRS=XADRS+1
000006      XCADRS=XCADRS+1
004360      MENT2      †/BC/,†/BYTE COUNT/,\XADRS,\XCADRS,177777,†/177777/
                                ;BYTE COUNT MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 6 OF BYTE COUNT MEMORY
                                ;CAN BE SET TO 177777 AND CLEARED TO 0

004360      TS \XN,100,2$

```

```

004360 012767 000340 173410 T31:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
004366 012767 000100 005006      MOV    #100,ICOUNT       ;SET UP FOR 100 ITERATIONS
004374 012767 004454 004774      MOV    #2#,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
004402 012705 177777      MOV    #177777,R5       ;EXPECTED RESULT=177777
004406 012777 000006 004720      MOV    #6,@DHSCR        ;SELECT LOCATION 6
                                ;OF BYTE COUNT MEMORY
004414 012777 177777 004722      MOV    #177777,@DHBC    ;WRITE 177777 INTO MEMORY
004422 017704 004716      MOV    @DHBC,R4         ;READ CONTENTS OF MEMORY LOCATION
004426 020504      CMP    R5,R4           ;COMPARE EXPECTED AND
004430 001401      BEQ    1#              ;RECEIVED MEMORY CONTENTS
                                .IIF IDN  <BC>,<BA>,      HLT    3              ;BUS ADDRESS MEMORY ERROR
                                .IIF IDN  <BC>,<BC>,      HLT    4              ;BYTE COUNT MEMORY ERROR
004432 104004      EMT    4
004434 005005      CLR    R5              ;EXPECTED RESULT AFTER CLEAR=0
004436 042777 177777 004700      BIC    #177777,@DHBC   ;CLEAR MEMORY LOCATION
004444 017704 004674      MOV    @DHBC,R4        ;READ CONTENTS OF BYTE COUNT
                                ;MEMORY ADDRESS 6
004450 001401      BEQ    2#
                                .IIF IDN  <BC>,<BA>,      HLT    3              ;BUS ADDRESS MEMORY ERROR
004452 104004      .IIF IDN  <BC>,<BC>,      HLT    4              ;BYTE COUNT MEMORY ERROR
                                EMT    4
                                ;ADDRESS 6 NOT 0, ERROR
004454 104400      2#:    SCOPE
                                000007      XADRS=XADRS+1
                                000007      XCADRS=XCADRS+1
004456      MENT2  †/BC/,†/BYTE COUNT/,\XADRS,\XCADRS,177777,†/177777/
                                ;BYTE COUNT MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 7 OF BYTE COUNT MEMORY
                                ;CAN BE SET TO 177777 AND CLEARED TO 0
004456 012767 000340 173312 TS \XN,100,2#
004456 012767 000100 004710 T32:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
004472 012767 004552 004676      MOV    #100,ICOUNT       ;SET UP FOR 100 ITERATIONS
                                MOV    #2#,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
004500 012705 177777      MOV    #177777,R5       ;EXPECTED RESULT=177777
004504 012777 000007 004622      MOV    #7,@DHSCR        ;SELECT LOCATION 7
                                ;OF BYTE COUNT MEMORY
004512 012777 177777 004624      MOV    #177777,@DHBC    ;WRITE 177777 INTO MEMORY
004520 017704 004620      MOV    @DHBC,R4         ;READ CONTENTS OF MEMORY LOCATION
004524 020504      CMP    R5,R4           ;COMPARE EXPECTED AND
004526 001401      BEQ    1#              ;RECEIVED MEMORY CONTENTS
                                .IIF IDN  <BC>,<BA>,      HLT    3              ;BUS ADDRESS MEMORY ERROR
                                .IIF IDN  <BC>,<BC>,      HLT    4              ;BYTE COUNT MEMORY ERROR
004530 104004      EMT    4
004532 005005      CLR    R5              ;EXPECTED RESULT AFTER CLEAR=0
004534 042777 177777 004602      BIC    #177777,@DHBC   ;CLEAR MEMORY LOCATION
004542 017704 004576      MOV    @DHBC,R4        ;READ CONTENTS OF BYTE COUNT
                                ;MEMORY ADDRESS 7

```

```

004546 001401          BEQ      2#
004550          .IIF IDN    <BC>,<BA>,    HLT      3          ;BUS ADDRESS MEMORY ERROR
004550 104004          .IIF IDN    <BC>,<BC>,    HLT      4          ;BYTE COUNT MEMORY ERROR
                                EMT      4
                                ;ADDRESS 7 NOT 0, ERROR
004552 104400          2#:      SCOPE
                                XADRS=XADRS+1
                                XCADRS=XCADRS+1
004554 000010          MEMT2   †/BC/,†/BYTE COUNT/,\XADRS,\XCADRS,177777,†/177777/
                                000010
                                ;BYTE COUNT MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 10 OF BYTE COUNT MEMORY
                                ;CAN BE SET TO 177777 AND CLEARED TO 0

004554          TS \XN,100,2#
004554 012767 000340 173214 T33:  MOV      #340,PS          ;DISABLE ALL INTERRUPTS
004562 012767 000100 004612      MOV      #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
004570 012767 004650 004600      MOV      #2#,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
004576 000034          MOV      #177777,R5        ;EXPECTED RESULT=177777
0C4602 012777 000010 004524      MOV      #10,8DHSCR      ;SELECT LOCATION 10
                                ;OF BYTE COUNT MEMORY
004610 012777 177777 004526      MOV      #177777,8DHBC   ;WRITE 177777 INTO MEMORY
004616 017704 004522          MOV      8DHBC,R4        ;READ CONTENTS OF MEMORY LOCATION
004622 020504          CMP      R5,R4          ;COMPARE EXPECTED AND
004624 001401          BEQ      1#            ;RECEIVED MEMORY CONTENTS
                                .IIF IDN    <BC>,<BA>,    HLT      3          ;BUS ADDRESS MEMORY ERROR
004626          .IIF IDN    <BC>,<BC>,    HLT      4          ;BYTE COUNT MEMORY ERROR
004626 104004          EMT      4
004630 005005          1#:      CLR      R5          ;EXPECTED RESULT AFTER CLEAR=0
004632 042777 177777 004504      BIC      #177777,8DHBC   ;CLEAR MEMORY LOCATION
004640 017704 004500          MOV      8DHBC,R4        ;READ CONTENTS OF BYTE COUNT
                                ;MEMORY ADDRESS 10
004644 001401          BEQ      2#
004646          .IIF IDN    <BC>,<BA>,    HLT      3          ;BUS ADDRESS MEMORY ERROR
004646 104004          .IIF IDN    <BC>,<BC>,    HLT      4          ;BYTE COUNT MEMORY ERROR
                                EMT      4
                                ;ADDRESS 10 NOT 0, ERROR
004650 104400          2#:      SCOPE
                                XADRS=XADRS+1
                                XCADRS=XCADRS+1
004652 000011          MEMT2   †/BC/,†/BYTE COUNT/,\XADRS,\XCADRS,177777,†/177777/
                                000011
                                ;BYTE COUNT MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 11 OF BYTE COUNT MEMORY
                                ;CAN BE SET TO 177777 AND CLEARED TO 0

004652          TS \XN,100,2#
004652 012767 000340 173116 T34:  MOV      #340,PS          ;DISABLE ALL INTERRUPTS
004660 012767 000100 004514      MOV      #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
004666 012767 004746 004502      MOV      #2#,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC

```

```

000035      XN=XN+1
004674 012705 177777      MOV      #177777,R5      ;EXPECTED RESULT=177777
004700 012777 000011 004426  MOV      #11,@DHSCR      ;SELECT LOCATION 11
                                ;OF BYTE COUNT MEMORY
004706 012777 177777 004430  MOV      #177777,@DHBC      ;WRITE 177777 INTO MEMORY
004714 017704 004424      MOV      @DHBC,R4      ;READ CONTENTS OF MEMORY LOCATION
004720 020504      CMP      R5,R4      ;COMPARE EXPECTED AND
004722 001401      BEQ      1$      ;RECEIVED MEMORY CONTENTS

004724      .IIF IDN      <BC>,<BA>,      HLT      3      ;BUS ADDRESS MEMORY ERROR
004724 104004      .IIF IDN      <BC>,<BC>,      HLT      4      ;BYTE COUNT MEMORY ERROR
004726 005005      EMT      4
004730 042777 177777 004406 1$:      CLR      R5      ;EXPECTED RESULT AFTER CLEAR=0
004736 017704 004402      BIC      #177777,@DHBC      ;CLEAR MEMORY LOCATION
                                ;READ CONTENTS OF BYTE COUNT
                                ;MEMORY ADDRESS 11
004742 001401      BEQ      2$

004744      .IIF IDN      <BC>,<BA>,      HLT      3      ;BUS ADDRESS MEMORY ERROR
004744 104004      .IIF IDN      <BC>,<BC>,      HLT      4      ;BYTE COUNT MEMORY ERROR
                                EMT      4
                                ;ADDRESS 11 NOT 0, ERROR

004746 104400      2$:      SCOPE
000012      XADRS=XADRS+1
000012      XCADRS=XCADRS+1
0C4750      MEMT2  +/BC/,+/BYTE COUNT/,\XADRS,\XCADRS,177777,+/177777/

                                ;BYTE COUNT MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 12 OF BYTE COUNT MEMORY
                                ;CAN BE SET TO 177777 AND CLEARED TO 0

004750      TS \XN,100,2$
004750 012767 000340 173020 T35:      MOV      #340,PS      ;DISABLE ALL INTERRUPTS
004756 012767 000100 004416      MOV      #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
004764 012767 005044 004404      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

                                .IF NB      <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      : 3
                                .ENDC
                                XN=XN+1

004772 012705 177777      MOV      #177777,R5      ;EXPECTED RESULT=177777
004776 012777 000012 004330  MOV      #12,@DHSCR      ;SELECT LOCATION 12
                                ;OF BYTE COUNT MEMORY
005004 012777 177777 004332  MOV      #177777,@DHBC      ;WRITE 177777 INTO MEMORY
005012 017704 004326      MOV      @DHBC,R4      ;READ CONTENTS OF MEMORY LOCATION
005016 020504      CMP      R5,R4      ;COMPARE EXPECTED AND
005020 001401      BEQ      1$      ;RECEIVED MEMORY CONTENTS

005022      .IIF IDN      <BC>,<BA>,      HLT      3      ;BUS ADDRESS MEMORY ERROR
005022 104004      .IIF IDN      <BC>,<BC>,      HLT      4      ;BYTE COUNT MEMORY ERROR
005024 005005      EMT      4
005026 042777 177777 004310 1$:      CLR      R5      ;EXPECTED RESULT AFTER CLEAR=0
005034 017704 004304      BIC      #177777,@DHBC      ;CLEAR MEMORY LOCATION
                                ;READ CONTENTS OF BYTE COUNT
                                ;MEMORY ADDRESS 12
005040 001401      BEQ      2$

005042      .IIF IDN      <BC>,<BA>,      HLT      3      ;BUS ADDRESS MEMORY ERROR
005042 104004      .IIF IDN      <BC>,<BC>,      HLT      4      ;BYTE COUNT MEMORY ERROR
                                EMT      4
                                ;ADDRESS 12 NOT 0, ERROR

005044 104400      2$:      SCOPE

```

```

000013          XADRS=XADRS+1
000013          XCADRS=XCADRS+1
005046          MEMENT2  +/BC/,+/BYTE COUNT/,\XADRS,\XCADRS,177777,+/177777/

                ;BYTE COUNT MEMORY DATA TEST
                ;VERIFY THAT ADDRESS 13 OF BYTE COUNT MEMORY
                ;CAN BE SET TO 177777 AND CLEARED TO 0

005046          TS \XN,100,2$
005046 012767 000340 172722 T36:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
005054 012767 000100 004320      MOV    #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
005062 012767 005142 004306      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                .IF NB  <>
                MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                .ENDC
                XN=XN+1
005070          MOV    #177777,R5        ;EXPECTED RESULT=177777
005074 012777 000013 004232      MOV    #13,@DHSCR      ;SELECT LOCATION 13
                ;OF BYTE COUNT MEMORY
005102 012777 177777 004234      MOV    #177777,@DHBC   ;WRITE 177777 INTO MEMORY
005110 017704 004230              MOV    @DHBC,R4        ;READ CONTENTS OF MEMORY LOCATION
005114 020504                    CMP    R5,R4          ;COMPARE EXPECTED AND
005116 001401                    BEQ    1$              ;RECEIVED MEMORY CONTENTS
                .IIF IDN  <BC>,<BA>,    HLT    3              ;BUS ADDRESS MEMORY ERROR
                .IIF IDN  <BC>,<BC>,    HLT    4              ;BYTE COUNT MEMORY ERROR
005120          EMT    4
005122 104004                    1$:  CLR    R5              ;EXPECTED RESULT AFTER CLEAR=0
005124 042777 177777 004212      BIC    #177777,@DHBC  ;CLEAR MEMORY LOCATION
005132 017704 004206              MOV    @DHBC,R4        ;READ CONTENTS OF BYTE COUNT
                ;MEMORY ADDRESS 13
005136 001401                    BEQ    2$
                .IIF IDN  <BC>,<BA>,    HLT    3              ;BUS ADDRESS MEMORY ERROR
                .IIF IDN  <BC>,<BC>,    HLT    4              ;BYTE COUNT MEMORY ERROR
005140          EMT    4
005142 104400                    2$:  SCOPE
000014          XADRS=XADRS+1
000014          XCADRS=XCADRS+1
005144          MEMENT2  +/BC/,+/BYTE COUNT/,\XADRS,\XCADRS,177777,+/177777/

                ;BYTE COUNT MEMORY DATA TEST
                ;VERIFY THAT ADDRESS 14 OF BYTE COUNT MEMORY
                ;CAN BE SET TO 177777 AND CLEARED TO 0

005144          TS \XN,100,2$
005144 012767 000340 172624 T37:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
005152 012767 000100 004222      MOV    #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
005160 012767 005240 004210      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                .IF NB  <>
                MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                .ENDC
                XN=XN+1
005166          MOV    #177777,R5        ;EXPECTED RESULT=177777
005172 012777 000014 004134      MOV    #14,@DHSCR      ;SELECT LOCATION 14
                ;OF BYTE COUNT MEMORY
005200 012777 177777 004136      MOV    #177777,@DHBC   ;WRITE 177777 INTO MEMORY
005206 017704 004132              MOV    @DHBC,R4        ;READ CONTENTS OF MEMORY LOCATION

```

```

005212 020504          CMP      R5,R4          ;COMPARE EXPECTED AND
005214 001401          BEQ      1$              ;RECEIVED MEMORY CONTENTS
                                .IIF IDN      <BC>,<BA>,      HLT      3              ;BUS ADDRESS MEMORY ERROR
                                .IIF IDN      <BC>,<BC>,      HLT      4              ;BYTE COUNT MEMORY ERROR
005216          EMT      4
005220 104004          CLR      R5              ;EXPECTED RESULT AFTER CLEAR=0
005222 042777 177777 004114 BIC      @177777,@DHBC  ;CLEAR MEMORY LOCATION
005230 017704 004110  MOV      @DHBC,R4      ;READ CONTENTS OF BYTE COUNT
                                BEQ      2$              ;MEMORY ADDRESS 14
005234 001401          .IIF IDN      <BC>,<BA>,      HLT      3              ;BUS ADDRESS MEMORY ERROR
                                .IIF IDN      <BC>,<BC>,      HLT      4              ;BYTE COUNT MEMORY ERROR
                                EMT      4
                                ;ADDRESS 14 NOT 0, ERROR
005240 104400          2$:      SCOPE
                                000015      XADRS=XADRS+1
                                000015      XCADRS=XCADRS+1
005242          MENT2   ↑/BC/,↑/BYTE COUNT/,\XADRS,\XCADRS,177777,↑/177777/
                                ;BYTE COUNT MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 15 OF BYTE COUNT MEMORY
                                ;CAN BE SET TO 177777 AND CLEARED TO 0

005242          TS      \XN,100,2$
005242 012767 000340 172526 T40:    MOV      @340,PS      ;DISABLE ALL INTERRUPTS
005250 012767 000100 004124      MOV      @100,ICOUNT    ;SET UP FOR 100 ITERATIONS
005256 012767 005336 004112      MOV      @2$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB      <>
                                MOV      @,FREEZ1        ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
005264 012705 177777          MOV      @177777,R5      ;EXPECTED RESULT=177777
005270 012777 000015 004036      MOV      @15,@DHSCR    ;SELECT LOCATION 15
                                MOV      @177777,@DHBC    ;WRITE 177777 INTO MEMORY
                                MOV      @DHBC,R4        ;READ CONTENTS OF MEMORY LOCATION
005276 012777 177777 004040      MOV      @177777,@DHBC ;COMPARE EXPECTED AND
005304 017704 004034          CMP      R5,R4          ;RECEIVED MEMORY CONTENTS
005310 020504          BEQ      1$              ;BUS ADDRESS MEMORY ERROR
005312 001401          .IIF IDN      <BC>,<BA>,      HLT      3              ;BYTE COUNT MEMORY ERROR
                                .IIF IDN      <BC>,<BC>,      HLT      4
                                EMT      4
005314 104004          CLR      R5              ;EXPECTED RESULT AFTER CLEAR=0
005316 005005          BIC      @177777,@DHBC ;CLEAR MEMORY LOCATION
005320 042777 177777 004016      MOV      @DHBC,R4      ;READ CONTENTS OF BYTE COUNT
005326 017704 004012          BEQ      2$              ;MEMORY ADDRESS 15
005332 001401          .IIF IDN      <BC>,<BA>,      HLT      3              ;BUS ADDRESS MEMORY ERROR
                                .IIF IDN      <BC>,<BC>,      HLT      4              ;BYTE COUNT MEMORY ERROR
                                EMT      4
                                ;ADDRESS 15 NOT 0, ERROR
005336 104400          2$:      SCOPE
                                000016      XADRS=XADRS+1
                                000016      XCADRS=XCADRS+1
005340          MENT2   ↑/BC/,↑/BYTE COUNT/,\XADRS,\XCADRS,177777,↑/177777/
                                ;BYTE COUNT MEMORY DATA TEST
                                ;VERIFY THAT ADDRESS 16 OF BYTE COUNT MEMORY

```

;CAN BE SET TO 17777 AND CLEARED TO 0

```

005340      000042      TS \XN,100,2#
005340 012767 000340 172430 T41:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
005346 012767 000100 004026      MOV    #100,ICOUNT       ;SET UP FOR 100 ITERATIONS
005354 012767 005434 004014      MOV    #2#,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
005362      000042      MOV    #177777,R5       ;EXPECTED RESULT=177777
005366 012705 177777      MOV    #16,@DHSCR       ;SELECT LOCATION 16
                                ;OF BYTE COUNT MEMORY
005374 012777 177777 003740      MOV    #177777,@DHBC    ;WRITE 177777 INTO MEMORY
005402 017704 003736      MOV    @DHBC,R4         ;READ CONTENTS OF MEMORY LOCATION
005406 020504      CMP    R5,R4           ;COMPARE EXPECTED AND
005410 001401      BEQ    1#              ;RECEIVED MEMORY CONTENTS
                                .IIF IDN  <BC>,<BA>,      HLT    3              ;BUS ADDRESS MEMORY ERROR
                                .IIF IDN  <BC>,<BC>,      HLT    4              ;BYTE COUNT MEMORY ERROR
005412      104004      EMT    4
005414 005005      1#:    CLR    R5         ;EXPECTED RESULT AFTER CLEAR=0
005416 042777 177777 003720      BIC    #177777,@DHBC   ;CLEAR MEMORY LOCATION
005424 017704 003714      MOV    @DHBC,R4        ;READ CONTENTS OF BYTE COUNT
                                ;MEMORY ADDRESS 16
005430 001401      BEQ    2#
                                .IIF IDN  <BC>,<BA>,      HLT    3              ;BUS ADDRESS MEMORY ERROR
005432      104004      .IIF IDN  <BC>,<BC>,      HLT    4              ;BYTE COUNT MEMORY ERROR
                                EMT    4
                                ;ADDRESS 16 NOT 0, ERROR
005434 104400      2#:    SCOPE
                                000017      XADRS=XADRS+1
                                000017      XCADRS=XCADRS+1
005436      000017      MENT2  †/BC/,†/BYTE COUNT/,\XADRS,\XCADRS,177777,†/177777/

```

```

;BYTE COUNT MEMORY DATA TEST
;VERIFY THAT ADDRESS 17 OF BYTE COUNT MEMORY
;CAN BE SET TO 17777 AND CLEARED TO 0

```

```

005436      000043      TS \XN,100,2#
005436 012767 000340 172332 T42:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
005444 012767 000100 003730      MOV    #100,ICOUNT       ;SET UP FOR 100 ITERATIONS
005452 012767 005532 003716      MOV    #2#,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
005460      000043      MOV    #177777,R5       ;EXPECTED RESULT=177777
005464 012777 000017 003642      MOV    #17,@DHSCR       ;SELECT LOCATION 17
                                ;OF BYTE COUNT MEMORY
005472 012777 177777 003644      MOV    #177777,@DHBC    ;WRITE 177777 INTO MEMORY
005500 017704 003640      MOV    @DHBC,R4         ;READ CONTENTS OF MEMORY LOCATION
005504 020504      CMP    R5,R4           ;COMPARE EXPECTED AND
005506 001401      BEQ    1#              ;RECEIVED MEMORY CONTENTS
                                .IIF IDN  <BC>,<BA>,      HLT    3              ;BUS ADDRESS MEMORY ERROR
                                .IIF IDN  <BC>,<BC>,      HLT    4              ;BYTE COUNT MEMORY ERROR
005510      104004      EMT    4
005512 005005      1#:    CLR    R5         ;EXPECTED RESULT AFTER CLEAR=0

```



```

005664 104003
005666 104410
005670 005277 003440
005674 005301
005676 001360
005700 005203
005702 001300
005704 001332
005706 104400
31 005710

      EMT      3
      .IIF IDN <BA>,<BC>, HLT 4 ;BYTE COUNT MEMORY ERROR
5$:   SCOPE1   ;CHECK FOR LOOP WITH CURRENT DATA
      INC     @DHSCR ;CHECK CONTENTS OF NEXT LOCATION
      DEC     R1
      BNE     3$
      INC     R3      ;NEXT ADDRESS TO BE WRITTEN
      DEC     R0
      BNE     1$
6$:   SCOPE   ;CHECK FOR ITERATIONS, LOOP
MEMT3 +/BA/,+/BUS ADDRESS/,125252,+/125252/

      ;BUS ADDRESS MEMORY TEST
      ;CLEAR ALL LOCATIONS IN BUS ADDRESS MEMORY
      ;SET SELECTED LOCATION TO VALUE 125252
      ;VERIFY THAT SELECTED LOCATION WAS SET
      ;TO 125252.
      ;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.

005710
005710 012767 000340 172060 TS \XN,100,6$,2$
005716 012767 000100 003456 T44:  MOV     @340,PS ;DISABLE ALL INTERRUPTS
005724 012767 006062 003444      MOV     @100,ICOUNT ;SET UP FOR 100 ITERATIONS
      MOV     @6$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <2$>
005732 012767 005756 003440      MOV     @2$,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
      .ENDC
      XN=XN+1
005740 012700 000020      MOV     @20,R0 ;SET UP TO TEST 20(OCTAL)
      ;LOCATIONS IN BUS ADDRESS MEMORY
005744 005003      CLR     R3 ;FIRST LOCATION TO BE
      ;WRITTEN INTO IS 0
005746 012701 000020 1$:  MOV     @20,R1 ;SET UP TO CLEAR 20 (OCTAL)
      ;LOCATIONS IN BUS ADDRESS MEMORY
005752 005077 003356      CLR     @DHSCR ;START AT LOCATION 0
005756 005077 003360 2$:  CLR     @DHBA ;CLEAR LOCATION IN
      ;BUS ADDRESS MEMORY
005762 005277 003346      INC     @DHSCR ;ADVANCE TO NEXT LOCATION
005766 005301      DEC     R1 ;CONTINUE CLEARING
005770 001372      BNE     2$ ;IF NOT DONE
005772 010377 003336      MOV     R3,@DHSCR ;SELECT ADDRESS TO BE TESTED
005776 012777 125252 003336      MOV     @125252,@DHBA ;WRITE 125252 INTO LOCATION
006004 005077 003324      CLR     @DHSCR ;ADDRESS LOCATION 0
006010 012701 000020      MOV     @20,R1 ;SET UP TO CHECK ALL ADDRESSES
      ;IN BUS ADDRESS MEMORY
006014 012705 125252 3$:  MOV     @125252,R5 ;125252-EXPECTED RESULT
      ;IF ADDRESS READ IS LOCATION
      ;WRITTEN INTO
006020 017704 003316      MOV     @DHBA,R4 ;READ MEMORY LOCATION
006024 027703 003304      CMP     @DHSCR,R3 ;IF LINE NUMBER=ADDRESS
      ;OF LOCATION WRITTEN INTO
      ;EXPECTED CONTENTS=125252
006030 001401      BEQ     4$
006032 005005      CLR     R5
006034 020504 4$:  CMP     R5,R4 ;OTHERWISE, EXPECTED RESULTS=0
006036 001401      BEQ     5$ ;DOES MEMORY LOCATION CONTAIN
006040      .IIF IDN <BA>,<BA>, HLT 3 ;EXPECTED RESULT
006040 104003      EMT     3 ;BUS ADDRESS MEMORY ERROR

```

```

00,042 104410          .IIF IDN      <BA>,<BC>,      HLT      4          ;BYTE COUNT MEMORY ERROR
006044 005277 003264 5$: SCOPE1          ;CHECK FOR LOOP WITH CURRENT DATA
006050 005301          INC      @DHSCR          ;CHECK CONTENTS OF NEXT LOCATION
006052 001360          DEC      R1
006054 005203          BNE     3$
006056 005300          INC      R3          ;NEXT ADDRESS TO BE WRITTEN
006060 001332          DEC      R0
006062 104400          BNE     1$
32 006064 32 MENT3      6$: SCOPE          ;CHECK FOR ITERATIONS, LOOP
          +/BA/,+/BUS ADDRESS/,52525,+/52525/
          ;BUS ADDRESS MEMORY TEST
          ;CLEAR ALL LOCATIONS IN BUS ADDRESS MEMORY
          ;SET SELECTED LOCATION TO VALUE 52525
          ;VERIFY THAT SELECTED LOCATION WAS SET
          ;TO 52525.
          ;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.

006064          TS \XN,100,6$,2$
006064 012767 000340 171704 T45: MOV      @340,PS          ;DISABLE ALL INTERRUPTS
006072 012767 000100 003302      MOV      @100,ICOUNT      ;SET UP FOR 100 ITERATIONS
006100 012767 006236 003270      MOV      @6$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

006106 012767 006132 003264 .IF NB <2$>
          MOV      @2$,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
          .ENDC
          XN=XN+1
006114 012700 000020      MOV      @20,R0          ;SET UP TO TEST 20(OCTAL)
          ;LOCATIONS IN BUS ADDRESS MEMORY
006120 005003          CLR      R3          ;FIRST LOCATION TO BE
          ;WRITTEN INTO IS 0
006122 012701 000020      1$: MOV      @20,R1          ;SET UP TO CLEAR 20 (OCTAL)
          ;LOCATIONS IN BUS ADDRESS MEMORY
006126 005077 003202      CLR      @DHSCR          ;START AT LOCATION 0
006132 005077 003204      2$: CLR      @DHBA          ;CLEAR LOCATION IN
          ;BUS ADDRESS MEMORY
006136 005277 003172          INC      @DHSCR          ;ADVANCE TO NEXT LOCATION
006142 005301          DEC      R1          ;CONTINUE CLEARING
006144 001372          BNE     2$          ;IF NOT DONE
006146 010377 003162          MOV      R3,@DHSCR          ;SELECT ADDRESS TO BE TESTED
006152 012777 052525 003162          MOV      @52525,@DHBA      ;WRITE 52525 INTO LOCATION
006160 005077 003150          CLR      @DHSCR          ;ADDRESS LOCATION 0
006164 012701 000020          MOV      @20,R1          ;SET UP TO CHECK ALL ADDRESSES
          ;IN BUS ADDRESS MEMORY
006170 012705 052525      3$: MOV      @52525,R5          ;52525-EXPECTED RESULT
          ;IF ADDRESS READ IS LOCATION
          ;WRITTEN INTO
006174 017704 003142          MOV      @DHBA,R4          ;READ MEMORY LOCATION
006200 027703 003130          CMP      @DHSCR,R3          ;IF LINE NUMBER=ADDRESS
          ;OF LOCATION WRITTEN INTO
          ;EXPECTED CONTENTS=52525
006204 001401          BEQ     4$
006206 005005          CLR      R5
006210 020504      4$: CMP      R5,R4          ;OTHERWISE, EXPECTED RESULTS=0
006212 001401          BEQ     5$          ;DOES MEMORY LOCATION CONTAIN
006214          .IIF IDN      <BA>,<BA>,      HLT      3          ;EXPECTED RESULT
          ;BUS ADDRESS MEMORY ERROR
006214 104003          EMT      3
          .IIF IDN      <BA>,<BC>,      HLT      4          ;BYTE COUNT MEMORY ERROR

```

```

006216 104410          5$: SCOPE1          ;CHECK FOR LOOP WITH CURRENT DATA
006220 005277 003110  INC      @DHSCR      ;CHECK CONTENTS OF NEXT LOCATION
006224 005301          DEC      R1
006226 001360          BNE      3$
006230 005203          INC      R3          ;NEXT ADDRESS TO BE WRITTEN
006232 005300          DEC      R0
006234 001332          BNE      1$
006236 104400          6$: SCOPE          ;CHECK FOR ITERATIONS, LOOP
33 006240 MEMT3    †/BC/,†/BYTE COUNT/,177777,†//177777/

;BYTE COUNT MEMORY TEST
;CLEAR ALL LOCATIONS IN BYTE COUNT MEMORY
;SET SELECTED LOCATION TO VALUE 177777
;VERIFY THAT SELECTED LOCATION WAS SET
;TO 177777.
;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.

006240          TS \XN,100,6$,2$
006240 012767 000340 171530 T46:  MOV      @340,PS          ;DISABLE ALL INTERRUPTS
006246 012767 000100 003126  MOV      @100,ICOUNT      ;SET UP FOR 100 ITERATIONS
006254 012767 006412 003114  MOV      @6$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

;IF NB <2$>
006262 012767 006306 003110  MOV      @2$,FREEZ1      ;SET UP TO LOOP WITH DATA          ; 3
;ENDC
;XN=XN+1
006270 012700 000020          MOV      @20,R0          ;SET UP TO TEST 20(OCTAL)
006274 005003          CLR      R3          ;LOCATIONS IN BYTE COUNT MEMORY
006276 012701 000020          1$: MOV      @20,R1          ;FIRST LOCATION TO BE
;WRITTEN INTO IS 0
;SET UP TO CLEAR 20 (OCTAL)
;LOCATIONS IN BYTE COUNT MEMORY
006302 005077 003026          CLR      @DHSCR          ;START AT LOCATION 0
006306 005077 003032          2$: CLR      @DHBC          ;CLEAR LOCATION IN
;BYTE COUNT MEMORY
006312 005277 003010          INC      @DHSCR          ;ADVANCE TO NEXT LOCATION
006316 005301          DEC      R1          ;CONTINUE CLEARING
006320 001372          BNE      2$          ;IF NOT DONE
006322 010377 003006          MOV      R3,@DHSCR      ;SELECT ADDRESS TO BE TESTED
006326 012777 177777 003010  MOV      @177777,@DHBC  ;WRITE 177777 INTO LOCATION
006334 005077 002774          CLR      @DHSCR          ;ADDRESS LOCATION 0
006340 012701 000020          MOV      @20,R1          ;SET UP TO CHECK ALL ADDRESSES
;IN BYTE COUNT MEMORY
006344 012705 177777          3$: MOV      @177777,R5  ;177777-EXPECTED RESULT
;IF ADDRESS READ IS LOCATION
;WRITTEN INTO
;READ MEMORY LOCATION
006350 017704 002770          MOV      @DHBC,R4          ;IF LINE NUMBER=ADDRESS
006354 027703 002754          CMP      @DHSCR,R3        ;OF LOCATION WRITTEN INTO
;EXPECTED CONTENTS=177777
;OTHERWISE, EXPECTED RESULTS=0
;DOES MEMORY LOCATION CONTAIN
;EXPECTED RESULT
006360 001401          BEQ      4$
006362 005005          CLR      R5
006364 020504          4$: CMP      R5,R4
006366 001401          BEQ      5$          ;BUS ADDRESS MEMORY ERROR
;BYTE COUNT MEMORY ERROR
006370          .IIF IDN <BC>,<BA>, HLT 3
006370          .IIF IDN <BC>,<BC>, HLT 4
006372 104410          5$: EMT      4
SCOPE1          ;CHECK FOR LOOP WITH CURRENT DATA

```

```

006374 005277 002734      INC      @DHSCR      ;CHECK CONTENTS OF NEXT LOCATION
006400 005301              DEC      R1
006402 001360              BNE     3$
006404 005203              INC      R3      ;NEXT ADDRESS TO BE WRITTEN
006406 005300              DEC      R0
006410 001332              BNE     1$
006412 104400              6$: SCOPE      ;CHECK FOR ITERATIONS, LOOP
34 006414 MEMT3  ↑/BC/,↑/BYTE COUNT/,125252,↑/125252/

;BYTE COUNT MEMORY TEST
;CLEAR ALL LOCATIONS IN BYTE COUNT MEMORY
;SET SELECTED LOCATION TO VALUE 125252
;VERIFY THAT SELECTED LOCATION WAS SET
;TO 125252.
;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.

006414      TS \XN,100,6$,2$
006414 012767 000340 171354 T47: MOV      #340,PS      ;DISABLE ALL INTERRUPTS
006422 012767 000100 002752 MOV      #100,ICOUNT    ;SET UP FOR 100 ITERATIONS
006430 012767 006566 002740 MOV      #6$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
. IF NB <2$>
006436 012767 006462 002734 MOV      #2$,FREEZ1     ;SET UP TO LOOP WITH DATA      ; 3
.ENDC
XN=XN+1
006444 000050 000020      MOV      #20,R0      ;SET UP TO TEST 20(OCTAL)
006450 005003              CLR      R3      ;LOCATIONS IN BYTE COUNT MEMORY
;FIRST LOCATION TO BE
;WRITTEN INTO IS 0
006452 012701 000020      1$: MOV      #20,R1    ;SET UP TO CLEAR 20 (OCTAL)
006456 005077 002652      CLR      @DHSCR      ;LOCATIONS IN BYTE COUNT MEMORY
006462 005077 002656      2$: CLR      @DHBC      ;START AT LOCATION 0
;CLEAR LOCATION IN
;BYTE COUNT MEMORY
;ADVANCE TO NEXT LOCATION
006466 005277 002642      INC      @DHSCR      ;CONTINUE CLEARING
006472 005301              DEC      R1
006474 001372              BNE     2$          ;IF NOT DONE
006476 010377 002632      MOV      R3,@DHSCR    ;SELECT ADDRESS TO BE TESTED
006502 012777 125252 002634 MOV      #125252,@DHBC ;WRITE 125252 INTO LOCATION
006510 005077 002620      CLR      @DHSCR      ;ADDRESS LOCATION 0
006514 012701 000020      MOV      #20,R1      ;SET UP TO CHECK ALL ADDRESSES
006520 012705 125252      3$: MOV      #125252,R5 ;IN BYTE COUNT MEMORY
;125252=EXPECTED RESULT
;IF ADDRESS READ IS LOCATION
;WRITTEN INTO
006524 017704 002614      MOV      @DHBC,R4     ;READ MEMORY LOCATION
006530 027703 002600      CMP      @DHSCR,R3    ;IF LINE NUMBER=ADDRESS
;OF LOCATION WRITTEN INTO
;EXPECTED CONTENTS=125252
;OTHERWISE, EXPECTED RESULTS=0
;DOES MEMORY LOCATION CONTAIN
;EXPECTED RESULT
006534 001401              BEQ     4$
006536 005005              CLR      R5
006540 020504      4$: CMP      R5,R4
006542 001401              BEQ     5$
.IIF IDN <BC>,<BA>,      HLT     3      ;BUS ADDRESS MEMORY ERROR
.IIF IDN <BC>,<BC>,      HLT     4      ;BYTE COUNT MEMORY ERROR
006544 104004              5$: EMT
006546 104410              SCOPE1
006550 005277 002560      INC      @DHSCR      ;CHECK FOR LOOP WITH CURRENT DATA
;CHECK CONTENTS OF NEXT LOCATION

```



```

007112 005203          INC      R3          ;NEXT ADDRESS TO BE WRITTEN
007114 005300          DEC      R0
007116 001331          BNE     1$
007120 104400          6$: SCOPE          ;CHECK FOR ITERATIONS, LOOP
37 007122 MEMT4  +/BC/,+/BYTE COUNT/,0,+/0/

;BYTE COUNT MEMORY TEST
;SET ALL LOCATIONS IN BYTE COUNT MEMORY TO 177777
;SET SELECTED LOCATION TO VALUE 0
;VERIFY THAT SELECTED LOCATION WAS SET
;TO 0.
;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.

007122          TS \XN,100,6$,2$
007122 012767 000340 170646 T52:  MOV     #340,PS          ;DISABLE ALL INTERRUPTS
007130 012767 000100 002244      MOV     #100,ICOUNT       ;SET UP FOR 100 ITERATIONS
007136 012767 007276 002232      MOV     #6$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <2$>
007144 012767 007170 002226      MOV     #2$,FREEZ1       ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
007152 012700 000020      MOV     #20,R0          ;SET UP TO TEST 20(OCTAL)
007156 005003          CLR     R3              ;LOCATIONS IN BYTE COUNT MEMORY
                                ;FIRST LOCATION TO BE
                                ;WRITTEN INTO IS 0
007160 012701 000020      1$: MOV     #20,R1          ;SET UP TO SET 20 (OCTAL)
                                ;LOCATIONS IN BYTE COUNT MEMORY TO 177777
007164 005077 002144      CLR     @DHSCR          ;START AT LOCATION 0
007170 012777 177777 002146 2$: MOV     @177777,@DHBC       ;SET LOCATION IN
                                ;BYTE COUNT MEMORY
                                ;ADVANCE TO NEXT LOCATION
007176 005277 002132      INC     @DHSCR          ;CONTINUE SETTING
007202 005301          DEC     R1              ;IF NOT DONE
007204 001371          BNE     2$              ;SELECT ADDRESS TO BE TESTED
007206 010377 002122      MOV     R3,@DHSCR       ;WRITE 0 INTO LOCATION
007212 012777 000000 002124      MOV     #0,@DHBC        ;ADDRESS LOCATION 0
007220 005077 002110      CLR     @DHSCR          ;SET UP TO CHECK ALL ADDRESSES
007224 012701 000020      MOV     #20,R1          ;IN BYTE COUNT MEMORY
007230 012705 000000      3$: MOV     #0,R5          ;0-EXPECTED RESULT
                                ;IF ADDRESS READ IS LOCATION
                                ;WRITTEN INTO
007234 017704 002104      MOV     @DHBC,R4        ;READ MEMORY LOCATION
007240 027703 002070      CMP     @DHSCR,R3       ;IF LINE NUMBER=ADDRESS
                                ;OF LOCATION WRITTEN INTO
                                ;EXPECTED CONTENTS=0
007244 001401          BEQ     4$
007246 005105          COM     R5
007250 020504      4$: CMP     R5,R4          ;OTHERWISE, EXPECTED RESULTS=177777
007252 001401          BEQ     5$              ;DOES MEMORY LOCATION CONTAIN
                                ;EXPECTED RESULT
007254          .IIF IDN          HLT     3          ;BUS ADDRESS MEMORY ERROR
007254 104004          .IIF IDN          HLT     4          ;BYTE COUNT MEMORY ERROR
007256 104410          EMT
007260 005277 002050      5$: SCOPE1
007264 005301          INC     @DHSCR
007266 001360          DEC     R1
007270 005203          BNE     3$
                                INC     R3          ;NEXT ADDRESS TO BE WRITTEN

```

```

007272 005300          DEC      R0
007274 001331          BNE      1$
007276 104400          6$:      SCOPE
38 007300          MXTST2  ↑/LOW AND HIGH/,60,300          ;CHECK FOR ITERATIONS, LOOP

;MEMORY EXTENSION MEMORY TEST
;VERIFY THAT LOW AND HIGH ORDER MEMORY EXTENSION BIT CAN BE
;SET AND CLEARED IN SELECTED MEMORY EXTENSION MEMORY LOCATION
007300          TS \XN,100,6$,2$
007300 012767 000340 170470 T53:      MOV      #340,PS          ;DISABLE ALL INTERRUPTS
007306 012767 000100 002066      MOV      #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
007314 012767 007470 002054      MOV      #6$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

007322 012767 007346 002050      .IF NB  <2$>
                                MOV      #2$,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
007330 012700 000020          MOV      #20,R0          ;SET UP TO TEST 20(OCTAL)
007334 005003          CLR      R3          ;LOCATIONS IN MEMORY EXTENSION MEMORY
                                ;FIRST LOCATION TO BE
                                ;WRITEEN INTO IS 0
007336 012701 000020          1$:      MOV      #20,R1          ;SET UP TO CLEAR 20 (OCTAL)
                                ;LOCATIONS IN MEMORY EXTENSION MEMORY
007342 005077 001766          CLR      @DHSCR          ;START AT LOCATION 0
007346 042777 000060 001760 2$:      BIC      #60,@DHSCR      ;CLEAR LOCATION IN
007354 012777 000000 001760      MOV      #0,@DHBA      ;MEMORY EXTENSION MEMORY
007362 005277 001746          INC      @DHSCR          ;ADVANCE TO NEXT LOCATION
007366 005301          DEC      R1          ;CONTINUE CLEARING
007370 001366          BNE      2$          ;IF NOT DONE
007372 010377 001736          MOV      R3,@DHSCR      ;SELECT ADDRESS TO BE TESTED
007376 052777 000060 001730      BIS      #60,@DHSCR      ;WRITE LOW AND HIGH INTO LOCATION
007404 012777 000000 001730      MOV      #0,@DHBA      ;LOAD ADDRESS
007412 005077 001716          CLR      @DHSCR          ;ADDRESS LOCATION 0
007416 012701 000020          MOV      #20,R1          ;SET UP TO CHECK ALL ADDRESSES
                                ;IN MEMORY EXTENSION MEMORY
007422 012705 000300          3$:      MOV      #300,R5          ;LOW AND HIGH=EXPECTED RESULT
                                ;IF ADDRESS READ IS LOCATION
                                ;WRITTEN INTO
007426 017704 001720          MOV      @DHSSR,R4      ;READ MEMORY LOCATION
007432 027703 001676          CMP      @DHSCR,R3      ;IF LINE NUMBER=ADDRESS
                                ;OF LOCATION WRITTEN INTO
                                ;EXPECTED CONTENTS=LOW AND HIGH

007436 001401          BEQ      4$
007440 005005          CLR      R5          ;OTHERWISE, EXPECTED RESULTS=0
007442 020504          4$:      CMP      R5,R4          ;DOES MEMORY LOCATION CONTAIN
007444 001401          BEQ      5$          ;EXPECTED RESULT
007446          HLT      5          ;MEMORY EXTENSION DATA ERROR
007446 104005          EMT      5
007450 104410          5$:      SCOPE1
007452 005277 001656          INC      @DHSCR          ;CHECK FOR LOOP WITH CURRENT DATA
007456 005301          DEC      R1          ;CHECK CONTENTS OF NEXT LOCATION
007460 001360          BNE      3$
007462 005203          INC      R3          ;NEXT ADDRESS TO BE WRITTEN
007464 005300          DEC      R0
007466 001323          BNE      1$
007470 104400          6$:      SCOPE
39 007472          MXTST2  ↑/LOW/,20,100          ;CHECK FOR ITERATIONS, LOOP

```

```

;MEMORY EXTENSION MEMORY TEST
;VERIFY THAT LOW ORDER MEMORY EXTENSION BIT CAN BE
;SET AND CLEARED IN SELECTED MEMORY EXTENSION MEMORY LOCATION
007472 TS \XN,100,6$,2$
007472 012767 000340 170276 T54: MOV #340,PS ;DISABLE ALL INTERRUPTS
007500 012767 000100 001674 MOV #100,ICOUNT ;SET UP FOR 100 ITERATIONS
007506 012767 007062 001662 MOV #6$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST

007514 012767 007540 001656 .IF NB <2$> MOV #2$,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1
007522 012700 000020 MOV #20,R0 ;SET UP TO TEST 20(OCTAL)
007526 005003 CLR R3 ;LOCATIONS IN MEMORY EXTENSION MEMORY
;FIRST LOCATION TO BE
;WRITEEN INTO IS 0
007530 012701 000020 1$: MOV #20,R1 ;SET UP TO CLEAR 20 (OCTAL)
;LOCATIONS IN MEMORY EXTENSION MEMORY
;START AT LOCATION 0
007534 005077 001574 CLR @DHSCR ;CLEAR LOCATION IN
007540 042777 000060 001566 2$: BIC #60,@DHSCR ;MEMORY EXTENSION MEMORY
007546 012777 000000 001566 MOV #0,@DHBA ;ADVANCE TO NEXT LOCATION
007554 005277 001554 INC @DHSCR ;CONTINUE CLEARING
007560 005301 DEC R1 ;IF NOT DONE
007562 001366 BNE 2$ ;SELECT ADDRESS TO BE TESTED
007564 010377 001544 MOV R3,@DHSCR ;WRITE LOW INTO LOCATION
007570 052777 000020 001536 BIS #20,@DHSCR ;LOAD ADDRESS
007576 012777 000000 001536 MOV #0,@DHBA ;ADDRESS LOCATION 0
007604 005077 001524 CLR @DHSCR ;SET UP TO CHECK ALL ADDRESSES
007610 012701 000020 MOV #20,R1 ;IN MEMORY EXTENSION MEMORY
;LOW-EXPECTED RESULT
007614 012705 000100 3$: MOV #100,R5 ;IF ADDRESS READ IS LOCATION
;WRITTEN INTO
;READ MEMORY LOCATION
;IF LINE NUMBER-ADDRESS
;OF LOCATION WRITTEN INTO
;EXPECTED CONTENTS=LOW
007620 017704 001526 MOV @DHSSR,R4
007624 027703 001504 CMP @DHSCR,R3
;OTHERWISE, EXPECTED RESULTS=0
;DOES MEMORY LOCATION CONTAIN
;EXPECTED RESULT
;MEMORY EXTENSION DATA ERROR
007630 001401 BEQ 4$
007632 005005 CLR R5
007634 020504 4$: CMP R5,R4
007636 001401 BEQ 5$
007640 HLT 5
007640 104005 EMT 5
007642 104410 5$: SCOPE1 ;CHECK FOR LOOP WITH CURRENT DATA
007644 005277 001464 INC @DHSCR ;CHECK CONTENTS OF NEXT LOCATION
007650 005301 DEC R1
007652 001360 BNE 3$
007654 005203 INC R3 ;NEXT ADDRESS TO BE WRITTEN
007656 005300 DEC R0
007660 001323 BNE 1$
007662 104400 6$: SCOPE ;CHECK FOR ITERATIONS, LOOP
40 007664 MXTST2 †/HIGH/,40,200

;MEMORY EXTENSION MEMORY TEST
;VERIFY THAT HIGH ORDER MEMORY EXTENSION BIT CAN BE
;SET AND CLEARED IN SELECTED MEMORY EXTENSION MEMORY LOCATION
007664 TS \XN,100,6$,2$
007664 012767 000340 170104 T55: MOV #340,PS ;DISABLE ALL INTERRUPTS

```


1
2 010056

.EOP +/BEGIN/
;END OF PASS
;TYPE NAME OF TEST
;UPDATE PASS COUNT
;CHECK FOR EXIT TO ACT-11
;RESTART TEST

010056	104401			EOP:	TYPE				;TYPE NAME OF TEST	
010060	012005				MEPASS					
010062	005067	001344			CLR	LAST			;CLEAR LAST ERROR PC	
010066	005067	001274			CLR	ERRFLG			;CLEAR ERROR FLAG	
010072	005267	001272			INC	PASCNT			;UPDATE PASS COUNT	
010076	005767	170700			TST	LIGHTS			; ARE WE USING LIGHTS?	: 4
010102	001005				BNE	2\$; BRANCH IF WE ARE	: 6
010104	104401				TYPE				; TYPE PASCOUNT MESSAGE	: 5
010106	012020				PASTXT					: 5
010110	104402				OCTASC				; PRINT PASCOUNT	: 4
010112	010150				PASARG					: 4
010114	000403				BR	3\$; CONTINUE	: 4 : 6
010116				2\$:						: 4
010116	016767	001246	170656		MOV	PASCNT,LIGHTS			;DISPLAY PASS COUNT	: 4
010124				3\$:						: 4
010124	013701	000042			MOV	@#42,R1			;CHECK FOR ACT-11 OR DDP	
010130	001405				BEQ	RESTRT			;IF NOT, CONTINUE TESTING	
010132	000005				RESET					
010134	004711			LOGICAL:	JSR	PC,(R1)				
010136	000240				NOP					
010140	000240				NOP					
010142	000240				NOP					
010144	000167	171136		RESTRT:	JMP	BEGIN				
010150	000001			PASARG:	.WORD	1			; PARAMETERS TO PRINT PASCOUNT	: 5
010152	006	002			.BYTE	6.2				: 5
010154	011370				.WORD	PASCNT				: 5
3 010156				.SCOPE						

;CHECK FOR LOOP ON CURRENT TEST
;CHECK FOR ITERATION SUPPRESSION

010156	032777	002000	170614	SCOPER:	BIT	@SW10,@SWR				: 4
010164	001030				BNE	4\$: 4
010166	032777	040000	170604	1\$:	BIT	@SW14,@SWR				: 4
010174	001021				BNE	3\$: 4
010176	032777	004000	170574		BIT	@SW1.1,@SWR				: 4
010204	001006				BNE	2\$				
010206	005267	001172			INC	LPCNT				
010212	026767	001166	001162		CMF	LPCNT,ICOUNT				
010220	001007				BNE	3\$				
010222	005067	001156		2\$:	CLR	LPCNT				
010226	005067	001134			CLR	ERRFLG				
010232	011667	001136			MOV	(SP),RETRN				
010236	000002				RTI					
010240	016716	001130		3\$:	MOV	RETRN,(5?)				
010244	000002				RTI					
010246	005767	001114		4\$:	TST	ERRFLG				
010252	001745				BEQ	1\$				

010254 000762
4 010256

DR 2\$
.SCOP1

;CHECK FOR FREEZE ON CURRENT DATA

010256 032777 001000 170514
010264 001402
010266 016716 001106
010272 000002

SCOP1R: BIT #SW09,@SWR
BEQ 1\$
MOV FREEZ1,(SP)
1\$: RTI

; 4

1 010274

.ERROR

:ERROR HANDLER

```

010274 032777 020000 170476 ERRORS: BIT      #SW13,@SWR
010302 001055          BNE      HALTS
010304 021667 001122      CMP      (SP),LAST
010310 001404          BEQ      1$
010312 011667 001114      MOV      (SP),LAST
010316 005067 001044      CLR      ERRFLG
010322 104406          1$: SAVO5P
010324 011605          MOV      (SP),R5
010326 162705 000002      SUB      #2,R5
010332 011504          MOV      (R5),R4
010334 006304          ASL      R4
010336 006304          ASL      R4
010340 042704 177001      BIC      #177001,R4
010344 062704 012140      ADD      @ERRTAB,R4
010350 012467 000040      MOV      (R4)+,ERRMSG
010354 011467 000052      MOV      (R4),DATABP
010360 005767 001002      TST      ERRFLG
010364 001403          BEQ      TYPMSG
010366 005767 000040      TST      DATABP
010372 001011          BNE      TYPDAT
010374 104401          TYPMSG: TYPE
010376 011715          MCRLF
010400 104402          OCTASC
010402 010500          ERTAB0
010404 012767 000001 000754 MOV      #1,ERRFLG
010412 104401          TYPE
010414 000000          ERRMSG: 0
010416 005767 000010      TYPDAT: TST      DATABP
010422 001404          BEQ      RESREG
010424 104401          TYPE
010426 011715          MCRLF
010430 104402          OCTASC
010432 000000          DATABP: 0
010434 104407          RESREG: RES05
010436 005777 170336      HALTS: TST      @SWR
010442 100005          BPL      EXITER
010444 010046          PUSHRO
010446 016600 000002      MOV      2(SP),R0
010452 000000          HALT
010454 012600          POPRO
010456 005267 000710      EXITER: INC      ERRCNT
010462 032777 002000 170310 BIT      #SW10,@SWR
010470 001402          BEQ      1$
010472 016716 000700      MOV      ESCAPE,(SP)
010476 000002          1$: RTI
010500 000001          ERTAB0: 1
010502 006 002          .BYTE 6,2
010504 011424          SAVPC
    
```

: 4

: 3
: 5
: 5

: 5
: 5

: 4

: 4

```

010506          .TRPSRV
                  ;TRAP DISPATCH SERVICE
                  ;ARGUMENT OF TRAP IS EXTRACTED
                  ;AND USED AS OFFSET TO OBTAIN POINTER
                  ;TO SELECTED SUBROUTINE
; 3

010506 011646          TRPSRV: MOV      (SP),-(SP)          ;GET PC OF RETURN
010510 162716 000002          SUB      @2,(SP)            ;=PC OF TRAP
010514 017616 000000          MOV      @0(SP),(SP)         ;GET TRP
010520 006316          TRPOK: ASL      (SP)                ;MULTIPLY TRAP ARG BY 2
010522 042716 177001          BIC      @177001,(SP)         ;CLEAR UNWANTED BITS
010526 062716 012060          ADD      @TRPTAB,(SP)        ;POINTER TO SUBROUTINE ADDRESS
010532 017616 000000          MOV      @0(SP),(SP)         ;SUBROUTINE ADDRESS
010536 000136          JMP      @0(SP)+                   ;GO TO SUBROUTINE
2 010540          .SAVREG
                  ;SAVE PC OF TEST THAT FAILED AND R0-R5

010540 016667 000004 000656 SV05P: MOV      4(SP),SAVPC
                  ;SAVE R0-R5

010546 010567 000646          SV05:  MOV      R5,SAVR5
010552 010467 000640          MOV      R4,SAVR4
010556 010367 000632          MOV      R3,SAVR3
010562 010267 000624          MOV      R2,SAVR2
010566 010167 000616          MOV      R1,SAVR1
010572 010067 000610          MOV      R0,SAVR0
010576 000002          RTI
3 010600          .RESREG
                  ;RESTORE R0-R5

010600 016700 000602          RS05:  MOV      SAVR0,R0
010604 016701 000600          MOV      SAVR1,R1
010610 016702 000576          MOV      SAVR2,R2
010614 016703 000574          MOV      SAVR3,R3
010620 016704 000572          MOV      SAVR4,R4
010624 016705 000570          MOV      SAVR5,R5
010630 000002          RTI

```

1 010632

.TYPER

;TELETYPE OUTPUT ROUTINE

010632 017605 000000
 010636 062716 000002
 010642 105777 000462
 010646 100375
 010650 105715
 010652 001001
 010654 000002
 010656 112577 000450
 010662 000767

TYPER: MOV @ (SP),R5
 ADD #2,(SP)
 1\$: TSTB @TPCSR
 BPL 1\$
 TSTB (R5)
 BNE 2\$
 RTI
 2\$: MOVB (R5)+,@TPDBR
 BR 1\$

; 3

2 010664

.INSTRG

;ASCII STRING INPUT ROUTINE

010664 017667 000000 000006
 010672 062716 000002
 010676 104401
 010700 000000
 010702 012704 012102
 010706 012703 000007
 010712 105777 000406
 010716 100375
 010720 117714 000402
 010724 142714 000200
 010730 122427 000015
 010734 001413
 010736 117777 000364 000366
 010744 105777 000360
 010750 100375
 010752 005303
 010754 001356
 010756 104401
 010760 011711
 010762 000745
 010764 000002

INSTRG: MOV @ (SP),MSG
 ADD #2,(SP)
 INSTR1: TYPE
 MSG: 0
 MOV @INBUF,R4
 MOV #7,R3
 1\$: TSTB @TKCSR
 BPL 1\$
 MOVB @TKDBR,(R4)
 BICB #200,(R4)
 CMPB (R4)+,#15
 BEQ INSTR2
 MOVB @TKDBR,@TPDBR
 2\$: TSTB @TPCSR
 BPL 2\$
 DEC R3
 BNE 1\$
 INSTR2: TYPE
 MQM
 BR INSTR1
 INSTR2: RTI

1 010766

.PARAMS

;CONVERT ASCII STRING TO OCTAL

; 3

010766 011605
 010770 012567 000146
 010774 012567 000144
 011000 012567 000142
 011004 112567 000140
 011010 112567 000135
 011014 010516
 011016 005005
 011020 012704 012102
 011024 122714 000015
 011030 001420
 011032 121427 000060
 011036 002415
 011040 121427 000067
 011044 003012
 011046 142714 000060
 011052 152405
 011054 122714 000015
 011060 001406
 011062 006305
 011064 006305
 011066 006305
 011070 000760
 011072 104404
 011074 000750

PARAMS: MOV (SP),R5
 MOV (R5)+,LOLIM
 MOV (R5)+,HILIM
 MOV (R5)+,DEVADR
 MOV (R5)+,LOBITS
 MOV (R5)+,ADRCNT
 MOV R5,(SP)
 PARAM1: CLR R5
 MOV #INBUF,R4
 CMPB #15,(R4)
 BEQ PARERR
 1\$: CMPB (R4),#60
 BLT PARERR
 CMPB (R4),#67
 BGT PARERR
 BICB #60,(R4)
 BISB (R4)+,R5
 CMPB #15,(R4)
 BEQ LIMITS
 ASL R5
 ASL R5
 ASL R5
 BR 1\$
 PARERR: INSTER
 BR PARAM1

;TEST TO SEE IF NUMBER IS WITHIN LIMITS

011076 020567 000042
 011102 101373
 011104 020567 000032
 011110 103770
 011112 136705 000032
 011116 001365

LIMITS: CMP R5,HILIM
 BHI PARERR
 CMP R5,LOLIM
 BLO PARERR
 BITB LOBITS,R5
 BNE PARERR

; 3

;STORE NUMBER AT SPECIFIED ADDRESS

011120 016704 000022
 011124 010524
 011126 062705 000002
 011132 105367 000013
 011136 001372
 011140 000002
 011142 000000
 011144 000000
 011146 000000
 011150 000000
 011151

1\$: MOV DEVADR,R4
 MOV R5,(R4)+
 ADD #2,R5
 DECB ADRCNT
 BNE 1\$
 RTI
 LOLIM: 0
 HILIM: 0
 DEVADR: 0
 LOBITS: 0
 ADRCNT=LOBITS+1

011152

.OCTASC

;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER

011152 017601 000000
 011156 062716 000002
 011162 012167 000130
 011166 112167 000126
 011172 112167 000123
 011176 013167 000120
 011202 016704 000114
 011206 116705 000106
 011212 012700 012114
 011216 010403
 011220 042703 177770
 011224 062703 000260
 011230 110320
 011232 006204
 011234 006204
 011236 006204
 011240 005305
 011242 001365
 011244 012703 012126
 011250 114023
 011252 105367 000042
 011256 001374
 011260 105767 000035
 011264 001405
 011266 112723 000240
 011272 105367 000023
 011276 001373
 011300 105013
 011302 104401
 011304 012126
 011306 005367 000004
 011312 001325
 011314 000002
 011316 000000
 011320 000000
 011322 000000

OCTASN: MOV @ (SP), R1
 ADD @2, (SP)
 MOV (R1)+, WRDCNT
 1\$: MOV (R1)+, CHRCNT
 MOV (R1)+, SPACNT
 MOV @ (R1)+, BINWRD
 2\$: MOV BINWRD, R4
 MOV CHRCNT, R5
 MOV @TEMP, R0
 3\$: MOV R4, R3
 BIC @177770, R3
 ADD @260, R3
 MOV R3, (R0)+
 ASR R4
 ASR R4
 ASR R4
 DEC R5
 BNE 3\$
 MOV @MDATA, R3
 4\$: MOV (R0), (R3)+
 DECB CHRCNT
 BNE 4\$
 TSTB SPACNT
 BEQ 6\$
 5\$: MOV @240, (R3)+
 DECB SPACNT
 BNE 5\$
 6\$: CLRB (R3)
 TYPE MDATA
 DEC WRDCNT
 BNE 1\$
 RTI
 WRDCNT: 0
 CHRCNT: 0
 SPACNT=CHRCNT+1
 BINWRD: 0

; 5
; 3

```

011324          .POINT  †/DHSCR,DHNRC,DHLPR,DHBA,DHBC,DHBAR,DHBCR,DHSSR,DHSLR,DHRVEC,DHRLVL,DHTVEC,DHTLVL/
                ;INDIRECT POINTERS
                ; 3

011324 177560   TKCSR:  177560
011326 177562   TKDBR:  177562
011330 177564   TPCSR:  177564
011332 177566   TPDBR:  177566
TLVL>          .IRP    A    <DHSCR,DHNRC,DHLPR,DHBA,DHBC,DHBAR,DHBCR,DHSSR,DHSLR,DHRVEC,DHRLVL,DHTVEC,DH
                A:      0
                .ENDM
011334 000000   DHSCR:  0
011336 000000   DHNRC:  0
011340 000000   DHLPR:  0
011342 000000   DHBA:   C
011344 000000   DHBC:   C
011346 000000   DHBAR:  0
011350 000000   DHBCR:  0
011352 000000   DHSSR:  0
011354 000000   DHSLR:  0
011356 000000   DHRVEC: 0
011360 000000   DHRLVL: 0
011362 000000   DHTVEC: 0
011364 000000   DHTLVL: 0
2 011366       .VARIA
                ;PROGRAM VARIABLES

011366 000000   ERRFLG: 0          ;ERROR FLAG
011370 000000   PASCNT: 0         ;PASS COUNT
011372 000000   ERRCNT: 0         ;ERROR COUNT
011374 000000   RETRN:  0         ;SCOPE RETURN ADDRESS FOR TEST LOOPING
011376 000000   ESCAPE: 0         ;ADDRESS FOR ERROR ESCAPE
011400 000000   FREEZ1: 0         ;DATA LOOPING RETURN ADDRESS
011402 000000   ICOUNT: 0        ;ITERATION COUNT FOR TEST IN PROGRESS
011404 000000   LPCNT:  0         ;NUMBER OF ITERATIONS THIS TEST
011406 000000   SAVRO:  0         ;R0 SAVE AREA
011410 000000   SAVR1:  0         ;R1 SAVE AREA
011412 000000   SAVR2:  0         ;R2 SAVE AREA
011414 000000   SAVR3:  0         ;R3 SAVE ARE
011416 000000   SAVR4:  0         ;R4 SAVE AREA
011420 000000   SAVR5:  0         ;R5 SAVE AREA
011422 000000   SAVSP:  0         ;STACK POINTER SAVE AREA
011424 000000   SAVPC:  0         ;CALLING ROUTINE SAVE AREA
011426 000000   INIFLG: 0         ;PROGRAM INITIALIZATION FLAG
011430 000000   STFLG:  0         ;PROGRAM START FLAG
011432 000000   LAST:   0         ;LAST ERROR PC
                .IRP    A    <>
                A:      0
                .ENDM

```

1 011434

.PFAIL

;ENTER HERE ON POWER FAILURE

```

011434 010046          PFAIL:  MOV    R0,-(SP)          ;SAVE R0-R5 ON PROCESSOR STACK
011436 010146          MOV    R1,-(SP)
011440 010246          MOV    R2,-(SP)
011442 010346          MOV    R3,-(SP)
011444 010446          MOV    R4,-(SP)
011446 010546          MOV    R5,-(SP)
011450 016746 166350   MOV    24,-(SP)
011454 010667 177742   MOV    SP,SAVSP          ;SAVE STACK POINTER
011460 012767 011472 166336  MOV    @RESTART,24      ;SET UP FOR POWER UP TRAP
011466 000000          HALT                                ;HALT ON POWER DOWN NORMAL
011470 000777          BR

```

;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

```

011472 016706 177724   RESTAR: MOV    SAVSP,SP          ;RESTORE STACK POINTER
011476 012605          MOV    (SP)+,R5          ;RESTORE R0-R5
011500 012604          MOV    (SP)+,R4
011502 012603          MOV    (SP)+,R3
011504 012602          MOV    (SP)+,R2
011506 012601          MOV    (SP)+,R1
011510 012600          MOV    (SP)+,R0
011512 012767 011434 166304  MOV    @PFAIL,24        ;SET UP FOR POWER FAILURE
011520 012767 000340 166250  MOV    @340,PS
011526 012706 012774   MOV    @STACK,SP
011532 005067 000356   CLR    TEMP
011536 005267 000352   INC    TEMP
011542 001375          BNE    .-4
011544 104401          TYPE                                ; 5
011546 011715          MCRLF                                ; 5
011550 104402          OCTASC
011552 011574          PFTAB
011554 104401          TYPE
011556 011720          MPFAIL
011560 005067 177602   CLR    ERRFLG
011564 005067 177642   CLR    LAST
011570 000177 177600   JMP    @RETRN
011574 000001          PFTAB: 1
011576 000006 000002   6,2
011602 011374          RETRN

```

011604				.MSG	†/DH11 MEMORY TEST/,†/CZDHB-CO/
C11604	015	012	012	MTITLE:	.ASCIZ <15><12><12>/DH11 MEMORY TEST /<15><12>
011607	104	110	061		
011612	061	040	115		
011615	105	115	117		
011620	122	131	040		
011623	124	105	123		
011626	124	040	015		
011631	012	000			
011633	015	012	126	MVECTOR:	.ASCIZ <15><12>/VECTOR ADDRESS-/
011636	105	103	124		
011641	117	122	040		
011644	101	104	104		
011647	122	105	123		
011652	123	055	000		
011655	015	012	103	MREGAD:	.ASCIZ <15><12>/CONTROL REGISTER ADDRESS-/
011660	117	116	124		
011663	122	117	114		
011666	040	122	105		
011671	107	111	123		
011674	124	105	122		
011677	040	101	104		
011702	104	122	105		
011705	123	123	055		
011710	000				
011711	040	040	077	MQM:	.ASCIZ / ?/
011714	000				
011715	015	012	000	MCRLF:	.ASCIZ <15><12>
011720	040	040	120	MPFAIL:	.ASCIZ / POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
011723	117	127	105		
011726	122	040	106		
011731	101	111	114		
011734	125	122	105		
011737	054	040	120		
011742	122	117	107		
011745	122	101	115		
011750	040	122	105		
011753	123	124	101		
011756	122	124	040		
011761	101	124	040		
011764	124	105	123		
011767	124	040	111		
011772	116	040	120		
011775	122	117	107		
012000	122	105	123		
012003	123	000			
012005	015	012	103	MEPASS:	.ASCIZ <15><12>/CZDHB-CO/
012010	132	104	110		
012013	102	055	103		
012016	060	000			
012020	015	012	120	PASTXT:	.ASCIZ <15><12>/PASS COUNT = /
012023	101	123	123		
012026	040	103	117		
012031	125	116	124		
012034	040	075	040		
012037	000				

```

012040 015 012 122 MR: .ASCIZ <15><12>/R/
012043 000
012044 015 012 124 MTSTPC: .ASCIZ <15><12>/TEST PC-/
012047 105 123 124
012052 040 120 103
012055 055 000

2 012060 .EVEN
      .TRPTAB

      ;TABLE OF POINTERS FOR TRAP DECODING

012060 010156 TRPTAB: SCOPER
012062 010632 TYPER
012064 011152 OCTASN
012066 010664 INSTRG
012070 010756 INSTRE
012072 010766 PARAMS
012074 010540 SVOSP
012076 010600 RSOS
012100 010256 SCOP1R

3 012102 .BUFFER

      ;BUFFERS FOR INPUT-OUTPUT

012102 000000 INBUF: 0
      012114 .=. +10
012114 000000 TEMP: 0
      012126 .=. +10

012126 000000 MDATA: 0
      012140 .=. +10
4 012140 .ERRTAB

      ;TABLE OF POINTERS TO ERROR MESSAGES AND DATA

012140 ERRTAB:
5 012140 000000 0
6 012142 000000 0
7 012144 012170 EM1
8 012146 012526 DT1
9 012150 012252 EM2
10 012152 012526 DT1
11 012154 012333 EM3
12 012156 012544 DT2
13 012160 012401 EM4
14 012162 012544 DT2
15 012164 012446 EM5
16 012166 012556 DT3
17 012170 102 125 123 EM1: .ASCIZ /BUS ADDRESS MEMORY ERROR/<15><12>/EXP REC ADDRESS/
012173 040 101 104
012176 104 122 105
012201 123 123 040
012204 115 105 115
012207 117 122 131
012212 040 105 122
012215 122 117 122
012220 015 012 105

```

	012223	130	120	040				
	012226	040	040	040				
	012231	040	122	105				
	012234	103	040	040				
	012237	040	040	040				
	012242	101	104	104				
	012245	122	105	123				
	012250	123	000					
18	012252	102	131	124	EM2:	.ASCIZ	/BYTE COUNT MEMORY ERROR/<15><12>/EXP	REC ADDRESS/
	012255	105	040	103				
	012260	117	125	116				
	012263	124	040	115				
	012266	105	115	117				
	012271	122	131	040				
	012274	105	122	122				
	012277	117	122	015				
	012302	012	105	130				
	012305	120	040	040				
	012310	040	040	040				
	012313	122	105	103				
	012316	040	040	040				
	012321	040	040	101				
	012324	104	104	122				
	012327	105	123	123				
	012332	000						
19	012333	102	125	123	EM3:	.ASCIZ	/BUS ADDRESS MEMORY ERROR/<15><12>/EXP	REC/
	012336	040	101	104				
	012341	104	122	105				
	012344	123	123	040				
	012347	115	105	115				
	012352	117	122	131				
	012355	040	105	122				
	012360	122	117	122				
	012363	015	012	105				
	012366	130	120	040				
	012371	040	040	040				
	012374	040	122	105				
	012377	103	000					
20	012401	102	131	124	EM4:	.ASCIZ	/BYTE COUNT MEMORY ERROR/<15><12>/EXP	REC/
	012404	105	040	103				
	012407	117	125	116				
	012412	124	040	115				
	012415	105	115	117				
	012420	122	131	040				
	012423	105	122	122				
	012426	117	122	015				
	012431	012	105	130				
	012434	120	040	040				
	012437	040	040	040				
	012442	122	105	103				
	012445	000						
21	012446	115	105	115	EM5:	.ASCIZ	/MEMORY EXTENSION ERROR/<15><12>/EXP	REC ADDRESS/
	012451	117	122	131				
	012454	040	105	130				
	012457	124	105	116				
	012462	123	111	117				
	012465	116	040	105				

	012470	122	122	117	
	012473	122	015	012	
	012476	105	130	120	
	012501	040	040	040	
	012504	040	040	122	
	012507	105	103	040	
	012512	040	040	040	
	012515	040	101	104	
	012520	104	122	105	
	012523	123	123	000	
22					.EVEN
23	012526	000003			DT1: 3
24	012530	006	002		.BYTE 6.2
25	012532	011412			SAVR2
26	012534	006	002		.BYTE 6.2
27	012536	011414			SAVR3
28	012540	002	000		.BYTE 2.0
29	012542	011416			SAVR4
30	012544	000002			DT2: 2
31	012546	006	002		.BYTE 6.2
32	012550	011420			SAVR5
33	012552	006	002		.BYTE 6.2
34	012554	011416			SAVR4
35	012556	000003			DT3: 3
36	012560	006	002		.BYTE 6.2
37	012562	011420			SAVR5
38	012564	006	002		.BYTE 6.2
39	012566	011416			SAVR4
40	012570	002	000		.BYTE 2.0
41	012572	011414			SAVR3
42	012574				.ENDCOD
	012574	000000			ENDCOD: 0
43		000001			.END

ADRCNT = 011151	EM3 012333	MVECTO 011633	STFLG 011430	T21 003400
ADRS = 000000	EM4 012401	N = 000001	SV05 010546	T22 003476
BEGIN 001306	EM5 012446	OCTASC = 104402	SV05P 010540	T23 003574
BINWRD 011322	ENDCOD 012574	OCTASN 011152	SWR 001000	T24 003672
BIT00 = 000001	EOP 010056	PARAM = 104405	SW00 = 000001	T25 003770
BIT01 = 000002	ERRCNT 011372	PARAMS 010756	SW01 = 000002	T26 004066
BIT02 = 000004	ERRFLG 011366	PARAM1 011016	SW02 = 000004	T27 004164
BIT03 = 000010	ERRMSG 010414	PARERR 011072	SW03 = 000010	T3 001634
BIT04 = 000020	ERRORS 010274	PASARG 010150	SW04 = 000020	T30 004262
BIT05 = 000040	ERRTAB 012140	PASCNT 011370	SW05 = 000040	T31 004360
BIT06 = 000100	ERTAB0 010500	PASTXT 012020	SW06 = 000100	T32 004456
BIT07 = 000200	ESCAPE 011376	PFAIL 011434	SW08 = 000400	T33 004554
BIT08 = 000400	EXITER 010456	PFTAB 011574	SW09 = 001000	T34 004652
BIT09 = 001000	FREEZ1 011400	POPRO = 012600	SW10 = 002000	T35 004750
BIT10 = 002000	HALTS 010436	POP1SP = 005726	SW11 = 004000	T36 005046
BIT11 = 004000	HILIM 011144	POP2SP = 022626	SW12 = 010000	T37 005144
BIT12 = 010000	ICOUNT 011402	PS = 177776	SW13 = 020000	T4 001732
BIT13 = 020000	INBUF 012102	PUSHRO = 010046	SW14 = 040000	T40 005242
BIT14 = 040000	INIFLG 011426	PUSH1S = 005746	SW15 = 100000	T41 005340
BIT15 = 100000	INSTER = 104404	PUSH2S = 024646	TEMP 012114	T42 005436
CADRS = 000000	INSTR = 104403	RESREG 010434	TKCSR 011324	T43 005534
CHRCNT 011320	INSTR1 010756	RESTAR 011472	TKDBR 011326	T44 005710
DATABP 010432	INSTRG 010664	RESTRT 010144	TPCSR 011330	T45 006064
DEVADR 011146	INSTR2 010676	RES05 = 104407	TPDBR 011332	T46 006240
DHBA 011342	INSTR1 010676	RETRN 011374	TRPOK 010520	T47 006414
DHBA 011342	INSTR2 010764	RS05 010600	TRPSRV 010506	T5 002030
DHBAR 011346	LAST 011432	SAVPC 011424	TRPTAB 012060	T50 006570
DHBC 011344	LIGHTS 001002	SAVR0 011406	TYPDAT 010416	T51 006744
DHBCR 011350	LIMITS 011076	SAVR1 011410	TYPE = 104401	T52 007122
DHLCR 011340	LOBITS 011150	SAVR2 011412	TYPER 010632	T53 007300
DHNRC 011336	LOGICA 010134	SAVR3 011414	TYPMSG 010374	T54 007472
DHRLVL 011360	LOLIM 011142	SAVR4 011416	T1 001400	T55 007664
DHRVEC 011356	LPCNT 011404	SAVR5 011420	T10 002322	T6 002126
DHSCR 011334	MCRLF 011715	SAVS 011422	T11 002420	T7 002224
DHSLR 011354	MDATA 012126	SAVSP 011422	T12 002516	VEC1 001164
DHSSR 011352	MEPASS 012005	SAV05P = 104406	T13 002614	VEC2 001174
DHTLVL 011364	MPFAIL 011720	SCOPE = 104400	T14 002712	WRDCNT 011316
DHTVEC 011362	MQM 011711	SCOPE1 = 104410	T15 003010	X = 000000
DT1 012526	MR 012040	SCOPE1R 010256	T16 003106	XADRS = 000020
DT2 012544	MREGAD 011655	SPACNT = 011321	T17 003204	XCADRS = 000020
DT3 012556	MSG 010700	STACK = 012774	T2 001516	XN = 000056
EM1 012170	MTITLE 011604	START 001004	T20 003302	Y = 000011
EM2 012252	MTSTPC 012044			

. ABS. 012576 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 20224 WORDS (79 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 71 PAGES
CZDHBC.BIN,CZDHBC.SEQ=CZDHBC.DOC,DHMACA.MAC,CZDHBC.P11