



.REM !

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

IDENTIFICATION  
-----

PRODUCT CODE: AC-8444D-MC  
PRODUCT NAME: CZDHADO DH11 STATIC LOGIC TEST  
DATE: JUNE 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, 1978, 1985 BY DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23

1. ABSTRACT

THE DH11 STATIC LOGIC TEST IS DESIGNED TO PROVIDE A MEANS FOR TESTING THE CORRECT FUNCTION OF ALL READ/WRITE BITS IN THE FOLLOWING DH11 REGISTERS:

DH11 SYSTEM CONTROL REGISTER

DH11 LINE PARAMETER REGISTER

DH11 BREAK CONTROL REGISTER

DH11 SILO STATUS REGISTER

IN ADDITION, TESTS ARE PROVIDED TO CHECK THE FUNCTION OF THOSE BITS THAT ARE READ ONLY IN MAINTENANCE MODE. ALSO PROVIDED ARE TESTS OF REGISTER ADDRESSABILITY, AND OF THE FUNCTION OF MASTER CLEAR.

THE DIAGNOSTIC HAS BEEN WRITTEN SO THAT THE TESTING OF EACH FUNCTION IS CONTAINED IN AN INDIVIDUAL TEST LOOP.

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 8KW OF MEMORY
  - ASR-33 TELETYPE OR EQUIVALENT
  - DH11 ASYNCHRONOUS MULTIPLEXER
  - DM11 MAINTENANCE CARD INSTALLED
- 2.2 STORAGE
  - THE PROGRAM LOADS INTO 8KW OF MEMORY
- 3. LOADING PROCEDURE
  - THE STANDARD PROCEDURE FOR LOADING ABSOLUTE BINARY TAPES CAN TO BE USED, OR CAN BE USED WITH XXDP.
- 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 IS 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 STATIC LOGIC 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  
55  
56  
57

- 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 STATIC LOGIC 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 STATIC LOGIC 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

58  
59  
60  
61  
62

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 IMMEDIATLY 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 RO. 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 "CZDHA-D" 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 "CZDHA-D" 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  
45

- 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 CZDHA-D TO END OF TYPEOUT OF CZDHA-D)  
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

9. PROGRAM DESCRIPTION

THIS PROGRAM IS A LOW LEVEL TEST OF DH11 CONTROL REGISTERS.

THE PROGRAM BEGINS BY CHECKING THE ADDRESSABILITY OF EACH DH11 REGISTER WITHOUT CONCERN FOR ANY DATA CONTENT. THE PURPOSE OF THESE TESTS IS TO VERIFY THAT THE ADDRESS SELECTORS FOR THE VARIOUS REGISTERS ARE FUNCTIONING.

THE NEXT SET OF TESTS VERIFIES THAT EACH DH11 REGISTER CAN BE MASTER CLEARED, AFTER ALL READ/WRITE BITS HAVE BEEN SET TO 1. THIS TEST DOES NOT VERIFY THAT ALL BITS HAVE BEEN SET, ONLY THAT THEY HAVE BEEN CLEARED.

THE NEXT GROUP OF TESTS EXERCISES EACH READ/WRITE BIT IN THE DH11 SYSTEM CONTROL REGISTER, IN BOTH NORMAL AND MAINTENANCE MODES OF OPERATION. IN NORMAL MODE, EACH READ/WRITE BIT IS SET AND CLEARED, AND READ ONLY BITS ARE CHECKED FOR READ ONLY FUNCTION.

IN MAINTENANCE MODE, THE BITS THAT ARE READ ONLY IN NORMAL MODE ARE CHECKED FOR READ/WRITE OPERATION.

THE NEXT GROUP OF TESTS CHECKS EACH READ/WRITE BIT OF THE DH11 LINE PARAMATER REGISTER, BREAK CONTROL REGISTER AND SILO STATUS REGISTER FOR READ/WRITE CAPABILITY. EACH BIT OF EACH REGISTER IS CHECKED IN AN INDIVIDUAL TEST LOOP.

THE NEXT GROUP OF TESTS CHECKS CLEARING OF A SINGLE BIT IN EACH OF THE LINE PARAMETER, BREAK CONTROL AND SILO STATUS REGISTERS WITH ALL OTHER READ/WRITE BITS SET TO 1.

THE FINAL TWO TESTS VERIFY THAT A MOVE BYTE TO ONE BYTE OF THE SYSTEM CONTROL REGISTER DOES NOT AFFECT THE OTHER BYTE OF THE SYSTEM CONTROL REGISTER.

AFTER ALL TESTS HAVE BEEN COMPLETED, THE PROGRAM TYPES "CZDHA-D" AND RESTARTS THE SEQUENCE OF TESTING JUST DESCRIBED.

10. LISTING

!

```

1      ; DHMAC-A - DH11 MACRO LIBRARY
2      ; COPYRIGHT 1985, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
3
4
5      .LIST  ME
6      .NLIST MC,MD,CND
7
104
119
131
148
158
167
303
339
373
520
563
595
607
652
664
691
712
743      ; CMS REPLACEMENT HISTORY
744
745
746      ; *9 SKONETSKI 26-APR-1985 16:23:08 "FIXED TYPO CAUSING ASSEMBLY ERRORS"
747      ; *8 SKONETSKI 22-APR-1985 16:48:03 "TYPO ERROR IN VECTOR CHANGE CODE SOURCE FIXED"
748      ; *7 SKONETSKI 22-APR-1985 16:26:04 "ADDED CODE TO SET VECTORS FOR PWR FAIL, ERRORS, AND EMT
TRAPS."
749      ; *6 SKONETSKI 22-APR-1985 14:22:35 "FIXED BRANCH ERROR IN END OF PASS ROUTINE"
750      ; *5 SKONETSKI 22-APR-1985 08:28:54 "FIXED BUG (AN OCTASC MACRO CALL WAS WRONG) AND ADDED A
CLEAN END OF PASS
MESSAGE.
751      ; *4 SKONETSKI 18-APR-1985 14:20:15 "ADDED SOFTWARE SWITCH REG SUPPORT, BUT UNTESTED"
752      ; *3 SKONETSKI 12-APR-1985 10:34:52 "FIXED PROBLEMS WITH SPURIOUS CR/LFS"
753      ; *2 SKONETSKI 11-APR-1985 16:00:24 "ADDED MACRO FROM SYSMAC.SML THAT SIZES FOR SOFTWARE SWI
TCH REGISTER"
754      ; *1 SKONETSKI 11-APR-1985 15:49:05 "LIBRARY FOR DH11 DIAGNOSTICS"

```

; 3

2  
3  
5 000000

```
.LIST ME
.NLIST MC,MD,CND
.HEADER †/1972,1978,1985/,†/DH11 STATIC LOGIC TEST/,†/CZDHA-D/

;STARTING PROCEDURE
;LOAD PROGRAM
;LOAD ADDRESS 000200
;PRESS START
;PROGRAM WILL TYPE DH11 STATIC LOGIC 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 " CZDHA-D "
;AND THEN RESUM TESTING
```

; 3

000000

```
.TITLE CZDHA-D
.ENABLE ABS
.NLIST MC,MD,CND
.LIST ME
.SYMBOLS
```

6 0C0000

```
;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 TYPEOUT

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
021544 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 ; 3
040000 BIT14=40000
020000 BIT13=20000
010000 BIT12=10000
004000 BIT11=4000
002000 BIT10=2000
001000 BIT09=1000
000400 BIT08=400
000200 BIT07=200
000100 BIT06=100
000040 BIT05=40
000020 BIT04=20
000010 BIT03=10
000004 BIT02=4
000002 BIT01=2
000001 BIT00=1
1 000000 .CATCH
    
```



```

0
000000 000000 ;TRAPCATCAER FOR ILLEGAL INTERRUPTS
000200 .=0
.REPT 200 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
          HALT ;EXAMINE STACK TO FIND CAUSE
.ENDR     .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000000 000002 HALT ;EXAMINE STACK TO FIND CAUSE
000002 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000004 000006 HALT ;EXAMINE STACK TO FIND CAUSE
000006 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000010 000012 HALT ;EXAMINE STACK TO FIND CAUSE
000012 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000014 000016 HALT ;EXAMINE STACK TO FIND CAUSE
000016 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000020 000022 HALT ;EXAMINE STACK TO FIND CAUSE
000022 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000024 000026 HALT ;EXAMINE STACK TO FIND CAUSE
000026 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000030 000032 HALT ;EXAMINE STACK TO FIND CAUSE
000032 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000034 000036 HALT ;EXAMINE STACK TO FIND CAUSE
000036 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000040 000042 HALT ;EXAMINE STACK TO FIND CAUSE
000042 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000044 000046 HALT ;EXAMINE STACK TO FIND CAUSE
000046 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000050 000052 HALT ;EXAMINE STACK TO FIND CAUSE
000052 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000054 000056 HALT ;EXAMINE STACK TO FIND CAUSE
000056 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000060 000062 HALT ;EXAMINE STACK TO FIND CAUSE
000062 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000064 000066 HALT ;EXAMINE STACK TO FIND CAUSE
000066 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000070 000072 HALT ;EXAMINE STACK TO FIND CAUSE
000072 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000074 000076 HALT ;EXAMINE STACK TO FIND CAUSE
000076 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000100 000102 HALT ;EXAMINE STACK TO FIND CAUSE
000102 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000104 000106 HALT ;EXAMINE STACK TO FIND CAUSE
000106 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000110 000112 HALT ;EXAMINE STACK TO FIND CAUSE
000112 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000114 000116 HALT ;EXAMINE STACK TO FIND CAUSE
000116 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000120 000122 HALT ;EXAMINE STACK TO FIND CAUSE
000122 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000124 000126 HALT ;EXAMINE STACK TO FIND CAUSE
000126 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000130 000132 HALT ;EXAMINE STACK TO FIND CAUSE
000132 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000134 000136 HALT ;EXAMINE STACK TO FIND CAUSE
000136 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000140 000142 HALT ;EXAMINE STACK TO FIND CAUSE
000142 000000 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000144 000146 HALT ;EXAMINE STACK TO FIND CAUSE
          .+2 ;UNEXPECTED TRAP TO THIS LOCATION

```

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          ;STANDARD INTERRUPT VECTORS
000200 000200 000167 000600  .-200  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/
          SCOPE=TRAP+Y          ;SCOPE LOOP AND ITERATION HANDLER
          Y=Y+1

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

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

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

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

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

000204  TRPDEF SAV05P,+/SAVE R0-R5, PC/
          SAV05P=TRAP+Y        ;SAVE R0-R5, PC
          Y=Y+1

000204  TRPDEF RES05,+/RESTORE R0-R5/
          RES05=TRAP+Y         ;RESTORE R0-R5
          Y=Y+1

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

2          .-46
3 000046  LOGICAL
4          .-52
5 000052  40000
6          .MACRO CODEM1
7          MOV     DHSSR,DHSLR          ;SET UP ADDRESS OF SILO
8          INC     DHSLR              ;STATUS REGISTER HIGH BYTE
9          .ENDM CODEM1
10 000054 .START DHRVEC,3,4,DHSCR,0,177776,7,10,...1
    
```

```

0          001000          .-1000

                                ;PROGRAM INITIALIZATION
                                ;LOCK OUT INTERRUPTS
                                ;SET UP PROCESSOR STACK
                                ;SET UP POWER FAIL VECTOR
                                ;CLEAR PROGRAM FLAGS AND COUNTS
                                ;TYPE TITLE MESSAGE
.IIF NB <>,          ;DETERMINE MEMORY SIZE
.IIF NB <>,          ;SET UP TRACE TRAP RETURN

001000 177570          SWR: .WORD 177570          ; SWITCH DHSCR ADDRESS          ; 4
001002 177570          LIGHTS: .WORD 177570          ; LIGHTS          ; 4
                                ; 4

001004 012767 000340 176764 START: MOV #340,PS          ;LOCK OUT INTERRUPTS
001012 012706 021544          MOV #STACK,SP          ;SET UP PROCESSOR STACK
001016 012702 000024          MOV #24,R2          ; POINT TO VECTOR AREA          ; 7
001022 012722 020110          MOV #PFAIL,(R2)+          ;SET UP POWER FAIL TRAP          ; 7
001026 012722 000340          MOV #340,(R2)+          ;SERVICE AT LEVEL 7          ; 7
001032 012722 016750          MOV #ERRORS,(R2)+          ;ERROR HANDLER          ; 7
001036 012722 000340          MOV #340,(R2)+          ;SERVICE AT LEVEL 7          ; 7
001042 012722 017162          MOV #TRPSRV,(R2)+          ;GENERAL HANDLER DISPATCH SERVICE ; 7
001046 012712 000340          MOV #340,(R2)          ;SERVICE AT LEVEL 7          ; 8
0C1052 005067 017026          CLR STFLG          ;CLEAR TEST START FLAG
001056 005067 016762          CLR PASCNT          ;CLEAR PASS COUNT
001062 005067 016760          CLR ERRCNT          ;CLEAR ERROR COUNT
001066 005067 016750          CLR ERRFLG          ;CLEAR ERROR FLAG
001072 005067 016744          CLR ERRFLG          ;CLEAR LAST ERROR PC
001076 016746 176702          MOV 4, -(SP)          ; PUSH TRAP VECTOR          ; 4
001102 016746 176700          MOV 6, -(SP)          ; 4
001106 012767 001122 176670 MOV #1$, 4          ; SET UP TRAP VECTOR          ; 4
001114 005777 177660          TST @SWR          ; TEST SWITCH REGISTER ADDRESS ; 4
001120 000405          BR 2$          ; IF SUCCESSFUL, LEAVE IT ALONE ; 4
001122          1$:          ; 4
001122 012767 000176 177650 MOV #176, SWR          ; POINT TO SOFT SWITCH DHSCR ; 4
001130 005067 177646          CLR LIGHTS          ; 0 MEANS WE ARE NOT GOING TO USE LIGHTS ; 4
001134          2$:          ; 5
001134 005726          TST (SP)+          ; CLEAN UP STACK          ; 4
001136 005726          TST (SP)+          ; 4
001140 012667 176642          MOV (SP)+, 6          ; 4
001144 012667 176634          MOV (SP)+, 4          ; 4
001150 104401 020260          TYPE ,MTITLE          ;TYPE TITLE MESSAGE
001154 005767 016722          TST INIFLG          ;CHECK INITIALIZATION FLAG

.IF NB <DHRVEC>
001160 001001          BNE VEC1          ;IF NOT 0, CHECK SWITCHES
                                ;FOR REINITIALIZATION

.IFF
                                ;IF NOT 0, START TEST

.ENDC
.IF NB <>
SIZE: CLR R0
      MOV #2$, @#4          ;SET UP TIME OUT RETURN
1$: TST (R0)+          ;WILL TRAP WHEN NO MEMORY          ; 9
   BR 1$          ;LOCATION RESPONDED, CONTINUE
2$: MOV R0, HCORE          ;R0 CONTAINS ADDRESS OF
   SUB #2, HCORE          ;NON EXISTANT MEMORY          ; 9
   MOV #6, @#4          ;RESTORE TRAPCATCHER

```

```

.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 ;RESTROE TRAPCATCHER
    MOV #TRTRET,@#16 ;SET UP TRACE TRAP VECTOR

.ENDC
.IF NB <DHRVEC> ; 3
.IF B <>
001162 000404 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$: MOV R2,(R1) ;RESTORE TRAPCATCHER
    CLR (R2) ;IN FLOATING VECTOR AREA
    ADD R3,R1
    ADD R3,R2
    CMP R1,#1000
    BNE 1$

;INPUT ADDRESS OF DEVICE VECTOR
;MESSAGE "VECTOR ADDRESS-"
;CONVERT STRING TO OCTAL
;LOW LIMIT
;HIGH LIMIT ; 3
;LOCATIONS TO BE FILLED
;NUMBER OF LOCATIONS
;LSB MASK
;INPUT ADDRESS OF DEVICE CSR
;MESSAGE "CONTROL REGISTER ADDRESS-"
;CONVERT STRING TO OCTAL
;LOW LIMIT
;HIGH LIMIT
;LOCATIONS TO BE FILLED
;NUMBER OF LOCATIONS
;LSB MASK

;BYTE 3
;BYTE 4
;BYTE 7
;BYTE 10
.ENDC
.IF NB <1>
001262 0016767 016540 016540 CODEM1
001262 016767 016540 016540 MOV DHSSR,DHSLR ;SET UP ADDRESS OF SILO
001270 005267 016534 INC DHSLR ;STATUS REGISTER HIGH BYTE

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

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

```



```

001306 012767 000340 176462 BEGIN: MOV #340,PS ;LOCK OUT INTERRUPTS
001314 012706 021544 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 020526 MTSTPC ;MESSAGE "TEST PC"
001334 104405 PARAM ;CONVERT STRING TO OCTAL
001336 000000 0
001340 017500 17500
001342 020050 RETRN
001344 001 .BYTE 1
001345 001 .BYTE 1
001346 000410 BR 2$
001350 012767 001400 016472 1$: MOV #T1,RETRN ;NORMAL START, TEST 1
001356 005767 016522 TST STFLG ;IF LOOPING, BYPASS TYPEOUT
001362 001004 BNE 3$
001364 005167 016514 COM STFLG
001370 104401 020522 2$: TYPE ,MR ;TYPE "R" TO INDICATE START
001374 000177 016450 3$: JMP @RETRN ;START TESTING ; 3
216 001400 RGADRS +/SCR/,+/DH11 SYSTEM CONTROL/

```

```

;DH11 SYSTEM CONTROL REGISTER ADDRESSING TEST
;VERIFY THAT DH11 SYSTEM CONTROL REGISTER RESPONDS TO ADDRESSING
;IF DH11 SYSTEM CONTROL REGISTER DOES NOT RESPOND, A BUSS ERROR TRAP
;WILL OCCUR, AND AN ERROR MESSAGE WILL BE TYPED.

```

```

001400 TS \XN,100,2$
001400 012767 000340 176370 T1: MOV #340,PS ;DISABLE ALL INTERRUPTS
001406 012767 000100 016442 MOV #100,ICOUNT ;SET UP FOR 100 ITERATIONS
001414 012767 001452 016430 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1
001422 000002 MOV #1$,@#4 ;SET UP TIME OUT TRAP
001430 012737 001444 000004 MOV #340,@#6
001436 005777 016346 TST @DHSCR ;ADDRESS DH11 SYSTEM CONTROL
;REGISTER
001442 000406 BR 3$ ;NO TRAP, REGISTER RESPONDS
;TO ADDRESSING
001444 016705 016340 1$: MOV DHSCR,R5 ;REGISTER DID NOT RESPOND
001450 HLT 0 ;TIME OUT TRAP, DH11 SYSTEM CONTROL
001450 104000 EMT 0
;REGISTER DID NOT RESPOND
001452 012716 001460 2$: MOV #3$, (SP) ;SET UP TO RETURN FROM TRAP
001456 000002 RTI ;RETURN FROM TRAP
001460 012737 000006 000004 3$: MOV #6,@#4
001466 005037 000006 CLR @#6 ;RESTORE TRAP CATCHER
001472 104400 SCOPE ;CHECK FOR ITERATIONS, LOOP
217 001474 RGADRS +/NRC/,+/DH11 NEXT RECEIVED CHARACTER/

```

```

;DH11 NEXT RECEIVED CHARACTER REGISTER ADDRESSING TEST
;VERIFY THAT DH11 NEXT RECEIVED CHARACTER REGISTER RESPONDS TO ADDRESSING
;IF DH11 NEXT RECEIVED CHARACTER REGISTER DOES NOT RESPOND, A BUSS ERROR TRAP
;WILL OCCUR, AND AN ERROR MESSAGE WILL BE TYPED.

```

```

001474      TS \XN,100,2$
001474 012767 000340 176274 T2:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
001502 012767 000100 016346      MOV    #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
001510 012767 001546 016334      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                .IF NB    <>
                MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                .ENDC
                XN=XN+1
001516      000003
001516 012737 001540 000004      MOV    #1$,#4          ;SET UP TIME OUT TRAP
001524 012737 000340 000006      MOV    #340,#6        ;ADDRESS DH11 NEXT RECEIVED CHARACTER
001532 005777 016254                TST    @DHNRC          ;REGISTER
                                ;NO TRAP, REGISTER RESPONDS
                                ;TO ADDRESSING
001536 000406                BR     3$              ;REGISTER DID NOT RESPOND
                                ;TIME OUT TRAP, DH11 NEXT RECEIVED CHARACTER
001540 016705 016246      1$:  MOV    DHNRC,R5
001544                HLT    0
001544 104000                EMT    0
                                ;REGISTER DID NOT RESPOND
001546 012716 001554      2$:  MOV    #3$, (SP)      ;SET UP TO RETURN FROM TRAP
001552 000002                RTI
                                ;RETURN FROM TRAP
001554 012737 000006 000004      3$:  MOV    #6,#4
001562 005037 000006                CLR    @#6
                                ;RESTORE TRAP CATCHER
001566 104400                SCOPE
                                ;CHECK FOR ITERATIONS, LOOP
218 001570      RGADRS  +/LPR/,+/DH 11 LINE PARAMETER/

```

```

;DH 11 LINE PARAMETER REGISTER ADDRESSING TEST
;VERIFY THAT DH 11 LINE PARAMETER REGISTER RESPONDS TO ADDRESSING
;IF DH 11 LINE PARAMETER REGISTER DOES NOT RESPOND, A BUSS ERROR TRAP
;WILL OCCUR, AND AN ERROR MESSAGE WILL BE TYPED.

```

```

001570      TS \XN,100,2$
001570 012767 000340 176200 T3:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
001576 012767 000100 016252      MOV    #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
001604 012767 001642 016240      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                .IF NB    <>
                MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                .ENDC
                XN=XN+1
001612      000004
001612 012737 001634 000004      MOV    #1$,#4          ;SET UP TIME OUT TRAP
001620 012737 000340 000006      MOV    #340,#6        ;ADDRESS DH 11 LINE PARAMETER
001626 005777 016162                TST    @DHLPR          ;REGISTER
                                ;NO TRAP, REGISTER RESPONDS
                                ;TO ADDRESSING
001632 000406                BR     3$              ;REGISTER DID NOT RESPOND
                                ;TIME OUT TRAP, DH 11 LINE PARAMETER
001634 016705 016154      1$:  MOV    DHLPR,R5
001640                HLT    0
001640 104000                EMT    0
                                ;REGISTER DID NOT RESPOND
001642 012716 001650      2$:  MOV    #3$, (SP)      ;SET UP TO RETURN FROM TRAP
001646 000002                RTI
                                ;RETURN FROM TRAP
001650 012737 000006 000004      3$:  MOV    #6,#4
001656 005037 000006                CLR    @#6
                                ;RESTORE TRAP CATCHER
001662 104400                SCOPE
                                ;CHECK FOR ITERATIONS, LOOP
219 001664      RGADRS  +/BA/,+/DH11 BUS ADDRESS/

```

```

;DH11 BUS ADDRESS REGISTER ADDRESSING TEST
;VERIFY THAT DH11 BUS ADDRESS REGISTER RESPONDS TO ADDRESSING

```

;IF DH11 BUS ADDRESS REGISTER DOES NOT RESPOND, A BUSS ERROR TRAP  
;WILL OCCUR, AND AN ERROR MESSAGE WILL BE TYPED.

```

001664      000005      TS \XN,100,2$
001664 012767 000340 176104 T4:  MOV  #340,PS          ;DISABLE ALL INTERRUPTS
001672 012767 000100 016156      MOV  #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
001700 012767 001736 016144      MOV  #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV  #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
001706 012737 001730 000004      MOV  #1$,@#4         ;SET UP TIME OUT TRAP
001714 012737 000340 000006      MOV  #340,@#6
001722 005777 016070      TST  @DHBA          ;ADDRESS DH11 BUS ADDRESS
                                ;REGISTER
001726 000406      BR   3$            ;NO TRAP, REGISTER RESPONDS
                                ;TO ADDRESSING
001730 016705 016062      1$:  MOV  DHBA,R5      ;REGISTER DID NOT RESPOND
001734      HLT  0          ;TIME OUT TRAP, DH11 BUS ADDRESS
001734 104000      EMT  0
                                ;REGISTER DID NOT RESPOND
001736 012716 001744      2$:  MOV  #3$, (SP)    ;SET UP TO RETURN FROM TRAP
001742 000002      RTI                    ;RETURN FROM TRAP
001744 012737 000006 000004      3$:  MOV  #6,@#4
001752 005037 000006      CLR  @#6           ;RESTORE TRAP CATCHER
001756 104400      SCOPE          ;CHECK FOR ITERATIONS, LOOP
220 001760      RGADRS  †/BC/,†/DH11 BYTE COUNT/
    
```

;DH11 BYTE COUNT REGISTER ADDRESSING TEST  
;VERIFY THAT DH11 BYTE COUNT REGISTER RESPONDS TO ADDRESSING  
;IF DH11 BYTE COUNT REGISTER DOES NOT RESPOND, A BUSS ERROR TRAP  
;WILL OCCUR, AND AN ERROR MESSAGE WILL BE TYPED.

```

001760      000006      TS \XN,100,2$
001760 012767 000340 176010 T5:  MOV  #340,PS          ;DISABLE ALL INTERRUPTS
001766 012767 000100 016062      MOV  #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
001774 012767 002032 016050      MOV  #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV  #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
002002 012737 002024 000004      MOV  #1$,@#4         ;SET UP TIME OUT TRAP
002010 012737 000340 000006      MOV  #340,@#6
002016 005777 015776      TST  @DHBC          ;ADDRESS DH11 BYTE COUNT
                                ;REGISTER
002022 000406      BR   3$            ;NO TRAP, REGISTER RESPONDS
                                ;TO ADDRESSING
002024 016705 015770      1$:  MOV  DHBC,R5      ;REGISTER DID NOT RESPOND
002030      HLT  0          ;TIME OUT TRAP, DH11 BYTE COUNT
002030 104000      EMT  0
                                ;REGISTER DID NOT RESPOND
002032 012716 002040      2$:  MOV  #3$, (SP)    ;SET UP TO RETURN FROM TRAP
002036 000002      RTI                    ;RETURN FROM TRAP
002040 012737 000006 000004      3$:  MOV  #6,@#4
002046 005037 000006      CLR  @#6           ;RESTORE TRAP CATCHER
002052 104400      SCOPE          ;CHECK FOR ITERATIONS, LOOP
221 002054      RGADRS  †/BCR/,†/DH11 BREAK CONTROL/
    
```

```

;DH11 BREAK CONTROL REGISTER ADDRESSING TEST
;VERIFY THAT DH11 BREAK CONTROL REGISTER RESPONDS TO ADDRESSING
;IF DH11 BREAK CONTROL REGISTER DOES NOT RESPOND, A BUSS ERROR TRAP
;WILL OCCUR, AND AN ERROR MESSAGE WILL BE TYPED.

```

```

002054      000007      TS \XN,100,2$
002054 012767 000340 175714 T6:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
002062 012767 000100 015766      MOV    #100,ICOUNT       ;SET UP FOR 100 ITERATIONS
002070 012767 002126 015754      MOV    #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                          .IF NB  <>
                          MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA      ; 3
                          .ENDC
                          XN=XN+1
002076 000007      MOV    #1$,#0#4         ;SET UP TIME OUT TRAP
002104 012737 000340 000004      MOV    #340,#0#6
002112 005777 015706      TST    @DHBCR           ;ADDRESS DH11 BREAK CONTROL
                          ;REGISTER
002116 000406      BR     3$              ;NO TRAP, REGISTER RESPONDS
                          ;TO ADDRESSING
002120 016705 015700      1$:  MOV    DHBCR,R5       ;REGISTER DID NOT RESPOND
002124      HLT    0              ;TIME OUT TRAP, DH11 BREAK CONTROL
002124 104000      EMT    0
                          ;REGISTER DID NOT RESPOND
002126 012716 002134      2$:  MOV    #3$, (SP)       ;SET UP TO RETURN FROM TRAP
002132 000002      RTI
                          ;RETURN FROM TRAP
002134 012737 000006 000004      3$:  MOV    #6,#0#4
002142 005037 000006      CLR    #0#6
002146 104400      SCOPE
222 002150      RGADRS  †/BAR/,†/BUS ACTIVE/
                          ;RESTORE TRAP CATCHER
                          ;CHECK FOR ITERATIONS, LOOP

```

```

;BUS ACTIVE REGISTER ADDRESSING TEST
;VERIFY THAT BUS ACTIVE REGISTER RESPONDS TO ADDRESSING
;IF BUS ACTIVE REGISTER DOES NOT RESPOND, A BUSS ERROR TRAP
;WILL OCCUR, AND AN ERROR MESSAGE WILL BE TYPED.

```

```

002150      000010      TS \XN,100,2$
002150 012767 000340 175620 T7:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
002156 012767 000100 015672      MOV    #100,ICOUNT       ;SET UP FOR 100 ITERATIONS
002164 012767 002222 015660      MOV    #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                          .IF NB  <>
                          MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA      ; 3
                          .ENDC
                          XN=XN+1
002172 000010      MOV    #1$,#0#4         ;SET UP TIME OUT TRAP
002200 012737 000340 000004      MOV    #340,#0#6
002206 005777 015610      TST    @DHBAR           ;ADDRESS BUS ACTIVE
                          ;REGISTER
002212 000406      BR     3$              ;NO TRAP, REGISTER RESPONDS
                          ;TO ADDRESSING
002214 016705 015602      1$:  MOV    DHBAR,R5       ;REGISTER DID NOT RESPOND
002220      HLT    0              ;TIME OUT TRAP, BUS ACTIVE
002220 104000      EMT    0
                          ;REGISTER DID NOT RESPOND
002222 012716 002230      2$:  MOV    #3$, (SP)       ;SET UP TO RETURN FROM TRAP
002226 000002      RTI
                          ;RETURN FROM TRAP
002230 012737 000006 000004      3$:  MOV    #6,#0#4

```

```

002236 005037 000006          CLR      @#6          ;RESTORE TRAP CATCHER
002242 104400                SCOPE                    ;CHECK FOR ITERATIONS, LOOP
223 002244                RGADRS  ↑/SSR/,↑/SILO STATUS/

;SILO STATUS REGISTER ADDRESSING TEST
;VERIFY THAT SILO STATUS REGISTER RESPONDS TO ADDRESSING
;IF SILO STATUS REGISTER DOES NOT RESPOND, A BUSS ERROR TRAP
;WILL OCCUR, AND AN ERROR MESSAGE WILL BE TYPED.

002244                TS \XN,100,2‡
002244 012767 000340 175524 T10:    MOV      @340,PS          ;DISABLE ALL INTERRUPTS
002252 012767 000100 015576        MOV      @100,ICOUNT      ;SET UP FOR 100 ITERATIONS
002260 012767 002316 015564        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

002266 000011                MOV      @1‡,@#4          ;SET UP TIME OUT TRAP
002274 012737 000340 000004        MOV      @340,@#6
002302 005777 015520                TST      @DHSSR          ;ADDRESS SILO STATUS
;REGISTER
002306 000406                BR       3‡              ;NO TRAP, REGISTER RESPONDS
;TO ADDRESSING
002310 016705 015512                1‡:    MOV      DHSSR,R5    ;REGISTER DID NOT RESPOND
002314                                HLT      0                ;TIME OUT TRAP, SILO STATUS
002314 104000                                EMT      0

002316 012716 002324                2‡:    MOV      @3‡,(SP)      ;REGISTER DID NOT RESPOND
002322 000002                                RTI                                     ;SET UP TO RETURN FROM TRAP
002324 012737 000006 000004        3‡:    MOV      @6,@#4          ;RETURN FROM TRAP
002332 005037 000006                CLR      @#6          ;RESTORE TRAP CATCHER
002336 104400                SCOPE                    ;CHECK FOR ITERATIONS, LOOP

224
225 002340                REGTSS  ↑/SYSTEM CONTROL/,↑/DHSCR/,173777

;MASTER CLEAR TEST
;SET SYSTEM CONTROL REGISTER TO 'CDATA'
;ISSUE MASTER CLEAR
;VERIFY THAT SYSTEM CONTROL WAS CLEARED

002340                TS \XN,4000,1‡
002340 012767 000340 175430 T11:    MOV      @340,PS          ;DISABLE ALL INTERRUPTS
002346 012767 004000 015502        MOV      @4000,ICOUNT    ;SET UP FOR 4000 ITERATIONS
002354 012767 002416 015470        MOV      @1‡,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

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

002362 000012                MOV      @173777,@DHSCR  ;SET SYSTEM CONTROL REGISTER
;TO 173777
002370 052777 004000 015412        BIS      @BIT11,@DHSCR   ;ISSUE MASTER CLEAR
002376 017704 015406        MOV      @DHSCR,R4      ;(R4)=ACTUAL DATA IN
;SYSTEM CONTROL REGISTER

;IF NB <>
BIC      @,R4              ;CLEAR UNWANTED BITS
.ENDC

002402 005704                TST      R4              ;VERIFY THAT SYSTEM CONTROL REGISTER

```

```

002404 001404          BEQ    1$
002406 005005          CLR    R5
                                ;WAS CLEARED
002410 016703 015374  MOV    DHSCR,R3
002414          HLT    5
                                ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER, 0
002414 104005          EMT    5
                                ;GET REGISTER ADDRESS
002416 104400          1$:
                                ;MASTER CLEAR FAILED
226 002420          REGTSS 1$/LINE PARAMETER/,1$/DHLPR/,177777
                                ;CHECK FOR ITERATIONS, LOOP

                                ;MASTER CLEAR TEST
                                ;SET LINE PARAMETER REGISTER TO 'CDATA'
                                ;ISSUE MASTER CLEAR
                                ;VERIFY THAT LINE PARAMETER WAS CLEARED

002420          TS \XN,4000,1$
002420 012767 000340 175350 T12:  MOV    #340,PS
                                ;DISABLE ALL INTERRUPTS
002426 012767 004000 015422  MOV    #4000,ICOUNT
                                ;SET UP FOR 4000 ITERATIONS
002434 012767 002476 015410  MOV    #1$,ESCAPE
                                ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV    #,FREEZ1
                                ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC
                                XN=XN+1
002442 000013 012777 177777 015344  MOV    #177777,@DHLPR
                                ;SET LINE PARAMETER REGISTER
                                ;TO 177777
002450 052777 004000 015332  BIS    #BIT11,@DHSCR
                                ;ISSUE MASTER CLEAR
002456 017704 015332  MOV    @DHLPR,R4
                                ;(R4)=ACTUAL DATA IN
                                ;LINE PARAMETER REGISTER
                                .IF NB <>
                                BIC    #,R4
                                ;CLEAR UNWANTED BITS
                                .ENDC
                                TST    R4
                                ;VERIFY THAT LINE PARAMETER REGISTER
                                ;WAS CLEARED
002462 005704          BEQ    1$
002464 001404          CLR    R5
                                ;(R5)=EXPECTED DATA IN
002466 005005          MOV    DHLPR,R3
                                ;LINE PARAMETER REGISTER, 0
002470 016703 015320  HLT    5
                                ;GET REGISTER ADDRESS
002474          EMT    5
                                ;MASTER CLEAR FAILED
002474 104005          1$:
                                ;CHECK FOR ITERATIONS, LOOP
227 002500          REGTSS 1$/BREAK CONTROL/,1$/DHBCR/,177777
                                ;MASTER CLEAR TEST
                                ;SET BREAK CONTROL REGISTER TO 'CDATA'
                                ;ISSUE MASTER CLEAR
                                ;VERIFY THAT BREAK CONTROL WAS CLEARED

002500          TS \XN,4000,1$
002500 012767 000340 175270 T13:  MOV    #340,PS
                                ;DISABLE ALL INTERRUPTS
002506 012767 004000 015342  MOV    #4000,ICOUNT
                                ;SET UP FOR 4000 ITERATIONS
002514 012767 002556 015330  MOV    #1$,ESCAPE
                                ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV    #,FREEZ1
                                ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC
                                XN=XN+1
002522 000014 012777 177777 015274  MOV    #177777,@DHBCR
                                ;SET BREAK CONTROL REGISTER
                                ;TO 177777

```

```

002530 052777 004000 015252      BIS    #BIT11,@DHSCR      ;ISSUE MASTER CLEAR
002536 017704 015262              MOV    @DHBCR,R4         ;(R4)=ACTUAL DATA IN
                                ;BREAK CONTROL REGISTER
                                .IF NB  <>
                                BIC    #,R4         ;CLEAR UNWANTED BITS
                                .ENDC
002542 005704                    TST    R4                 ;VERIFY THAT BREAK CONTROL REGISTER
                                ;WAS CLEARED
002544 001404                    BEQ    1$                 ;(R5)=EXPECTED DATA IN
002546 005005                    CLR    R5                 ;BREAK CONTROL REGISTER, 0
                                ;GET REGISTER ADDRESS
002550 016703 015250            MOV    DHBCR,R3         ;MASTER CLEAR FAILED
002554                                HLT    5
002554 104005                    EMT    5
002556 104400                    1$:   SCOPE                ;CHECK FOR ITERATIONS, LOOP
228 002560                    REGTS5  ↑/SILO STATUS/,↑/DHSSR/,100077,77700
                                ;MASTER CLEAR TEST
                                ;SET SILO STATUS REGISTER TO 'CDATA'
                                ;ISSUE MASTER CLEAR
                                ;VERIFY THAT SILO STATUS WAS CLEARED

002560                                TS  \XN,4000,1$
002560 012767 000340 175210      T14:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
002566 012767 004000 015262      MOV    #4000,ICOUNT     ;SET UP FOR 4000 ITERATIONS
002574 012767 002642 015250      MOV    #1$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1     ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
002602 000015 012777 100077 015216  MOV    #100077,@DHSSR   ;SET SILO STATUS REGISTER
                                ;TO 100077
002610 052777 004000 015172      BIS    #BIT11,@DHSCR   ;ISSUE MASTER CLEAR
002616 017704 015204              MOV    @DHSSR,R4       ;(R4)=ACTUAL DATA IN
                                ;SILO STATUS REGISTER
                                .IF NB  <77700>
                                BIC    #77700,R4    ;CLEAR UNWANTED BITS
                                .ENDC
002626 005704                    TST    R4                 ;VERIFY THAT SILO STATUS REGISTER
                                ;WAS CLEARED
002630 001404                    BEQ    1$                 ;(R5)=EXPECTED DATA IN
002632 005005                    CLR    R5                 ;SILO STATUS REGISTER, 0
                                ;GET REGISTER ADDRESS
002634 016703 015166            MOV    DHSSR,R3         ;MASTER CLEAR FAILED
002640                                HLT    5
002640 104005                    EMT    5
002642 104400                    1$:   SCOPE                ;CHECK FOR ITERATIONS, LOOP
229 002644                    REGTS1  ↑/LINE SELECT BIT 0/,BIT00
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;SET LINE SELECT BIT 0 IN SYSTEM CONTROL REGISTER
                                ;VERIFY THAT LINE SELECT BIT 0 WAS SET
                                ;CLEAR LINE SELECT BIT 0
                                ;VERIFY THAT LINE SELECT BIT 0 WAS CLEARED

002644                                TS  \XN,4000,2$
002644 012767 000340 175124      T15:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS

```

```

002652 012767 004000 015176      MOV      #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
002660 012767 002730 015164      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
002666 016703 015116      MOV      DHSCR,R3      ;PUT ADDRESS OF SYSTEM CONTROL REGISTER IN R3
002672 012713 000001      MOV      #BIT00,(R3)    ;SET LINE SELECT BIT 0
002676 022713 000001      CMP      #BIT00,(R3)    ;VERIFY THAT LINE SELECT BIT 0 WAS SET
002702 001404      BEQ      1$
002704 012705 000001      MOV      #BIT00,R5      ;(R5)= EXPECTED VALUE
                                ;IN SYSTEM CONTROL REGISTER
                                ;LINE SELECT BIT 0
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
002710 011304      MOV      (R3),R4
002712      HLT      1
002712 104001      EMT      1
                                ;WRITE/READ ERROR
                                ;CLEAR LINE SELECT BIT 0
002714 042713 000001      1$:    BIC      #BIT00,(R3)
002720 001403      BEQ      2$
002722 005005      CLR      R5      ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER, 0
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
002724 011304      MOV      (R3),R4
002726      HLT      1
002726 104001      EMT      1
                                ;WRITE/READ ERROR
                                ;CHECK FOR ITERATIONS, LOOP
002730 104400      2$:    SCOPE
230 002732 REGS1  +/LINE SELECT BIT 1/,BIT01
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;SET LINE SELECT BIT 1 IN SYSTEM CONTROL REGISTER
                                ;VERIFY THAT LINE SELECT BIT 1 WAS SET
                                ;CLEAR LINE SELECT BIT 1
                                ;VERIFY THAT LINE SELECT BIT 1 WAS CLEARED
002732      TS \XN,4000,2$
002732 012767 000340 175036      T16:    MOV      #340,PS      ;DISABLE ALL INTERRUPTS
002740 012767 004000 015110      MOV      #4000,ICOUNT    ;SET UP FOR 4000 ITERATIONS
002746 012767 003016 015076      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
002754 016703 015030      MOV      DHSCR,R3      ;PUT ADDRESS OF SYSTEM CONTROL REGISTER IN R3
002760 012713 000002      MOV      #BIT01,(R3)    ;SET LINE SELECT BIT 1
002764 022713 000002      CMP      #BIT01,(R3)    ;VERIFY THAT LINE SELECT BIT 1 WAS SET
002770 001404      BEQ      1$
002772 012705 000002      MOV      #BIT01,R5      ;(R5)= EXPECTED VALUE
                                ;IN SYSTEM CONTROL REGISTER
                                ;LINE SELECT BIT 1
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
002776 011304      MOV      (R3),R4
003000      HLT      1
003000 104001      EMT      1
                                ;WRITE/READ ERROR
                                ;CLEAR LINE SELECT BIT 1
003002 042713 000002      1$:    BIC      #BIT01,(R3)

```



```

003006 001403      BEQ    2$
003010 005005      CLR    R5                ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER, 0
003012 011304      MOV    (R3),R4          ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
003014           HLT    1                ;SYSTEM CONTROL REGISTER
003014 104001      EMT    1                ;SYSTEM CONTROL REGISTER
                                ;WRITE/READ ERROR
003016 104400      2$: SCOPE                                ;CHECK FOR ITERATIONS, LOOP
231 003020          REGTS1  +/LINE SELECT BIT 2/,BIT02
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;SET LINE SELECT BIT 2 IN SYSTEM CONTROL REGISTER
                                ;VERIFY THAT LINE SELECT BIT 2 WAS SET
                                ;CLEAR LINE SELECT BIT 2
                                ;VERIFY THAT LINE SELECT BIT 2 WAS CLEARED

003020           TS \XN,4000,2$
003020 012767 000340 174750 T17: MOV    #340,PS                ;DISABLE ALL INTERRUPTS
003026 012767 004000 015022   MOV    #4000,ICOUNT          ;SET UP FOR 4000 ITERATIONS
003034 012767 003104 015010   MOV    #2$,ESCAPE           ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV    #,FREEZ1                ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC
                                XN=XN+1
003042 016703 014742      MOV    DHSCR,R3            ;PUT ADDRESS OF SYSTEM CONTROL REGISTER IN R3
003046 012713 000004      MOV    #BIT02,(R3)        ;SET LINE SELECT BIT 2
003052 022713 000004      CMP    #BIT02,(R3)        ;VERIFY THAT LINE SELECT BIT 2 WAS SET
003056 001404           BEQ    1$
003060 012705 000004      MOV    #BIT02,R5          ;(R5)= EXPECTED VALUE
                                ;IN SYSTEM CONTROL REGISTER
                                ;LINE SELECT BIT 2
003064 011304      MOV    (R3),R4          ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
003066           HLT    1                ;SYSTEM CONTROL REGISTER
003066 104001      EMT    1                ;SYSTEM CONTROL REGISTER
                                ;WRITE/READ ERROR
003070 042713 000004      1$: BIC    #BIT02,(R3)        ;CLEAR LINE SELECT BIT 2
003074 001403           BEQ    2$
003076 005005      CLR    R5                ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER, 0
003100 011304      MOV    (R3),R4          ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
003102           HLT    1                ;SYSTEM CONTROL REGISTER
003102 104001      EMT    1                ;SYSTEM CONTROL REGISTER
                                ;WRITE/READ ERROR
003104 104400      2$: SCOPE                                ;CHECK FOR ITERATIONS, LOOP
232 003106          REGTS1  +/LINE SELECT BIT 3/,BIT03
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;SET LINE SELECT BIT 3 IN SYSTEM CONTROL REGISTER
                                ;VERIFY THAT LINE SELECT BIT 3 WAS SET
                                ;CLEAR LINE SELECT BIT 3
                                ;VERIFY THAT LINE SELECT BIT 3 WAS CLEARED

003106           TS \XN,4000,2$
003106 012767 000340 174662 T20: MOV    #340,PS                ;DISABLE ALL INTERRUPTS

```

```

003114 012767 004000 014734      MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
003122 012767 003172 014722      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
003130 000021      MOV    DHSCR,R3        ;PUT ADDRESS OF SYSTEM CONTROL REGISTER IN R3
003134 016703 014654      MOV    #BIT03,(R3)     ;SET LINE SELECT BIT 3
003140 012713 000010      CMP    #BIT03,(R3)     ;VERIFY THAT LINE SELECT BIT 3 WAS SET
003144 022713 0C0010      BEQ    1$
003146 001404 000010      MOV    #BIT03,R5
                                ;(R5)= EXPECTED VALUE
                                ;IN SYSTEM CONTROL REGISTER
                                ;LINE SELECT BIT 3
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
003152 011304      MOV    (R3),R4
003154      HLT    1
003154 104001      EMT    1
                                ;WRITE/READ ERROR
                                ;CLEAR LINE SELECT BIT 3
003156 042713 000010      1$:   BIC    #BIT03,(R3)
003162 001403      BEQ    2$
003164 005005      CLR    R5
                                ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER, 0
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
003166 011304      MOV    (R3),R4
003170      HLT    1
003170 104001      EMT    1
                                ;WRITE/READ ERROR
                                ;CHECK FOR ITERATIONS, LOOP
003172 104400      2$:   SCOPE
233 003174      REGTS1  +/MEMORY EXTENSION BIT 0/,BIT04
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;SET MEMORY EXTENSION BIT 0 IN SYSTEM CONTROL REGISTER
                                ;VERIFY THAT MEMORY EXTENSION BIT 0 WAS SET
                                ;CLEAR MEMORY EXTENSION BIT 0
                                ;VERIFY THAT MEMORY EXTENSION BIT 0 WAS CLEARED
003174      TS \XN,4000,2$
003174 012767 000340 174574      T21:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
003202 012767 004000 014646      MOV    #4000,ICOUNT   ;SET UP FOR 4000 ITERATIONS
003210 012767 003260 014634      MOV    #2$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
003216 000022      MOV    DHSCR,R3        ;PUT ADDRESS OF SYSTEM CONTROL REGISTER IN R3
003222 016703 014566      MOV    #BIT04,(R3)     ;SET MEMORY EXTENSION BIT 0
003226 012713 0C0020      CMP    #BIT04,(R3)     ;VERIFY THAT MEMORY EXTENSION BIT 0 WAS SET
003232 022713 000020      BEQ    1$
003234 001404 000020      MOV    #BIT04,R5
                                ;(R5)= EXPECTED VALUE
                                ;IN SYSTEM CONTROL REGISTER
                                ;MEMORY EXTENSION BIT 0
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
003240 011304      MOV    (R3),R4
003242      HLT    1
003242 104001      EMT    1
                                ;WRITE/READ ERROR
                                ;CLEAR MEMORY EXTENSION BIT 0
003244 042713 000020      1$:   BIC    #BIT04,(R3)

```

```

003250 001403      BEQ      2$
003252 005005      CLR      R5                ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER, 0
003254 011304      MOV      (R3),R4          ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
003256           HLT      1                ;SYSTEM CONTROL REGISTER
003256 104001      EMT      1                ;SYSTEM CONTROL REGISTER
                                ;WRITE/READ ERROR
003260 104400      2$: SCOPE                                ;CHECK FOR ITERATIONS, LOOP
234 003262          REGTS1  +/MEMORY EXTENSION BIT 1/,BIT05
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;SET MEMORY EXTENSION BIT 1 IN SYSTEM CONTROL REGISTER
                                ;VERIFY THAT MEMORY EXTENSION BIT 1 WAS SET
                                ;CLEAR MEMORY EXTENSION BIT 1
                                ;VERITY THAT MEMORY EXTENSION BIT 1 WAS CLEARED

003262           TS \XN,4000,2$
003262 012767 000340 174506 T22:  MOV      #340,PS                ;DISABLE ALL INTERRUPTS
003270 012767 004000 014560      MOV      #4000,ICOUNT          ;SET UP FOR 4000 ITERATIONS
003276 012767 003346 014546      MOV      #2$,ESCAPE          ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #,FREEZ1                ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
003304 000023      MOV      DHSCR,R3            ;PUT ADDRESS OF SYSTEM CONTROL REGISTER IN R3
003310 016703 014500      MOV      #BIT05,(R3)        ;SET MEMORY EXTENSION BIT 1
003314 012713 000040      CMP      #BIT05,(R3)        ;VERIFY THAT MEMORY EXTENSION BIT 1 WAS SET
003320 001404      BEQ      1$
003322 012705 000040      MOV      #BIT05,R5          ;(R5)= EXPECTED VALUE
                                ;IN SYSTEM CONTROL REGISTER
                                ;MEMORY EXTENSION BIT 1
003326 011304      MOV      (R3),R4          ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
003330           HLT      1                ;SYSTEM CONTROL REGISTER
003330 104001      EMT      1                ;SYSTEM CONTROL REGISTER
                                ;WRITE/READ ERROR
003332 042713 000040      1$: BIC      #BIT05,(R3)        ;CLEAR MEMORY EXTENSION BIT 1
003336 001403      BEQ      2$
003340 005005      CLR      R5                ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER, 0
003342 011304      MOV      (R3),R4          ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
003344           HLT      1                ;SYSTEM CONTROL REGISTER
003344 104001      EMT      1                ;SYSTEM CONTROL REGISTER
                                ;WRITE/READ ERROR
003346 104400      2$: SCOPE                                ;CHECK FOR ITERATIONS, LOOP
235 003350          REGTS1  +/RECEIVER INTERRUPT ENABLE/,BIT06
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;SET RECEIVER INTERRUPT ENABLE IN SYSTEM CONTROL REGISTER
                                ;VERIFY THAT RECEIVER INTERRUPT ENABLE WAS SET
                                ;CLEAR RECEIVER INTERRUPT ENABLE
                                ;VERITY THAT RECEIVER INTERRUPT ENABLE WAS CLEARED

003350           TS \XN,4000,2$
003350 012767 000340 174420 T23:  MOV      #340,PS                ;DISABLE ALL INTERRUPTS

```

```

003356 012767 004000 014472      MOV      #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
003364 012767 003434 014460      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
003372 000024 016703 014412      MOV      DHSCR,R3        ;PUT ADDRESS OF SYSTEM CONTROL REGISTER IN R3
003376 012713 000100      MOV      #BIT06,(R3)     ;SET RECEIVER INTERRUPT ENABLE
003402 022713 000100      CMP      #BIT06,(R3)     ;VERIFY THAT RECEIVER INTERRUPT ENABLE WAS SET
003406 001404      BEQ      1$
003410 012705 000100      MOV      #BIT06,R5       ;(R5)= EXPECTED VALUE
                                ;IN SYSTEM CONTROL REGISTER
                                ;RECEIVER INTERRUPT ENABLE
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
003414 011304      MOV      (R3),R4
003416      HLT      1
003416 104001      EMT      1
                                ;WRITE/READ ERROR
                                ;CLEAR RECEIVER INTERRUPT ENABLE
003420 042713 000100      1$:    BIC      #BIT06,(R3)
003424 001403      BEQ      2$
003426 005005      CLR      R5
                                ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER, 0
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
003430 011304      MOV      (R3),R4
003432      HLT      1
003432 104001      EMT      1
                                ;WRITE/READ ERROR
                                ;CHECK FOR ITERATIONS, LOOP
003434 104400      2$:    SCOPE
236 003436 REGTS1  +/MAINTENANCE MODE/,BIT09
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;SET MAINTENANCE MODE IN SYSTEM CONTROL REGISTER
                                ;VERIFY THAT MAINTENANCE MODE WAS SET
                                ;CLEAR MAINTENANCE MODE
                                ;VERIFY THAT MAINTENANCE MODE WAS CLEARED
003436      TS \XN,4000,2$
003436 012767 000340 174332 T24:   MOV      #340,PS        ;DISABLE ALL INTERRUPTS
003444 012767 004000 014404      MOV      #4000,ICOUNT    ;SET UP FOR 4000 ITERATIONS
003452 012767 003522 014372      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
003460 000025 016703 014324      MOV      DHSCR,R3        ;PUT ADDRESS OF SYSTEM CONTROL REGISTER IN R3
003464 012713 001000      MOV      #BIT09,(R3)     ;SET MAINTENANCE MODE
003470 022713 001000      CMP      #BIT09,(R3)     ;VERIFY THAT MAINTENANCE MODE WAS SET
003474 001404      BEQ      1$
003476 012705 001000      MOV      #BIT09,R5       ;(R5)= EXPECTED VALUE
                                ;IN SYSTEM CONTROL REGISTER
                                ;MAINTENANCE MODE
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
003502 011304      MOV      (R3),R4
003504      HLT      1
003504 104001      EMT      1
                                ;WRITE/READ ERROR
                                ;CLEAR MAINTENANCE MODE
003506 042713 001000      1$:    BIC      #BIT09,(R3)

```

```

003512 001403      BEQ      2$
003514 005005      CLR      R5                      ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER, 0
003516 011304      MOV      (R3),R4                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
003520 104001      HLT      1                      ;SYSTEM CONTROL REGISTER
003520 104001      EMT      1
                                ;WRITE/READ ERROR
003522 104400      2$:      SCOPE                ;CHECK FOR ITERATIONS, LOOP
237 003524      REGTS1  +/SILO OVERFLOW INTERRUPT ENABLE/,BIT12
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;SET SILO OVERFLOW INTERRUPT ENABLE IN SYSTEM CONTROL REGISTER
                                ;VERIFY THAT SILO OVERFLOW INTERRUPT ENABLE WAS SET
                                ;CLEAR SILO OVERFLOW INTERRUPT ENABLE
                                ;VERIFY THAT SILO OVERFLOW INTERRUPT ENABLE WAS CLEARED

003524 012767 000340 174244  TS \XN,4000,2$
003524 012767 000340 174244  T25:  MOV      #340,PS                ;DISABLE ALL INTERRUPTS
003532 012767 004000 014316      MOV      #4000,ICOUNT          ;SET UP FOR 4000 ITERATIONS
003540 012767 003610 014304      MOV      #2$,ESCAPE          ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #,FREEZ1                ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
003546 016703 014236      MOV      DHSCR,R3            ;PUT ADDRESS OF SYSTEM CONTROL REGISTER IN R3
003552 012713 010000      MOV      #BIT12,(R3)        ;SET SILO OVERFLOW INTERRUPT ENABLE
003556 022713 010000      CMP      #BIT12,(R3)        ;VERIFY THAT SILO OVERFLOW INTERRUPT ENABLE WAS SET
003562 001404      BEQ      1$
003564 012705 010000      MOV      #BIT12,R5          ;(R5)= EXPECTED VALUE
                                ;IN SYSTEM CONTROL REGISTER
                                ;SILO OVERFLOW INTERRUPT ENABLE
003570 011304      MOV      (R3),R4            ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
003572 104001      HLT      1                      ;SYSTEM CONTROL REGISTER
003572 104001      EMT      1
                                ;WRITE/READ ERROR
003574 042713 010000      1$:      BIC      #BIT12,(R3)    ;CLEAR SILO OVERFLOW INTERRUPT ENABLE
003600 001403      BEQ      2$
003602 005005      CLR      R5                      ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER, 0
003604 011304      MOV      (R3),R4            ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
003606 104001      HLT      1                      ;SYSTEM CONTROL REGISTER
003606 104001      EMT      1
                                ;WRITE/READ ERROR
003610 104400      2$:      SCOPE                ;CHECK FOR ITERATIONS, LOOP
238 003612      REGTS1  +/TRANSMITTER INTERRUPT ENABLE/,BIT13
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;SET TRANSMITTER INTERRUPT ENABLE IN SYSTEM CONTROL REGISTER
                                ;VERIFY THAT TRANSMITTER INTERRUPT ENABLE WAS SET
                                ;CLEAR TRANSMITTER INTERRUPT ENABLE
                                ;VERIFY THAT TRANSMITTER INTERRUPT ENABLE WAS CLEARED

003612 012767 000340 174156  TS \XN,4000,2$
003612 012767 000340 174156  T26:  MOV      #340,PS                ;DISABLE ALL INTERRUPTS

```

```

003620 012767 004000 014230      MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
003626 012767 003676 014216      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
003634 016703 014150      MOV    DHSCR,R3        ;PUT ADDRESS OF SYSTEM CONTROL REGISTER IN R3
003640 012713 020000      MOV    #BIT13,(R3)     ;SET TRANSMITTER INTERRUPT ENABLE
003644 022713 020000      CMP    #BIT13,(R3)     ;VERIFY THAT TRANSMITTER INTERRUPT ENABLE WAS SET
003650 001404                BEQ    1$
003652 012705 020000      MOV    #BIT13,R5       ;(R5)= EXPECTED VALUE
                                ;IN SYSTEM CONTROL REGISTER
                                ;TRANSMITTER INTERRUPT ENABLE
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
003656 011304                MOV    (R3),R4
003660                HLT    1
003660 104001                EMT    1
                                ;WRITE/READ ERROR
                                ;CLEAR TRANSMITTER INTERRUPT ENABLE
003662 042713 020000      1$:  BIC    #BIT13,(R3)
003666 001403                BEQ    2$
003670 005005                CLR    R5
                                ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER, 0
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
003672 011304                MOV    (R3),R4
003674                HLT    1
003674 104001                EMT    1
                                ;WRITE/READ ERROR
                                ;CHECK FOR ITERATIONS, LOOP
003676 104400      2$:  SCOPE
239 003700      REGTS1  +/TRANSMITTER DONE/,BIT15
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;SET TRANSMITTER DONE IN SYSTEM CONTROL REGISTER
                                ;VERIFY THAT TRANSMITTER DONE WAS SET
                                ;CLEAR TRANSMITTER DONE
                                ;VERIFY THAT TRANSMITTER DONE WAS CLEARED
003700                TS \XN,4000,2$
003700 012767 000340 174070      T27:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
003706 012767 004000 014142      MOV    #4000,ICOUNT    ;SET UP FOR 4000 ITERATIONS
003714 012767 003764 014130      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
003722 016703 014062      MOV    DHSCR,R3        ;PUT ADDRESS OF SYSTEM CONTROL REGISTER IN R3
003726 012713 100000      MOV    #BIT15,(R3)     ;SET TRANSMITTER DONE
003732 022713 100000      CMP    #BIT15,(R3)     ;VERIFY THAT TRANSMITTER DONE WAS SET
003736 001404                BEQ    1$
003740 012705 100000      MOV    #BIT15,R5       ;(R5)= EXPECTED VALUE
                                ;IN SYSTEM CONTROL REGISTER
                                ;TRANSMITTER DONE
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
003744 011304                MOV    (R3),R4
003746                HLT    1
003746 104001                EMT    1
                                ;WRITE/READ ERROR
                                ;CLEAR TRANSMITTER DONE
003750 042713 100000      1$:  BIC    #BIT15,(R3)

```

```

003754 001403      BEQ      2$
003756 005005      CLR      R5                ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER, 0
003760 011304      MOV      (R3),R4          ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
003762              HLT      1                ;SYSTEM CONTROL REGISTER
003762 104001      EMT      1
                                ;WRITE/READ ERROR
003764 104400      2$:      SCOPE                ;CHECK FOR ITERATIONS, LOOP
240 003766      REGTS2  †/CHARACTER AVAILABLE/,BIT07
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;VERIFY THAT CHARACTER AVAILABLE IS READ ONLY IN NORMAL MODE

003766              TS \XN,4000,1$
003766 012767 000340 174002 T30:  MOV      #340,PS                ;DISABLE ALL INTERRUPTS
003774 012767 004000 014054      MOV      #4000,ICOUNT          ;SET UP FOR 4000 ITERATIONS
004002 012767 004040 014042      MOV      #1$,ESCAPE          ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #,FREEZ1                ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
004010 000031      00200 013772      MOV      #BIT07,@DHSCR          ;ATTEMPT TO WRITE
                                ;CHARACTER AVAILABLE IN
                                ;SYSTEM CONTROL REGISTER
                                ;WAS CHARACTER AVAILABLE SET
004016 005777 013766      TST      @DHSCR
004022 001406      BEQ      1$
004024 005005      CLR      R5                ;(R5)=EXPECTED DATA
                                ;IN SYSTEM CONTROL REGISTER, 0
004026 017704 013756      MOV      @DHSCR,R4          ;(R4)=ACTUAL DATA IN SYSTEM
                                ;CONTROL REGISTER
004032 016703 013752      MOV      DHSCR,R3          ;ADDRESS OF SYSTEM CONTROL REGISTER
004036              HLT      1                ;SYSTEM CONTROL REGISTER
004036 104001      EMT      1
                                ;WRITE/READ ERROR
004040 104400      1$:      SCOPE                ;CHECK FOR ITERATIONS, LOOP
241 004042      REGTS2  †/CLEAR NON EXISTANT MEMORY/,BIT08
                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;VERIFY THAT CLEAR      NON EXISTANT MEMORY IS READ ONLY IN NORMAL MODE

004042              TS \XN,4000,1$
004042 012767 000340 173726 T31:  MOV      #340,PS                ;DISABLE ALL INTERRUPTS
004050 012767 004000 014000      MOV      #4000,ICOUNT          ;SET UP FOR 4000 ITERATIONS
004056 012767 004114 013766      MOV      #1$,ESCAPE          ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #,FREEZ1                ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
004064 000032      000400 013716      MOV      #BIT08,@DHSCR          ;ATTEMPT TO WRITE
                                ;CLEAR NON EXISTANT MEMORY IN
                                ;SYSTEM CONTROL REGISTER
                                ;WAS CLEAR      NON EXISTANT MEMORY SET
004072 005777 013712      TST      @DHSCR
004076 001406      BEQ      1$
004100 005005      CLR      R5                ;(R5)=EXPECTED DATA
                                ;IN SYSTEM CONTROL REGISTER, 0
004102 017704 013702      MOV      @DHSCR,R4          ;(R4)=ACTUAL DATA IN SYSTEM

```

```

004106 016703 013676      MOV      DHSCR,R3      ;CONTROL REGISTER
004112                      HLT      1              ;ADDRESS OF SYSTEM CONTROL REGISTER
004112 104001              EMT      1              ;SYSTEM CONTROL REGISTER

                                ;WRITE/READ ERROR

004114 104400              1$:      SCOPE
242 004116                      REGTS2  †/NON EXISTANT MEMORY/.BIT10

                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;VERIFY THAT NON EXISTANT MEMORY IS READ ONLY IN NORMAL MODE

004116                      TS \XN,4000,1$
004116 012767 000340 173652 T32:    MOV      #340,PS      ;DISABLE ALL INTERRUPTS
004124 012767 004000 013724      MOV      #4000,ICOUNT  ;SET UP FOR 4000 ITERATIONS
004132 012767 004170 013712      MOV      #1$,ESCAPE   ;SET UP TO ESCAPE TO NEXT TEST

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

004140 000033 012777 002000 013642      MOV      #BIT10,@DHSCR ;ATTEMPT TO WRITE
                                ;NON EXISTANT MEMORY IN
                                ;SYSTEM CONTROL REGISTER
                                ;WAS NON EXISTANT MEMORY SET

004146 005777 013636      TST      @DHSCR
004152 001406              BEQ      1$
004154 005005              CLR      R5              ;(R5)=EXPECTED DATA
                                ;IN SYSTEM CONTROL REGISTER, 0
                                ;(R4)=ACTUAL DATA IN SYSTEM
                                ;CONTROL REGISTER
                                ;ADDRESS OF SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER

004156 017704 013626      MOV      @DHSCR,R4
                                ;WRITE/READ ERROR

004162 016703 013622      MOV      DHSCR,R3
004166                      HLT      1
004166 104001              EMT      1

004170 104400              1$:      SCOPE
243 004172                      REGTS2  †/MASTER CLEAR/,BIT11

                                ;SYSTEM CONTROL REGISTER WRITE/READ TEST (NORMAL MODE)
                                ;VERIFY THAT MASTER CLEAR IS READ ONLY IN NORMAL MODE

004172                      TS \XN,4000,1$
004172 012767 000340 173576 T33:    MOV      #340,PS      ;DISABLE ALL INTERRUPTS
004200 012767 004000 013650      MOV      #4000,ICOUNT  ;SET UP FOR 4000 ITERATIONS
004206 012767 004244 013636      MOV      #1$,ESCAPE   ;SET UP TO ESCAPE TO NEXT TEST

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

004214 000034 012777 004000 013566      MOV      #BIT11,@DHSCR ;ATTEMPT TO WRITE
                                ;MASTER CLEAR IN
                                ;SYSTEM CONTROL REGISTER
                                ;WAS MASTER CLEAR SET

004222 005777 013562      TST      @DHSCR
004226 001406              BEQ      1$
004230 005005              CLR      R5              ;(R5)=EXPECTED DATA
                                ;IN SYSTEM CONTROL REGISTER, 0
                                ;(R4)=ACTUAL DATA IN SYSTEM
                                ;CONTROL REGISTER
                                ;ADDRESS OF SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER

004232 017704 013552      MOV      @DHSCR,R4

004236 016703 013546      MOV      DHSCR,R3
004242                      HLT      1

```









```

004652 011304          MOV      (R3),R4          ;MAINTENANCE MODE AND SILO OVERFLOW
004654          HLT      1          ;(R4)=ACTUAL DATA IN
004654 104001          EMT      1          ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
004656 042713 001000    1$:   BIC      #BIT09,(R3)      ;WRITE/READ ERROR
004662 042713 040000    BIC      #BIT14,(R3)      ;CLEAR MAINTENANCE MODE
004666 022713 040000    CMP      #BIT14,(R3)      ;ATTEMPT TO CLEAR SILO OVERFLOW
004672 001403          BEQ      2$                ;SILO OVERFLOW SHOULD BE SET
004674 012705 040000    MOV      #BIT14,R5        ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SILO OVERFLOW
                                ;SYSTEM CONTROL REGISTER
004700          HLT      1
004700 104001          EMT      1
                                ;WRITE/READ ERROR
004702 052713 001000    2$:   BIS      #BIT09,(R3)      ;SET MAINTENANCE MODE
004706 042713 040000    BIC      #BIT14,(R3)      ;CLEAR SILO OVERFLOW
004712 022713 001000    CMP      #BIT09,(R3)      ;EXPECT ONLY MAINTENANCE
                                ;MODE TO BE SET
004716 001404          BEQ      3$
004720 012705 001000    MOV      #BIT09,R5        ;(R5)=EXPECTED DATA IN
                                ;SYSTEM CONTROL REGISTER,
                                ;MAINTENANCE MODE BIT
                                ;(R4)=ACTUAL DATA IN
                                ;SYSTEM CONTROL REGISTER
                                ;SYSTEM CONTROL REGISTER
004724 011304          MOV      (R3),R4
004726          HLT      1
004726 104001          EMT      1
                                ;WRITE/READ ERROR
004730 104400          3$:   SCOPE                ;CHECK FOR ITERATIONS, LOOP
249      000000    XBIT=BITX
250      000020    CBIT=BITC
251      000001    BITX=1
252      000000    BITC=0
254      000017    .REPT 15.
255      REGTS6  +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC
256      .IF EQ BITX-4
257      .NLIST
258      BITX=BITX+BITX
259      BITC=BITC+1
260      .LIST
261      .ENDC
262      .NLIST
263      BITX=BITX+BITX
264      BITC=BITC+1
265      .LIST
266      .ENDR
004732          REGTS6  +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC
                                ;LINE PARAMETER REGISTER DATA TEST
                                ;SET BIT 0 IN LINE PARAMETER TO 1
                                ;VERIFY THAT BIT 0 WAS SET
                                ;CLEAR BIT 0
                                ;VERIFY THAT BIT 0 WAS CLEARED
004732          TS \XN,4000,2$
004732 012767 000340 173036 T40:   MOV      #340,PS          ;DISABLE ALL INTERRUPTS

```

```

004740 012767 004000 013110      MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
004746 012767 005020 013076      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
004754 000041 004000 013026      MOV    #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
004762 016703 013026 013026      MOV    DHLPR,R3       ;SET UP POINTER TO LINE PARAMETER
004766 012705 000001      MOV    #1,R5          ;BIT 0 WILL BE SET IN LINE PARAMETER
004772 010513      MOV    R5,(R3)        ;SET BIT 0
004774 011304      MOV    (R3),R4        ;GET CONTENTS OF LINE PARAMETER
                                .IIF NB  <>,
                                BIC    #,R4          ;CLEAR UNWANTED BITS
                                CMP    R5,R4        ;WAS BIT 0 SET
                                BEQ    1$
004776 020504      .IIF IDN  <DHLPR>,<DHLPR>,  HLT    2          ;LINE PARAMETER REGI
005000 001401      EMT    2
005002      .IIF IDN  <DHLPR>,<DHBCR>,  HLT    3          ;LINE PARAMETER REGI
                                .IIF IDN  <DHLPR>,<DHSSR>,  HLT    4          ;LINE PARAMETER REGI
STER ERROR 005002 104002
                                .IIF IDN  <DHLPR>,<DHBCR>,  HLT    3          ;LINE PARAMETER REGI
STER ERROR 005004 040513
005006 011304      1$:    BIC    R5,(R3)   ;CLEAR BIT 0
                                MOV    (R3),R4     ;READ CONTENTS OF LINE PARAMETER
                                .IIF NB  <>,
                                BIC    #,R4        ;CLEAR UNWANTED BITS
                                TST    R4          ;WAS BIT 0 CLEARED
                                BEQ    2$
                                CLR    R5
005010 005704      .IIF IDN  <DHLPR>,<DHLPR>,  HLT    2          ;LINE PARAMETER REGI
005012 001402      EMT    2
005014 005005      .IIF IDN  <DHLPR>,<DHBCR>,  HLT    3          ;LINE PARAMETER REGI
005016      .IIF IDN  <DHLPR>,<DHSSR>,  HLT    4          ;LINE PARAMETER REGI
STER ERROR 005016 104002
STER ERROR 005020 104400
                                2$:    SCOPE
                                .IF EQ  BITX-4
                                .NLIST
                                BITX=BITX+BITX
                                BITC=BITC+1
                                .LIST
                                .ENDC
                                BITX=BITX+BITX
                                BITC=BITC+1
                                REGTS6  +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC
                                ;LINE PARAMETER REGISTER DATA TEST
                                ;SET BIT 1 IN LINE PARAMETER TO 1
                                ;VERIFY THAT BIT 1 WAS SET
                                ;CLEAR BIT 1
                                ;VERIFY THAT BIT 1 WAS CLEARED
005022 000002
000001
005022 012767 000340 172746      TS \XN,4000,2$
005022 012767 004000 013020      T41:  MOV    #340,PS    ;DISABLE ALL INTERRUPTS
005030 012767 004000 013020      MOV    #4000,ICOUNT   ;SET UP FOR 4000 ITERATIONS
005036 012767 005110 013006      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
005044 000042 004000 012736      MOV    #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
005052 016703 012736 012736      MOV    DHLPR,R3       ;SET UP POINTER TO LINE PARAMETER
005056 012705 000002      MOV    #2,R5          ;BIT 1 WILL BE SET IN LINE PARAMETER
005062 010513      MOV    R5,(R3)        ;SET BIT 1

```

```

005064 011304          MOV      (R3),R4          ;GET CONTENTS OF LINE PARAMETER
.IIF NB <>,          BIC      #,R4          ;CLEAR UNWANTED BITS
005066 020504          CMP      R5,R4          ;WAS BIT 1 SET
005070 001401          BEQ      1$
005072          .IIF IDN <DHLPR>,<DHLPR>, HLT      2          ;LINE PARAMETER REGI
STER ERROR
005072 104002          EMT      2
.IIF IDN <DHLPR>,<DHBCR>, HLT      3          ;LINE PARAMETER REGI
STER ERROR
.IIF IDN <DHLPR>,<DHSSR>, HLT      4          ;LINE PARAMETER REGI
STER ERROR
005074 040513          1$:      BIC      R5,(R3)          ;CLEAR BIT 1
005076 011304          MOV      (R3),R4          ;READ CONTENTS OF LINE PARAMETER
.IIF NB <>,          BIC      #,R4          ;CLEAR UNWANTED BITS
005100 005704          R4          ;WAS BIT 1 CLEARED
005102 001402          BEQ      2$
005104 005005          CLR      R5
005106          .IIF IDN <DHLPR>,<DHLPR>, HLT      2          ;LINE PARAMETER REGI
STER ERROR
005106 104002          EMT      2
.IIF IDN <DHLPR>,<DHBCR>, HLT      3          ;LINE PARAMETER REGI
STER ERROR
.IIF IDN <DHLPR>,<DHSSR>, HLT      4          ;LINE PARAMETER REGI
STER ERROR
005110 104400          2$:      SCOPE
.IF EQ BITX-4
.NLIST
BITX=BITX+BITX
BITC=BITC+1
.LIST
.ENDC
000004          BITX=BITX+BITX
000002          BITC=BITC+1
005112          REGTS6 †/DHLPR/,\BITX,†/LINE PARAMETER/,\BITC

;LINE PARAMETER REGISTER DATA TEST
;SET BIT 2 IN LINE PARAMETER TO 1
;VERIFY THAT BIT 2 WAS SET
;CLEAR BIT 2
;VERIFY THAT BIT 2 WAS CLEARED

005112          TS \XN,4000,2$
005112 012767 000340 172656 T42:      MOV      #340,PS          ;DISABLE ALL INTERRUPTS
005120 012767 004000 012730      MOV      #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
005126 012767 005200 012716      MOV      #2$,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST
.IF NB <>
MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
.ENDC
XN=XN+1
005134 000043          MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
005142 016703 012646          MOV      DHLPR,R3          ;SET UP POINTER TO LINE PARAMETER
005146 012705 000004          MOV      #4,R5            ;BIT 2 WILL BE SET IN LINE PARAMETER
005152 010513          MOV      R5,(R3)          ;SET BIT 2
005154 011304          MOV      (R3),R4          ;GET CONTENTS OF LINE PARAMETER
.IIF NB <>,          BIC      #,R4          ;CLEAR UNWANTED BITS
005156 020504          CMP      R5,R4          ;WAS BIT 2 SET
005160 001401          BEQ      1$
005162          .IIF IDN <DHLPR>,<DHLPR>, HLT      2          ;LINE PARAMETER REGI
STER ERROR
005162 104002          EMT      2
.IIF IDN <DHLPR>,<DHBCR>, HLT      3          ;LINE PARAMETER REGI
STER ERROR
.IIF IDN <DHLPR>,<DHSSR>, HLT      4          ;LINE PARAMETER REGI
STER ERROR
005164 040513          1$:      BIC      R5,(R3)          ;CLEAR BIT 2
005166 011304          MOV      (R3),R4          ;READ CONTENTS OF LINE PARAMETER

```

```

005170 005704      .IIF NB <>,      BIC    #,R4      ;CLEAR UNWANTED BITS
005172 001402      TST      R4      ;WAS BIT 2 CLEARED
005174 005005      BEQ      2$
005176      CLR      R5
STER ERROR      .IIF IDN      <DHLPR>,<DHLPR>,      HLT    2      ;LINE PARAMETER REGI
005176 104002      EMT      2
STER ERROR      .IIF IDN      <DHLPR>,<DHBCR>,      HLT    3      ;LINE PARAMETER REGI
STER ERROR      .IIF IDN      <DHLPR>,<DHSSR>,      HLT    4      ;LINE PARAMETER REGI
005200 104400      2$:      SCOPE
000010      .IF EQ BITX-4
000003      BITX=BITX+BITX
000020      BITC=BITC+1
000004      .ENDC
005202      BITX=BITX+BITX
000004      BITC=BITC+1
REGTS6 †/DHLPR/,\BITX,†/LINE PARAMETER/,\BITC
;LINE PARAMETER REGISTER DATA TEST
;SET BIT 4 IN LINE PARAMETER TO 1
;VERIFY THAT BIT 4 WAS SET
;CLEAR BIT 4
;VERIFY THAT BIT 4 WAS CLEARED

005202      TS \XN,4000,2$
005202 012767 000340 172566 T43:      MOV    #340,FS      ;DISABLE ALL INTERRUPTS
005210 012767 004000 012640      MOV    #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
005216 012767 005270 012626      MOV    #2$,ESCAPE   ;SET UP TO ESCAPE TO NEXT TEST
.IIF NB <>
MOV    #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
.ENDC
XN=XN+1
005224 012777 004000 012556      MOV    #BIT11,0DHSCR ;MASTER CLEAR INTERFACE
005232 016703 012556      MOV    DHLPR,R3      ;SET UP POINTER TO LINE PARAMETER
005236 012705 000020      MOV    #20,R5        ;BIT 4 WILL BE SET IN LINE PARAMETER
005242 010513      MOV    R5,(R3)       ;SET BIT 4
005244 011304      MOV    (R3),R4       ;GET CONTENTS OF LINE PARAMETER
.IIF NB <>,      BIC    #,R4      ;CLEAR UNWANTED BITS
CMP    R5,R4          ;WAS BIT 4 SET
BEQ    1$
STER ERROR      .IIF IDN      <DHLPR>,<DHLPR>,      HLT    2      ;LINE PARAMETER REGI
005252 104002      EMT      2
STER ERROR      .IIF IDN      <DHLPR>,<DHBCR>,      HLT    3      ;LINE PARAMETER REGI
STER ERROR      .IIF IDN      <DHLPR>,<DHSSR>,      HLT    4      ;LINE PARAMETER REGI
005254 040513      1$:      BIC    R5,(R3)     ;CLEAR BIT 4
005256 011304      MOV    (R3),R4       ;READ CONTENTS OF LINE PARAMETER
.IIF NB <>,      BIC    #,R4      ;CLEAR UNWANTED BITS
TST    R4            ;WAS BIT 4 CLEARED
BEQ    2$
STER ERROR      .IIF IDN      <DHLPR>,<DHLPR>,      HLT    2      ;LINE PARAMETER REGI
005266 104002      EMT      2
STER ERROR      .IIF IDN      <DHLPR>,<DHBCR>,      HLT    3      ;LINE PARAMETER REGI
STER ERROR      .IIF IDN      <DHLPR>,<DHSSR>,      HLT    4      ;LINE PARAMETER REGI
005270 104400      2$:      SCOPE
.IF EQ BITX-4
.NLIST
BITX=BITX+BITX

```





```

;CLEAR BIT 6
;VERIFY THAT BIT 6 WAS CLEARED

005362      005362 012767 000340 172406 TS \XN,4000,2$
005370      012767 004000 012460 T45:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
005376      012767 005450 012446      MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
;SET UP TO ESCAPE TO NEXT TEST
      MOV    #2$,ESCAPE
      .IF NB <>
      MOV    #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
005404      000046 005404 012777 004000 012376      MOV    #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
005412      016703 005412 016703      MOV    DHLPR,R3          ;SET UP POINTER TO LINE PARAMETER
005416      012705 005416 012705      MOV    #100,R5          ;BIT 6 WILL BE SET IN LINE PARAMETER
005422      010513 005422 010513      MOV    R5,(R3)          ;SET BIT 6
005424      011304 005424 011304      MOV    (R3),R4          ;GET CONTENTS OF LINE PARAMETER
;CLEAR UNWANTED BITS
;WAS BIT 6 SET
005426      020504 005426 020504      .IIF NB <>, BIC    #,R4
005430      001401 005430 001401      CMP    R5,R4
005432      005432 005432 005432      BEQ    1$
      .IIF IDN <DHLPR>,<DHLPR>, HLT    2      ;LINE PARAMETER REGI
      EMT    2
      .IIF IDN <DHLPR>,<DHBCR>, HLT    3      ;LINE PARAMETER REGI
      .IIF IDN <DHLPR>,<DHSSR>, HLT    4      ;LINE PARAMETER REGI
005434      040513 005434 040513      1$:   BIC    R5,(R3)      ;CLEAR BIT 6
005436      011304 005436 011304      MOV    (R3),R4          ;READ CONTENTS OF LINE PARAMETER
;CLEAR UNWANTED BITS
;WAS BIT 6 CLEARED
005440      005704 005440 005704      .IIF NB <>, BIC    #,R4
005442      001402 005442 001402      TST    R4
005444      005005 005444 005005      BEQ    2$
005446      005446 005446 005446      CLR    R5
      .IIF IDN <DHLPR>,<DHLPR>, HLT    2      ;LINE PARAMETER REGI
      EMT    2
      .IIF IDN <DHLPR>,<DHBCR>, HLT    3      ;LINE PARAMETER REGI
      .IIF IDN <DHLPR>,<DHSSR>, HLT    4      ;LINE PARAMETER REGI
005450      104400 005450 104400      2$:   SCOPE
      .IF EQ BITX-4
      .NLIST
      BITX=BITX+BITX
      BITC=BITC+1
      .LIST
      .ENDC
      BITX=BITX+BITX
      BITC=BITC+1
005452      000200 005452 000200      REGTS6 †/DHLPR/,\BITX,†/LINE PARAMETER/,\BITC
      000007
;LINE PARAMETER REGISTER DATA TEST
;SET BIT 7 IN LINE PARAMETER TO 1
;VERIFY THAT BIT 7 WAS SET
;CLEAR BIT 7
;VERIFY THAT BIT 7 WAS CLEARED

005452      005452 012767 000340 172316 TS \XN,4000,2$
005460      012767 005460 012767 004000 012370 T46:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
005466      012767 005466 012767 005540 012356      MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
;SET UP TO ESCAPE TO NEXT TEST
      MOV    #2$,ESCAPE
      .IF NB <>
      MOV    #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC

```

```

005474 000047 004000 012306 XN=XN+1
005502 012777 012306      MOV #BIT11, @DHSCR ;MASTER CLEAR INTERFACE
005506 016703 000200      MOV DHLPR, R3 ;SET UP POINTER TO LINE PARAMETER
005512 012705      MOV #200, R5 ;BIT 7 WILL BE SET IN LINE PARAMETER
005514 010513      MOV R5, (R3) ;SET BIT 7
005514 011304      MOV (R3), R4 ;GET CONTENTS OF LINE PARAMETER
;CLEAR UNWANTED BITS
;WAS BIT 7 SET
.IIF NB <>, BIC #, R4
005516 020504      CMP R5, R4
005520 001401      BEQ 1$
005522      .IIF IDN <DHLPR>, <DHLPR>, HLT 2 ;LINE PARAMETER REGI
STER ERROR
005522 104002      EMT 2
.IIF IDN <DHLPR>, <DHBCR>, HLT 3 ;LINE PARAMETER REGI
STER ERROR
.IIF IDN <DHLPR>, <DHSSR>, HLT 4 ;LINE PARAMETER REGI
STER ERROR
005524 040513      1$: BIC R5, (R3) ;CLEAR BIT 7
005526 011304      MOV (R3), R4 ;READ CONTENTS OF LINE PARAMETER
.IIF NB <>, BIC #, R4 ;CLEAR UNWANTED BITS
;WAS BIT 7 CLEARED
005530 005704      TST R4
005532 001402      BEQ 2$
005534 005005      CLR R5
005536      .IIF IDN <DHLPR>, <DHLPR>, HLT 2 ;LINE PARAMETER REGI
STER ERROR
005536 104002      EMT 2
.IIF IDN <DHLPR>, <DHBCR>, HLT 3 ;LINE PARAMETER REGI
STER ERROR
.IIF IDN <DHLPR>, <DHSSR>, HLT 4 ;LINE PARAMETER REGI
STER ERROR
005540 104400      2$: SCOPE
.IF EQ BITX-4
.NLIST
BITX=BITX+BITX
BITC=BITC+1
.LIST
.ENDC
BITX=BITX+BITX
BITC=BITC+1
005542 000400      REGTS6 †/DHLPR/, \BITX, †/LINE PARAMETER/, \BITC
000010
;LINE PARAMETER REGISTER DATA TEST
;SET BIT 10 IN LINE PARAMETER TO 1
;VERIFY THAT BIT 10 WAS SET
;CLEAR BIT 10
;VERIFY THAT BIT 10 WAS CLEARED
005542 TS \XN, 4000, 2$
005542 012767 000340 172226 T47: MOV #340, PS ;DISABLE ALL INTERRUPTS
005550 012767 004000 012300      MOV #4000, ICOUNT ;SET UP FOR 4000 ITERATIONS
005556 012767 005630 012266      MOV #2$, ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
.IIF NB <>
MOV #, FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1
005564 000050 004000 012216      MOV #BIT11, @DHSCR ;MASTER CLEAR INTERFACE
005572 012777 012216      MOV DHLPR, R3 ;SET UP POINTER TO LINE PARAMETER
005576 012705 000400      MOV #400, R5 ;BIT 10 WILL BE SET IN LINE PARAMETER
005602 010513      MOV R5, (R3) ;SET BIT 10
005604 011304      MOV (R3), R4 ;GET CONTENTS OF LINE PARAMETER
;CLEAR UNWANTED BITS
;WAS BIT 10 SET
.IIF NB <>, BIC #, R4
005606 020504      CMP R5, R4
005610 001401      BEQ 1$
005612      .IIF IDN <DHLPR>, <DHLPR>, HLT 2 ;LINE PARAMETER REGI
STER ERROR

```

```

005612 104002          EMT      2
STER ERROR          .IIF IDN    <DHLPR>,<DHBCR>,      HLT      3          ;LINE PARAMETER REGI
                                .IIF IDN    <DHLPR>,<DHSSR>,      HLT      4          ;LINE PARAMETER REGI
STER ERROR          1$:      BIC      R5,(R3)          ;CLEAR BIT 10
005614 040513          MOV      (R3),R4          ;READ CONTENTS OF LINE PARAMETER
005616 011304          .IIF NB  <>,      BIC      #,R4          ;CLEAR UNWANTED BITS
                                TST      R4          ;WAS BIT 10 CLEARED
005620 005704          BEQ      2$
005622 001402          CLR      R5
005624 005005          .IIF IDN    <DHLPR>,<DHLPR>,      HLT      2          ;LINE PARAMETER REGI
005626          EMT      2
STER ERROR          .IIF IDN    <DHLPR>,<DHBCR>,      HLT      3          ;LINE PARAMETER REGI
                                .IIF IDN    <DHLPR>,<DHSSR>,      HLT      4          ;LINE PARAMETER REGI
STER ERROR          2$:      SCOPE
005630 104400          .IF EQ  BITX-4
                                .NLIST
                                BITX=BITX+BITX
                                BITC=BITC+1
                                .LIST
                                .ENDC
                                BITX=BITX+BITX
                                BITC=BITC+1
005632 104100          REGTS6  +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC
                                001000
                                000011
                                ;LINE PARAMETER REGISTER DATA TEST
                                ;SET BIT 11 IN LINE PARAMETER TO 1
                                ;VERIFY THAT BIT 11 WAS SET
                                ;CLEAR BIT 11
                                ;VERIFY THAT BIT 11 WAS CLEARED

005632          TS \XN,4000,2$
005632 012767 000340 172136 T50:      MOV      #340,PS          ;DISABLE ALL INTERRUPTS
005640 012767 004000 012210      MOV      #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
005646 012767 005720 012176      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
005654 000051          MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
005662 012777 004000 012126      MOV      DHLPR,R3          ;SET UP POINTER TO LINE PARAMETER
005666 012705 001000          MOV      #1000,R5          ;BIT 11 WILL BE SET IN LINE PARAMETER
005672 010513          MOV      R5,(R3)          ;SET BIT 11
005674 011304          MOV      (R3),R4          ;GET CONTENTS OF LINE PARAMETER
                                .IIF NB  <>,      BIC      #,R4          ;CLEAR UNWANTED BITS
                                CMP      R5,R4          ;WAS BIT 11 SET
005676 020504          BEQ      1$
005700 001401          .IIF IDN    <DHLPR>,<DHLPR>,      HLT      2          ;LINE PARAMETER REGI
005702          EMT      2
STER ERROR          .IIF IDN    <DHLPR>,<DHBCR>,      HLT      3          ;LINE PARAMETER REGI
                                .IIF IDN    <DHLPR>,<DHSSR>,      HLT      4          ;LINE PARAMETER REGI
STER ERROR          1$:      BIC      R5,(R3)          ;CLEAR BIT 11
005704 040513          MOV      (R3),R4          ;READ CONTENTS OF LINE PARAMETER
005706 011304          .IIF NB  <>,      BIC      #,R4          ;CLEAR UNWANTED BITS
                                TST      R4          ;WAS BIT 11 CLEARED
005710 005704          BEQ      2$
005712 001402          CLR      R5
005714 005005          .IIF IDN    <DHLPR>,<DHLPR>,      HLT      2          ;LINE PARAMETER REGI
005716          STER ERROR

```

```

005716 104002
STER ERROR          EMT      2
                   .IIF IDN  <DHLPR>,<DHBCR> ,      HLT      3          ;LINE PARAMETER REGI
                   .IIF IDN  <DHLPR>,<DHSSR> ,      HLT      4          ;LINE PARAMETER REGI
005720 104400
2$: SCOPE
.IF EQ BITX-4
.NLIST
BITX=BITX+BITX
BITC=BITC+1
.LIST
.ENDC
002000
000012
005722              BITX=BITX+BITX
                   BITC=BITC+1
                   REGTS6 ↑/DHLPR/,\BITX,↑/LINE PARAMETER/,\BITC

                   ;LINE PARAMETER REGISTER DATA TEST
                   ;SET BIT 12 IN LINE PARAMETER TO 1
                   ;VERIFY THAT BIT 12 WAS SET
                   ;CLEAR BIT 12
                   ;VERIFY THAT BIT 12 WAS CLEARED

005722              TS \XN,4000,2$
005722 012767 000340 172046 T51: MOV #340,PS ;DISABLE ALL INTERRUPTS
005730 012767 004000 012120 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
005736 012767 006010 012106 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
                   .IF NB <>
                   MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
                   .ENDC
                   XN=XN+1
005744 012777 004000 012036 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
005752 016703 012036 MOV DHLPR,R3 ;SET UP POINTER TO LINE PARAMETER
005756 012705 002000 MOV #2000,R5 ;BIT 12 WILL BE SET IN LINE PARAMETER
005762 010513 MOV R5,(R3) ;SET BIT 12
005764 011304 MOV (R3),R4 ;GET CONTENTS OF LINE PARAMETER
                   .IIF NB <> , BIC #,R4 ;CLEAR UNWANTED BITS
005766 020504 CMP R5,R4 ;WAS BIT 12 SET
005770 001401 BEQ 1$
005772              .IIF IDN <DHLPR>,<DHLPR> ,      HLT      2          ;LINE PARAMETER REGI
STER ERROR          EMT      2
005772 104002          .IIF IDN  <DHLPR>,<DHBCR> ,      HLT      3          ;LINE PARAMETER REGI
STER ERROR          .IIF IDN  <DHLPR>,<DHSSR> ,      HLT      4          ;LINE PARAMETER REGI
STER ERROR
005774 040513 1$: BIC R5,(R3) ;CLEAR BIT 12
005776 011304 MOV (R3),R4 ;READ CONTENTS OF LINE PARAMETER
                   .IIF NB <> , BIC #,R4 ;CLEAR UNWANTED BITS
006000 005704 TST R4 ;WAS BIT 12 CLEARED
006002 001402 BEQ 2$
006004 005005 CLR R5
006006              .IIF IDN <DHLPR>,<DHLPR> ,      HLT      2          ;LINE PARAMETER REGI
STER ERROR
006006 104002          EMT      2
STER ERROR          .IIF IDN  <DHLPR>,<DHBCR> ,      HLT      3          ;LINE PARAMETER REGI
STER ERROR          .IIF IDN  <DHLPR>,<DHSSR> ,      HLT      4          ;LINE PARAMETER REGI
STER ERROR
006010 104400
2$: SCOPE
.IF EQ BITX-4
.NLIST
BITX=BITX+BITX
BITC=BITC+1
.LIST
.ENDC

```

```

004000 BITX=BITX+BITX
000013 BITC=BITC+1
006012 REGTS6 †/DHLPR/,\BITX,†/LINE PARAMETER/,\BITC

;LINE PARAMETER REGISTER DATA TEST
;SET BIT 13 IN LINE PARAMETER TO 1
;VERIFY THAT BIT 13 WAS SET
;CLEAR BIT 13
;VERIFY THAT BIT 13 WAS CLEARED

006012 TS \XN,4000,2‡
006012 012767 000340 171756 T52: MOV #340,PS ;DISABLE ALL INTERRUPTS
006020 012767 004000 012030 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
006026 012767 006100 012016 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
006034 000053 004000 011746 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
006042 016703 011746 MOV DHLPR,R3 ;SET UP POINTER TO LINE PARAMETER
006046 012705 004000 MOV #4000,R5 ;BIT 13 WILL BE SET IN LINE PARAMETER
006052 010513 MOV R5,(R3) ;SET BIT 13
006054 011304 MOV (R3),R4 ;GET CONTENTS OF LINE PARAMETER
.IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
006056 020504 CMP R5,R4 ;WAS BIT 13 SET
006060 001401 BEQ 1‡
006062 .IIF IDN <DHLPR>,<DHLPR>, HLT 2 ;LINE PARAMETER REGI
STER ERROR
006062 104002 EMT 2
.IIF IDN <DHLPR>,<DHBCR>, HLT 3 ;LINE PARAMETER REGI
STER ERROR
.IIF IDN <DHLPR>,<DHSSR>, HLT 4 ;LINE PARAMETER REGI
STER ERROR
006064 040513 1‡: BIC R5,(R3) ;CLEAR BIT 13
006066 011304 MOV (R3),R4 ;READ CONTENTS OF LINE PARAMETER
.IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
006070 005704 TST R4 ;WAS BIT 13 CLEARED
006072 001402 BEQ 2‡
006074 005005 CLR R5
006076 .IIF IDN <DHLPR>,<DHLPR>, HLT 2 ;LINE PARAMETER REGI
STER ERROR
006076 104002 EMT 2
.IIF IDN <DHLPR>,<DHBCR>, HLT 3 ;LINE PARAMETER REGI
STER ERROR
.IIF IDN <DHLPR>,<DHSSR>, HLT 4 ;LINE PARAMETER REGI
STER ERROR
006100 104400 2‡: SCOPE
.IF EQ BITX-4
.NLIST
BITX=BITX+BITX
BITC=BITC+1
.LIST
.ENDC
010000 BITX=BITX+BITX
000014 BITC=BITC+1
006102 REGTS6 †/DHLPR/,\BITX,†/LINE PARAMETER/,\BITC

;LINE PARAMETER REGISTER DATA TEST
;SET BIT 14 IN LINE PARAMETER TO 1
;VERIFY THAT BIT 14 WAS SET
;CLEAR BIT 14
;VERIFY THAT BIT 14 WAS CLEARED

```

```

006102          006102 012767 000340 171666 TS \XN,4000,2$
006110          006110 012767 004000 011740 T53:  MOV  #340,PS           ;DISABLE ALL INTERRUPTS
006116          006116 012767 006170 011726      MOV  #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
                                MOV  #2$,ESCAPE           ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV  #,FREEZ1             ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
006124          006124 012777 004000 011656      MOV  #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
006132          006132 016703 011656      MOV  DHLPR,R3         ;SET UP POINTER TO LINE PARAMETER
006136          006136 012705 010000      MOV  #10000,R5        ;BIT 14 WILL BE SET IN LINE PARAMETER
006142          006142 010513      MOV  R5,(R3)          ;SET BIT 14
006144          006144 011304      MOV  (R3),R4          ;GET CONTENTS OF LINE PARAMETER
                                .IIF NB <>,
                                BIC  #,R4                 ;CLEAR UNWANTED BITS
                                CMP  R5,R4                ;WAS BIT 14 SET
                                BEQ  1$
006146          006146 020504      .IIF IDN <DHLPR>,<DHLPR>, HLT  2           ;LINE PARAMETER REGI
006150          006150 001401      .IIF IDN EMT 2
006152          006152 104002      .IIF IDN <DHLPR>,<DHBCR>, HLT  3           ;LINE PARAMETER REGI
                                .IIF IDN <DHLPR>,<DHSSR>, HLT  4           ;LINE PARAMETER REGI
006154          006154 040513      1$:  BIC  R5,(R3)      ;CLEAR BIT 14
006156          006156 011304      MOV  (R3),R4          ;READ CONTENTS OF LINE PARAMETER
                                .IIF NB <>,
                                BIC  #,R4                 ;CLEAR UNWANTED BITS
                                TST  R4                    ;WAS BIT 14 CLEARED
                                BEQ  2$
                                CLR  R5
006160          006160 005704      .IIF IDN <DHLPR>,<DHLPR>, HLT  2           ;LINE PARAMETER REGI
006162          006162 001402      .IIF IDN EMT 2
006164          006164 005005      .IIF IDN <DHLPR>,<DHBCR>, HLT  3           ;LINE PARAMETER REGI
006166          006166 104002      .IIF IDN <DHLPR>,<DHSSR>, HLT  4           ;LINE PARAMETER REGI
006170          006170 104400      .IIF IDN <DHLPR>,<DHSSR>, HLT  4           ;LINE PARAMETER REGI
                                2$:  SCOPE
                                .IF EQ BITX-4
                                .NLIST
                                BITX=BITX+BITX
                                BITC=BITC+1
                                .LIST
                                .ENDC
                                BITX=BITX+BITX
                                BITC=BITC+1
                                REGTS6 †/DHLPR/,\BITX,†/LINE PARAMETER/,\BITC
                                ;LINE PARAMETER REGISTER DATA TEST
                                ;SET BIT 15 IN LINE PARAMETER TO 1
                                ;VERIFY THAT BIT 15 WAS SET
                                ;CLEAR BIT 15
                                ;VERIFY THAT BIT 15 WAS CLEARED
006172          006172 020000
006172          006172 012767 000340 171576 TS \XN,4000,2$
006200          006200 012767 004000 011650 T54:  MOV  #340,PS           ;DISABLE ALL INTERRUPTS
006206          006206 012767 006260 011636      MOV  #4000,ICOUNT      ;SET UP FOR 4000 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
006214          006214 012777 004000 011566      MOV  #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
006222          006222 016703 011566      MOV  DHLPR,R3         ;SET UP POINTER TO LINE PARAMETER

```

```

006226 012705 020000          MOV    #20000,R5          ;BIT 15 WILL BE SET IN LINE PARAMETER
006232 010513          MOV    R5,(R3)          ;SET BIT 15
006234 011304          MOV    (R3),R4          ;GET CONTENTS OF LINE PARAMETER
                          .IIF NB <>, BIC    #,R4          ;CLEAR UNWANTED BITS
006236 020504          CMP    R5,R4          ;WAS BIT 15 SET
006240 001401          BEQ    1$
006242          .IIF IDN <DHLPR>,<DHLPR>, HLT    2          ;LINE PARAMETER REGI
STER ERROR
006242 104002          EMT    2
                          .IIF IDN <DHLPR>,<DHBCR>, HLT    3          ;LINE PARAMETER REGI
STER ERROR
                          .IIF IDN <DHLPR>,<DHSSR>, HLT    4          ;LINE PARAMETER REGI
STER ERROR
006244 040513          1$:    BIC    R5,(R3)          ;CLEAR BIT 15
006246 011304          MOV    (R3),R4          ;READ CONTENTS OF LINE PARAMETER
                          .IIF NB <>, BIC    #,R4          ;CLEAR UNWANTED BITS
006250 005704          TST    R4          ;WAS BIT 15 CLEARED
006252 001402          BEQ    2$
006254 005005          CLR    R5
006256          .IIF IDN <DHLPR>,<DHLPR>, HLT    2          ;LINE PARAMETER REGI
STER ERROR
006256 104002          EMT    2
                          .IIF IDN <DHLPR>,<DHBCR>, HLT    3          ;LINE PARAMETER REGI
STER ERROR
                          .IIF IDN <DHLPR>,<DHSSR>, HLT    4          ;LINE PARAMETER REGI
STER ERROR
006260 104400          2$:    SCOPE
                          .IF EQ BITX-4
                          .NLIST
                          BITX=BITX+BITX
                          BITC=BITC+1
                          .LIST
                          .ENDC
                          BITX=BITX+BITX
                          BITC=BITC+1
006262          REGS6 †/DHLPR/,\BITX,†/LINE PARAMETER/,\BITC
                          ;LINE PARAMETER REGISTER DATA TEST
                          ;SET BIT 16 IN LINE PARAMETER TO 1
                          ;VERIFY THAT BIT 16 WAS SET
                          ;CLEAR BIT 16
                          ;VERIFY THAT BIT 16 WAS CLEARED

006262          TS \XN,4000,2$
005262 012767 000340 171506 T55:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
006270 012767 004000 011560      MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
006276 012767 006350 011546      MOV    #2$,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST
                          .IF NB <>, MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA      ; 3
                          .ENDC
                          XN=XN+1
006304 000056          MOV    #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
006312 016703 011476          MOV    DHLPR,R3          ;SET UP POINTER TO LINE PARAMETER
006316 012705 040000          MOV    #40000,R5          ;BIT 16 WILL BE SET IN LINE PARAMETER
006322 010513          MOV    R5,(R3)          ;SET BIT 16
006324 011304          MOV    (R3),R4          ;GET CONTENTS OF LINE PARAMETER
                          .IIF NB <>, BIC    #,R4          ;CLEAR UNWANTED BITS
006326 020504          CMP    R5,R4          ;WAS BIT 16 SET
006330 001401          BEQ    1$
006332          .IIF IDN <DHLPR>,<DHLPR>, HLT    2          ;LINE PARAMETER REGI
STER ERROR
006332 104002          EMT    2
                          .IIF IDN <DHLPR>,<DHBCR>, HLT    3          ;LINE PARAMETER REGI
STER ERROR
                          .IIF IDN <DHLPR>,<DHSSR>, HLT    4          ;LINE PARAMETER REGI
STER ERROR

```

```

006334 040513          1$:   BIC   R5,(R3)          ;CLEAR BIT 16
006336 011304          MOV   (R3),R4          ;READ CONTENTS OF LINE PARAMETER
                                .IIF NB <>, BIC   #,R4          ;CLEAR UNWANTED BITS
                                TST   R4              ;WAS BIT 16 CLEARED
006340 005704          BEQ   2$
006342 001402          CLR   R5
006344 005005          .IIF IDN <DHLPR>,<DHLPR>, HLT   2          ;LINE PARAMETER REGI
006346          EMT   2
STER ERROR          .IIF IDN <DHLPR>,<DHBCR>, HLT   3          ;LINE PARAMETER REGI
006346 104002          .IIF IDN <DHLPR>,<DHSSR>, HLT   4          ;LINE PARAMETER REGI
STER ERROR
STER ERROR
006350 104400          2$:   SCOPE
                                .IF EQ BITX-4
                                .NLIST
                                BITX=BITX+BITX
                                BITC=BITC+1
                                .LIST
                                .ENDC
                                BITX=BITX+BITX
                                BITC=BITC+1
                                REGTS6 †/DHLPR/,\BITX,†/LINE PARAMETER/,\BITC
                                ;LINE PARAMETER REGISTER DATA TEST
                                ;SET BIT 17 IN LINE PARAMETER TO 1
                                ;VERIFY THAT BIT 17 WAS SET
                                ;CLEAR BIT 17
                                ;VERIFY THAT BIT 17 WAS CLEARED
                                100000
                                000017
006352
006352 012767 000340 171416 TS \XN,4000,2$
006352 012767 004000 011470 T56:  MOV   #340,PS          ;DISABLE ALL INTERRUPTS
006360 012767 004000 011470     MOV   #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
006366 012767 006440 011456     MOV   #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV   #,FREEZ1          ;SET UP TO LOOP WITH DATA          : 3
                                .ENDC
                                XN=XN+1
006374 012777 004000 011406     MOV   #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
006402 016703 011406          MOV   DHLPR,R3        ;SET UP POINTER TO LINE PARAMETER
006406 012705 100000          MOV   #100000,R5      ;BIT 17 WILL BE SET IN LINE PARAMETER
006412 010513          MOV   R5,(R3)        ;SET BIT 17
006414 011304          MOV   (R3),R4        ;GET CONTENTS OF LINE PARAMETER
                                .IIF NB <>, BIC   #,R4          ;CLEAR UNWANTED BITS
                                CMP   R5,R4          ;WAS BIT 17 SET
                                BEQ   1$
006416 020504          .IIF IDN <DHLPR>,<DHLPR>, HLT   2          ;LINE PARAMETER REGI
006420 001401          EMT   2
006422          .IIF IDN <DHLPR>,<DHBCR>, HLT   3          ;LINE PARAMETER REGI
STER ERROR          .IIF IDN <DHLPR>,<DHSSR>, HLT   4          ;LINE PARAMETER REGI
006422 104002          1$:   BIC   R5,(R3)          ;CLEAR BIT 17
006424 040513          MOV   (R3),R4          ;READ CONTENTS OF LINE PARAMETER
006426 011304          .IIF NB <>, BIC   #,R4          ;CLEAR UNWANTED BITS
                                TST   R4              ;WAS BIT 17 CLEARED
006430 005704          BEQ   2$
006432 001402          CLR   R5
006434 005005          .IIF IDN <DHLPR>,<DHLPR>, HLT   2          ;LINE PARAMETER REGI
006436          EMT   2
STER ERROR          .IIF IDN <DHLPR>,<DHBCR>, HLT   3          ;LINE PARAMETER REGI
006436 104002          .IIF IDN <DHLPR>,<DHSSR>, HLT   4          ;LINE PARAMETER REGI
STER ERROR
STER ERROR

```



```

006440 104400          2$: SCOPE
                        .IF EQ BITX-4
                        .NLIST
                        BITX=BITX+BITX
                        BITC=BITC+1
                        .LIST
                        .ENDC
                        BITX=BITX+BITX
                        BITC=BITC+1
268          000000    XBIT=BITX
269          000020    CBIT=BITC
270          000001    BITX=1
271          000000    BITC=0
273          000020    .REPT 16.
274          REGS6  +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC
275          .NLIST
276          BITX=BITX+BITX
277          BITC=BITC+1
278          .LIST
279          .ENDR
006442          REGS6  +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC

;BREAK CONTROL REGISTER DATA TEST
;SET BIT 0 IN BREAK CONTROL TO 1
;VERIFY THAT BIT 0 WAS SET
;CLEAR BIT 0
;VERIFY THAT BIT 0 WAS CLEARED

006442          TS \XN,4000,2$
006442 012767 000340 171326  T57:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
006450 012767 004000 011400      MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
006456 012767 006530 011366      MOV    #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST

                        .IF NB <>
                        MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA      ; 3
                        .ENDC
006464 012777 004000 011316  XN=XN+1
006472 016703 011326          MOV    #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
006476 012705 000001          MOV    DHBCR,R3        ;SET UP POINTER TO BREAK CONTROL
006502 010513          MOV    #1,R5           ;BIT 0 WILL BE SET IN BREAK CONTROL
006504 011304          MOV    R5,(R3)        ;SET BIT 0
                        MOV    (R3),R4          ;GET CONTENTS OF BREAK CONTROL
006506 020504          .IIF NB <>, BIC    #,R4    ;CLEAR UNWANTED BITS
006510 001401          CMP    R5,R4          ;WAS BIT 0 SET
                        BEQ    1$

                        .IIF IDN <DHBCR>,<DHLPR>, HLT    2          ;BREAK CONTROL REGIS
TER ERROR 006512          .IIF IDN <DHBCR>,<DHBCR>, HLT    3          ;BREAK CONTROL REGIS
TER ERROR 006512 104003          EMT    3
                        .IIF IDN <DHBCR>,<DHSSR>, HLT    4          ;BREAK CONTROL REGIS
TER ERROR 006514 040513 1$: BIC    R5,(R3)          ;CLEAR BIT 0
006516 011304          MOV    (R3),R4        ;READ CONTENTS OF BREAK CONTROL
                        .IIF NB <>, BIC    #,R4    ;CLEAR UNWANTED BITS
006520 005704          TST    R4            ;WAS BIT 0 CLEARED
006522 001402          BEQ    2$
006524 005005          CLR    R5

                        .IIF IDN <DHBCR>,<DHLPR>, HLT    2          ;BREAK CONTROL REGIS
TER ERROR 006526          .IIF IDN <DHBCR>,<DHBCR>, HLT    3          ;BREAK CONTROL REGIS
TER ERROR 006526 104003          EMT    3

```

```

                .IIF IDN      <DHBCR>,<DHSSR>.,      HLT      4      ;BREAK CONTROL REGIS
TER ERROR
006530 104400      2$:      SCOPE
                BITX=BITX+BITX
                BITC=BITC+1
006532      REGTS6  +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC

                ;BREAK CONTROL REGISTER DATA TEST
                ;SET BIT 1 IN BREAK CONTROL TO 1
                ;VERIFY THAT BIT 1 WAS SET
                ;CLEAR BIT 1
                ;VERIFY THAT BIT 1 WAS CLEARED

006532      TS \XN,4000,2$
006532 012767 000340 171236 T60:  MOV      #340,PS      ;DISABLE ALL INTERRUPTS
006540 012767 004000 011310      MOV      #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
006546 012767 006620 011276      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
006554 012777 004000 011226      MOV      #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
006562 016703 011236      MOV      DHBCR,R3      ;SET UP POINTER TO BREAK CONTROL
006566 012705 000002      MOV      #2,R5      ;BIT 1 WILL BE SET IN BREAK CONTROL
006572 010513      MOV      R5,(R3)      ;SET BIT 1
006574 011304      MOV      (R3),R4      ;GET CONTENTS OF BREAK CONTROL
                .IIF NB      <>.,      BIC      #,R4      ;CLEAR UNWANTED BITS
                CMP      R5,R4      ;WAS BIT 1 SET
                BEQ      1$
006576 020504      .IIF IDN      <DHBCR>,<DHLPR>.,      HLT      2      ;BREAK CONTROL REGIS
006600 001401      .IIF IDN      <DHBCR>,<DHBCR>.,      HLT      3      ;BREAK CONTROL REGIS
TER ERROR
006602      EMT      3
TER ERROR      .IIF IDN      <DHBCR>,<DHSSR>.,      HLT      4      ;BREAK CONTROL REGIS
006602 104003
TER ERROR      1$:      BIC      R5,(R3)      ;CLEAR BIT 1
006604 040513      MOV      (R3),R4      ;READ CONTENTS OF BREAK CONTROL
006606 011304      .IIF NB      <>.,      BIC      #,R4      ;CLEAR UNWANTED BITS
                TST      R4      ;WAS BIT 1 CLEARED
                BEQ      2$
                CLR      R5
006610 005704      .IIF IDN      <DHBCR>,<DHLPR>.,      HLT      2      ;BREAK CONTROL REGIS
006612 001402      .IIF IDN      <DHBCR>,<DHBCR>.,      HLT      3      ;BREAK CONTROL REGIS
006614 005005      EMT      3
TER ERROR      .IIF IDN      <DHBCR>,<DHSSR>.,      HLT      4      ;BREAK CONTROL REGIS
006616 104003
TER ERROR      2$:      SCOPE
006620 104400      BITX=BITX+BITX
                BITC=BITC+1
006622      REGTS6  +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC

                ;BREAK CONTROL REGISTER DATA TEST
                ;SET BIT 2 IN BREAK CONTROL TO 1
                ;VERIFY THAT BIT 2 WAS SET
                ;CLEAR BIT 2
                ;VERIFY THAT BIT 2 WAS CLEARED

006622      TS \XN,4000,2$
006622 012767 000340 171146 T61:  MOV      #340,PS      ;DISABLE ALL INTERRUPTS
006630 012767 004000 011220      MOV      #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
006636 012767 006710 011206      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
006644 000062 004000 011136 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
006652 012777 016703 011146 MOV DHBCR,R3 ;SET UP POINTER TO BREAK CONTROL
006656 012705 000004 MOV #4,R5 ;BIT 2 WILL BE SET IN BREAK CONTROL
006662 010513 MOV R5,(R3) ;SET BIT 2
006664 011304 MOV (R3),R4 ;GET CONTENTS OF BREAK CONTROL
        .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
        CMP R5,R4 ;WAS BIT 2 SET
        BEQ 1$
        .IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
        .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
        EMT 3
        .IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
1$: BIC R5,(R3) ;CLEAR BIT 2
    MOV (R3),R4 ;READ CONTENTS OF BREAK CONTROL
        .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
        TST R4 ;WAS BIT 2 CLEARED
        BEQ 2$
        CLR R5
        .IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
        .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
        EMT 3
        .IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
2$: SCOPE
    BITX=BITX+BITX
    BITC=BITC+1
    REGTS6 +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC

;BREAK CONTROL REGISTER DATA TEST
;SET BIT 3 IN BREAK CONTROL TO 1
;VERIFY THAT BIT 3 WAS SET
;CLEAR BIT 3
;VERIFY THAT BIT 3 WAS CLEARED

006712 000063 004000 011046 TS \XN,4000,2$
006712 012767 000340 171056 T62: MOV #340,PS ;DISABLE ALL INTERRUPTS
006720 012767 004000 011130 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
006726 012767 007000 011116 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
        .IF NB <>
        MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
        .ENDC
        XN=XN+1
006734 000063 004000 011046 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
006742 012777 016703 011056 MOV DHBCR,R3 ;SET UP POINTER TO BREAK CONTROL
006746 012705 000010 MOV #10,R5 ;BIT 3 WILL BE SET IN BREAK CONTROL
006752 010513 MOV R5,(R3) ;SET BIT 3
006754 011304 MOV (R3),R4 ;GET CONTENTS OF BREAK CONTROL
        .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
        CMP R5,R4 ;WAS BIT 3 SET
        BEQ 1$
        .IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
        .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
        EMT 3
        .IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
    
```

TER ERROR

```

006764 040513      1$:   BIC   R5,(R3)           ;CLEAR BIT 3
006766 011304      MOV   (R3),R4           ;READ CONTENTS OF BREAK CONTROL
                                BIC   #,R4           ;CLEAR UNWANTED BITS
006770 005704      TST   R4               ;WAS BIT 3 CLEARED
006772 001402      BEQ   2$
006774 005005      CLR   R5

                                .IIF IDN   <DHBCR>,<DHLPR>,   HLT   2           ;BREAK CONTROL REGIS
                                .IIF IDN   <DHBCR>,<DHBCR>,   HLT   3           ;BREAK CONTROL REGIS
                                EMT       3
                                .IIF IDN   <DHBCR>,<DHSSR>,   HLT   4           ;BREAK CONTROL REGIS

TER ERROR
006776
TER ERROR
006776 104003

TER ERROR
007000 104400
                                2$:   SCOPE
                                BITX=BITX+BITX
                                BITC=BITC+1
                                REGTS6  +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC

                                ;BREAK CONTROL REGISTER DATA TEST
                                ;SET BIT 4 IN BREAK CONTROL TO 1
                                ;VERIFY THAT BIT 4 WAS SET
                                ;CLEAR BIT 4
                                ;VERIFY THAT BIT 4 WAS CLEARED

007002
007002 012767 000340 170766 TS \XN,4000,2$
007010 012767 004000 011040 T63:  MOV   #340,PS           ;DISABLE ALL INTERRUPTS
007016 012767 007070 011026 MOV   #4000,ICOUNT       ;SET UP FOR 4000 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

007024 012777 004000 010756 MOV   #BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
007032 016703 010766 MOV   DHBCR,R3         ;SET UP POINTER TO BREAK CONTROL
007036 012705 00J020 MOV   #20,R5           ;BIT 4 WILL BE SET IN BREAK CONTROL
007042 010513 MOV   R5,(R3)         ;SET BIT 4
007044 011304 MOV   (R3),R4         ;GET CONTENTS OF BREAK CONTROL
                                .IIF NB   <> ,         BIC   #,R4           ;CLEAR UNWANTED BITS
                                CMP   R5,R4           ;WAS BIT 4 SET
                                BEQ   1$

                                .IIF IDN   <DHBCR>,<DHLPR>,   HLT   2           ;BREAK CONTROL REGIS
                                .IIF IDN   <DHBCR>,<DHBCR>,   HLT   3           ;BREAK CONTROL REGIS
                                EMT       3
                                .IIF IDN   <DHBCR>,<DHSSR>,   HLT   4           ;BREAK CONTROL REGIS

TER ERROR
007052
TER ERROR
007052 104003

TER ERROR
007054 040513      1$:   BIC   R5,(R3)           ;CLEAR BIT 4
007056 011304      MOV   (R3),R4           ;READ CONTENTS OF BREAK CONTROL
                                .IIF NB   <> ,         BIC   #,R4           ;CLEAR UNWANTED BITS
                                TST   R4               ;WAS BIT 4 CLEARED
                                BEQ   2$
                                CLR   R5

                                .IIF IDN   <DHBCR>,<DHLPR>,   HLT   2           ;BREAK CONTROL REGIS
                                .IIF IDN   <DHBCR>,<DHBCR>,   HLT   3           ;BREAK CONTROL REGIS
                                EMT       3
                                .IIF IDN   <DHBCR>,<DHSSR>,   HLT   4           ;BREAK CONTROL REGIS

TER ERROR
007066
TER ERROR
007066 104003

TER ERROR
007070 104400
                                2$:   SCOPE
                                BITX=BITX+BITX
                                BITC=BITC+1
                                REGTS6  +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC

                                ;BREAK CONTROL REGISTER DATA TEST
007072

```

;BREAK CONTROL REGISTER DATA TEST

```

;SET BIT 5 IN BREAK CONTROL TO 1
;VERIFY THAT BIT 5 WAS SET
;CLEAR BIT 5
;VERIFY THAT BIT 5 WAS CLEARED

```

```

007072      000065      TS \XN,4000,2$
007072 012767 000340 170676 T64:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
007100 012767 004000 010750      MOV    #4000,ICOUNT  ;SET UP FOR 4000 ITERATIONS
007106 012767 007160 010736      MOV    #2$,ESCAPE   ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
007114      000065      004000 010666      MOV    #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
007122 016703 010676      MOV    DHBCR,R3      ;SET UP POINTER TO BREAK CONTROL
007126 012705 000040      MOV    #40,R5        ;BIT 5 WILL BE SET IN BREAK CONTROL
007132 010513      MOV    R5,(R3)       ;SET BIT 5
007134 011304      MOV    (R3),R4       ;GET CONTENTS OF BREAK CONTROL
                                .IIF NB  <>,
                                BIC    #,R4          ;CLEAR UNWANTED BITS
007136 020504      CMP    R5,R4         ;WAS BIT 5 SET
007140 001401      BEQ    1$
                                .IIF IDN  <DHBCR>,<DHLPR>,
                                HLT    2              ;BREAK CONTROL REGIS
TER ERROR
007142      .IIF IDN  <DHBCR>,<DHBCR>,
                                HLT    3              ;BREAK CONTROL REGIS
TER ERROR
007142 104003      EMT    3
                                .IIF IDN  <DHBCR>,<DHSSR>,
                                HLT    4              ;BREAK CONTROL REGIS
TER ERROR
007144 040513      1$:   BIC    R5,(R3)    ;CLEAR BIT 5
007146 011304      MOV    (R3),R4       ;READ CONTENTS OF BREAK CONTROL
                                .IIF NB  <>,
                                BIC    #,R4         ;CLEAR UNWANTED BITS
007150 005704      TST    R4            ;WAS BIT 5 CLEARED
007152 001402      BEQ    2$
007154 005005      CLR    R5
                                .IIF IDN  <DHBCR>,<DHLPR>,
                                HLT    2              ;BREAK CONTROL REGIS
TER ERROR
007156      .IIF IDN  <DHBCR>,<DHBCR>,
                                HLT    3              ;BREAK CONTROL REGIS
TER ERROR
007156 104003      EMT    3
                                .IIF IDN  <DHBCR>,<DHSSR>,
                                HLT    4              ;BREAK CONTROL REGIS
TER ERROR
007160 104400      2$:   SCOPE
000100      BITX=BITX+BITX
000006      BITC=BITC+1
007162      REGTS6  +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC

```

```

;BREAK CONTROL REGISTER DATA TEST
;SET BIT 6 IN BREAK CONTROL TO 1
;VERIFY THAT BIT 6 WAS SET
;CLEAR BIT 6
;VERIFY THAT BIT 6 WAS CLEARED

```

```

007162      000066      TS \XN,4000,2$
007162 012767 000340 170606 T65:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
007170 012767 004000 010660      MOV    #4000,ICOUNT  ;SET UP FOR 4000 ITERATIONS
007176 012767 007250 010646      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
007204      000066      004000 010576      MOV    #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
007212 016703 010606      MOV    DHBCR,R3      ;SET UP POINTER TO BREAK CONTROL
007216 012705 000100      MOV    #100,R5       ;BIT 6 WILL BE SET IN BREAK CONTROL

```

```

007222 010513      MOV      R5,(R3)      ;SET BIT 6
007224 011304      MOV      (R3),R4      ;GET CONTENTS OF BREAK CONTROL
                                BIC      #,R4      ;CLEAR UNWANTED BITS
007226 020504      CMP      R5,R4      ;WAS BIT 6 SET
007230 001401      BEQ      1$
                                .IIF IDN <DHBCR>,<DHLPR>, HLT      2      ;BREAK CONTROL REGIS
TER ERROR 007232      .IIF IDN <DHBCR>,<DHBCR>, HLT      3      ;BREAK CONTROL REGIS
007232 104003      EMT      3
                                .IIF IDN <DHBCR>,<DHSSR>, HLT      4      ;BREAK CONTROL REGIS
TER ERROR 007234 040513      1$:      BIC      R5,(R3)      ;CLEAR BIT 6
007236 011304      MOV      (R3),R4      ;READ CONTENTS OF BREAK CONTROL
                                .IIF NB <>, BIC      #,R4      ;CLEAR UNWANTED BITS
007240 005704      TST      R4      ;WAS BIT 6 CLEARED
007242 001402      BEQ      2$
007244 005005      CLR      R5
                                .IIF IDN <DHBCR>,<DHLPR>, HLT      2      ;BREAK CONTROL REGIS
TER ERROR 007246      .IIF IDN <DHBCR>,<DHBCR>, HLT      3      ;BREAK CONTROL REGIS
TER ERROR 007246 104003      EMT      3
                                .IIF IDN <DHBCR>,<DHSSR>, HLT      4      ;BREAK CONTROL REGIS
TER ERROR 007250 104400      2$:      SCOPE
007252 000200      BITX=BITX+BITX
007252 000007      BITC=BITC+1
                                REGTS6 +/DHBCR/,\BITX,+/BREAK CONTROL/, \BITC
                                ;BREAK CONTROL REGISTER DATA TEST
                                ;SET BIT 7 IN BREAK CONTROL TO 1
                                ;VERIFY THAT BIT 7 WAS SET
                                ;CLEAR BIT 7
                                ;VERIFY THAT BIT 7 WAS CLEARED
007252 000067      TS \XN,4000,2$
007252 012767 000340 170516 T66:      MOV      #340,PS      ;DISABLE ALL INTERRUPTS
007260 012767 004000 010570      MOV      #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
007266 012767 007340 010556      MOV      #2$,ESCAPE  ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      : 3
                                .ENDC
                                XN=XN+1
007274 000067      MOV      #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
007302 012777 004000 010506      MOV      DHBCR,R3      ;SET UP POINTER TO BREAK CONTROL
007306 016703 010516      MOV      #200,R5      ;BIT 7 WILL BE SET IN BREAK CONTROL
007312 012705 000200      MOV      R5,(R3)      ;SET BIT 7
007314 010513      MOV      (R3),R4      ;GET CONTENTS OF BREAK CONTROL
                                .IIF NB <>, BIC      #,R4      ;CLEAR UNWANTED BITS
007316 020504      CMP      R5,R4      ;WAS BIT 7 SET
007320 001401      BEQ      1$
                                .IIF IDN <DHBCR>,<DHLPR>, HLT      2      ;BREAK CONTROL REGIS
TER ERROR 007322      .IIF IDN <DHBCR>,<DHBCR>, HLT      3      ;BREAK CONTROL REGIS
007322 104003      EMT      3
                                .IIF IDN <DHBCR>,<DHSSR>, HLT      4      ;BREAK CONTROL REGIS
TER ERROR 007324 040513      1$:      BIC      R5,(R3)      ;CLEAR BIT 7
007326 011304      MOV      (R3),R4      ;READ CONTENTS OF BREAK CONTROL
                                .IIF NB <>, BIC      #,R4      ;CLEAR UNWANTED BITS
007330 005704      TST      R4      ;WAS BIT 7 CLEARED
007332 001402      BEQ      2$
007334 005005      CLR      R5
                                .IIF IDN <DHBCR>,<DHLPR>, HLT      2      ;BREAK CONTROL REGIS

```

```

007336          .IIF IDN      <DHBCR>,<DHBCR>,      HLT      3          ;BREAK CONTROL REGIS
TER ERROR
007336 104003          EMT      3
007340 104400          .IIF IDN      <DHBCR>,<DHSSR>,      HLT      4          ;BREAK CONTROL REGIS
000400
000010          2$:      SCOPE
007342          BITX=BITX+BITX
          BITC=BITC+1
          REGTS6 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC

          ;BREAK CONTROL REGISTER DATA TEST
          ;SET BIT 10 IN BREAK CONTROL TO 1
          ;VERIFY THAT BIT 10 WAS SET
          ;CLEAR BIT 10
          ;VERIFY THAT BIT 10 WAS CLEARED

007342          TS \XN,4000,2$
007342 012767 000340 170426 T67:      MOV      #340,PS          ;DISABLE ALL INTERRUPTS
007350 012767 004000 010500          MOV      #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
007356 012767 007430 010466          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

007364 012777 004000 010416          MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
007372 016703 010426          MOV      DHBCR,R3          ;SET UP POINTER TO BREAK CONTROL
007376 012705 000400          MOV      #400,R5          ;BIT 10 WILL BE SET IN BREAK CONTROL
007402 010513          MOV      R5,(R3)          ;SET BIT 10
007404 011304          MOV      (R3),R4          ;GET CONTENTS OF BREAK CONTROL
          .IIF NB      <>,          BIC      #,R4          ;CLEAR UNWANTED BITS
007406 020504          CMP      R5,R4          ;WAS BIT 10 SET
007410 001401          BEQ      1$

          .IIF IDN      <DHBCR>,<DHLPR>,      HLT      2          ;BREAK CONTROL REGIS
TER ERROR
007412          .IIF IDN      <DHBCR>,<DHBCR>,      HLT      3          ;BREAK CONTROL REGIS
TER ERROR
007412 104003          EMT      3
          .IIF IDN      <DHBCR>,<DHSSR>,      HLT      4          ;BREAK CONTROL REGIS
TER ERROR
007414 040513          1$:      BIC      R5,(R3)          ;CLEAR BIT 10
007416 011304          MOV      (R3),R4          ;READ CONTENTS OF BREAK CONTROL
          .IIF NB      <>,          BIC      #,R4          ;CLEAR UNWANTED BITS
007420 005704          TST      R4          ;WAS BIT 10 CLEARED
007422 001402          BEQ      2$
007424 005005          CLR      R5

          .IIF IDN      <DHBCR>,<DHLPR>,      HLT      2          ;BREAK CONTROL REGIS
TER ERROR
007426          .IIF IDN      <DHBCR>,<DHBCR>,      HLT      3          ;BREAK CONTROL REGIS
TER ERROR
007426 104003          EMT      3
          .IIF IDN      <DHBCR>,<DHSSR>,      HLT      4          ;BREAK CONTROL REGIS
TER ERROR
007430 104400          2$:      SCOPE
001000          BITX=BITX+BITX
000011          BITC=BITC+1
007432          REGTS6 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC

          ;BREAK CONTROL REGISTER DATA TEST
          ;SET BIT 11 IN BREAK CONTROL TO 1
          ;VERIFY THAT BIT 11 WAS SET
          ;CLEAR BIT 11
          ;VERIFY THAT BIT 11 WAS CLEARED

007432          TS \XN,4000,2$
007432 012767 000340 170336 T70:      MOV      #340,PS          ;DISABLE ALL INTERRUPTS

```

```

007440 012767 004000 010410      MOV      #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
007446 012767 007520 010376      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB    <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
007454 012777 004000 010326      MOV      #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
007462 016703 010336              MOV      DHBCR,R3        ;SET UP POINTER TO BREAK CONTROL
007466 012705 001000              MOV      #1000,R5        ;BIT 11 WILL BE SET IN BREAK CONTROL
007472 010513              MOV      R5,(R3)        ;SET BIT 11
007474 011304              MOV      (R3),R4        ;GET CONTENTS OF BREAK CONTROL
                                .IIF NB    <>,      BIC      #,R4          ;CLEAR UNWANTED BITS
                                CMP      R5,R4        ;WAS BIT 11 SET
                                BEQ      1$
007476 020504              .IIF IDN  <DHBCR>,<DHLPR>,  HLT      2          ;BREAK CONTROL REGIS
007500 001401              .IIF IDN  <DHBCR>,<DHBCR>,  HLT      3          ;BREAK CONTROL REGIS
TER ERROR 007502
TER ERROR 007502 104003          EMT      3
                                .IIF IDN  <DHBCR>,<DHSSR>,  HLT      4          ;BREAK CONTROL REGIS
TER ERROR 007504 040513          1$:      BIC      R5,(R3)    ;CLEAR BIT 11
007506 011304              MOV      (R3),R4        ;READ CONTENTS OF BREAK CONTROL
                                .IIF NB    <>,      BIC      #,R4          ;CLEAR UNWANTED BITS
                                TST      R4           ;WAS BIT 11 CLEARED
                                BEQ      2$
                                CLR      R5
007510 005704              .IIF IDN  <DHBCR>,<DHLPR>,  HLT      2          ;BREAK CONTROL REGIS
007512 001402              .IIF IDN  <DHBCR>,<DHBCR>,  HLT      3          ;BREAK CONTROL REGIS
007514 005005              EMT      3
                                .IIF IDN  <DHBCR>,<DHSSR>,  HLT      4          ;BREAK CONTROL REGIS
TER ERROR 007516
TER ERROR 007516 104003          EMT      3
TER ERROR 007520 104400          2$:      SCOPE
007522 002000          BITX=BITX+BITX
007522 000012          BITC=BITC+1
                                REGTS6  †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC
                                ;BREAK CONTROL REGISTER DATA TEST
                                ;SET BIT 12 IN BREAK CONTROL TO 1
                                ;VERIFY THAT BIT 12 WAS SET
                                ;CLEAR BIT 12
                                ;VERIFY THAT BIT 12 WAS CLEARED
007522 012767 000340 170246      TS \XN,4000,2$
007522 012767 004000 010320      T71:    MOV      #340,PS    ;DISABLE ALL INTERRUPTS
007530 012767 004000 010320      MOV      #4000,ICOUNT    ;SET UP FOR 4000 ITERATIONS
007536 012767 007610 010306      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB    <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
007544 012777 004000 010236      MOV      #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
007552 016703 010246              MOV      DHBCR,R3        ;SET UP POINTER TO BREAK CONTROL
007556 012705 002000              MOV      #2000,R5        ;BIT 12 WILL BE SET IN BREAK CONTROL
007562 010513              MOV      R5,(R3)        ;SET BIT 12
007564 011304              MOV      (R3),R4        ;GET CONTENTS OF BREAK CONTROL
                                .IIF NB    <>,      BIC      #,R4          ;CLEAR UNWANTED BITS
                                CMP      R5,R4        ;WAS BIT 12 SET
                                BEQ      1$
007566 020504              .IIF IDN  <DHBCR>,<DHLPR>,  HLT      2          ;BREAK CONTROL REGIS
007570 001401              .IIF IDN  <DHBCR>,<DHBCR>,  HLT      3          ;BREAK CONTROL REGIS
TER ERROR 007572
TER ERROR

```



```

007572 104003
TER ERROR
007574 040513
007576 011304
007600 005704
007602 001402
007604 005005
TER ERROR
007606
TER ERROR
007606 104003
TER ERROR
007610 104400
004000
000013
007612

                EMT      3
                .IIF IDN  <DHBCR>,<DHSSR>,      HLT      4      ;BREAK CONTROL REGIS
1$:      BIC      R5,(R3)      ;CLEAR BIT 12
                MOV      (R3),R4      ;READ CONTENTS OF BREAK CONTROL
                .IIF NB <>,      BIC      #,R4      ;CLEAR UNWANTED BITS
                TST      R4      ;WAS BIT 12 CLEARED
                BEQ      2$,
                CLR      R5
                .IIF IDN  <DHBCR>,<DHLPR>,      HLT      2      ;BREAK CONTROL REGIS
                .IIF IDN  <DHBCR>,<DHBCR>,      HLT      3      ;BREAK CONTROL REGIS
                EMT      3
                .IIF IDN  <DHBCR>,<DHSSR>,      HLT      4      ;BREAK CONTROL REGIS
2$:      SCOPE
                BITX=BITX+BITX
                BITC=BITC+1
                REGTS6 ↑/DHBCR/,\BITX,↑/BREAK CONTROL/, \BITC

                ;BREAK CONTROL REGISTER DATA TEST
                ;SET BIT 13 IN BREAK CONTROL TO 1
                ;VERIFY THAT BIT 13 WAS SET
                ;CLEAR BIT 13
                ;VERIFY THAT BIT 13 WAS CLEARED

007612
007612 012767 000340 170156 TS \XN,4000,2$
007620 012767 004000 010230 T72:      MOV      #340,PS      ;DISABLE ALL INTERRUPTS
007626 012767 007700 010216      MOV      #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
                MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                .IF NB <>
                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                .ENDC
                XN=XN+1
007634 012777 004000 010146      MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
007642 016703 010156      MOV      DHBCR,R3      ;SET UP POINTER TO BREAK CONTROL
007646 012705 004000      MOV      #4000,R5      ;BIT 13 WILL BE SET IN BREAK CONTROL
007652 010513      MOV      R5,(R3)      ;SET BIT 13
007654 011304      MOV      (R3),R4      ;GET CONTENTS OF BREAK CONTROL
                .IIF NB <>,      BIC      #,R4      ;CLEAR UNWANTED BITS
                CMP      R5,R4      ;WAS BIT 13 SET
                BEQ      1$,
                .IIF IDN  <DHBCR>,<DHLPR>,      HLT      2      ;BREAK CONTROL REGIS
                .IIF IDN  <DHBCR>,<DHBCR>,      HLT      3      ;BREAK CONTROL REGIS
                EMT      3
                .IIF IDN  <DHBCR>,<DHSSR>,      HLT      4      ;BREAK CONTROL REGIS
1$:      BIC      R5,(R3)      ;CLEAR BIT 13
                MOV      (R3),R4      ;READ CONTENTS OF BREAK CONTROL
                .IIF NB <>,      BIC      #,R4      ;CLEAR UNWANTED BITS
                TST      R4      ;WAS BIT 13 CLEARED
                BEQ      2$,
                CLR      R5
                .IIF IDN  <DHBCR>,<DHLPR>,      HLT      2      ;BREAK CONTROL REGIS
                .IIF IDN  <DHBCR>,<DHBCR>,      HLT      3      ;BREAK CONTROL REGIS
                EMT      3
                .IIF IDN  <DHBCR>,<DHSSR>,      HLT      4      ;BREAK CONTROL REGIS
2$:      SCOPE
                BITX=BITX+BITX
                BITC=BITC+1

```

```

;BREAK CONTROL REGISTER DATA TEST
;SET BIT 14 IN BREAK CONTROL TO 1
;VERIFY THAT BIT 14 WAS SET
;CLEAR BIT 14
;VERIFY THAT BIT 14 WAS CLEARED

```

```

007702      007702 012767 000340 170066 TS \XN,4000,2#
007710      007710 012767 004000 010140 T73:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
007716      007716 012767 007770 010126      MOV    #4000,ICOUNT      ;SET UP FOR 4000 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
007724      000074 007724 012777 004000 010056      MOV    #BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
007732      016703 007732 016703 010066      MOV    DHBCR,R3         ;SET UP POINTER TO BREAK CONTROL
007736      012705 007736 012705 010000      MOV    #10000,R5        ;BIT 14 WILL BE SET IN BREAK CONTROL
007742      010513 007742 010513 010000      MOV    R5,(R3)          ;SET BIT 14
007744      011304 007744 011304 010000      MOV    (R3),R4          ;GET CONTENTS OF BREAK CONTROL
                                .IIF NB  <>, BIC    #,R4         ;CLEAR UNWANTED BITS
007746      020504 007746 020504 010000      CMP    R5,R4            ;WAS BIT 14 SET
007750      001401 007750 001401 010000      BEQ    1#
                                .IIF IDN  <DHBCR>,<DHLPR>., HLT    2           ;BREAK CONTROL REGIS
TER ERROR 007752      .IIF IDN  <DHBCR>,<DHBCR>., HLT    3           ;BREAK CONTROL REGIS
TER ERROR 007752 104003      .IIF IDN  EMT    3           ;BREAK CONTROL REGIS
                                .IIF IDN  <DHBCR>,<DHSSR>., HLT    4           ;BREAK CONTROL REGIS
TER ERROR 007754 040513      1#:    BIC    R5,(R3)       ;CLEAR BIT 14
007756      011304 007756 011304 010000      MOV    (R3),R4         ;READ CONTENTS OF BREAK CONTROL
                                .IIF NB  <>, BIC    #,R4         ;CLEAR UNWANTED BITS
007760      005704 007760 005704 010000      TST    R4              ;WAS BIT 14 CLEARED
007762      001402 007762 001402 010000      BEQ    2#
007764      005005 007764 005005 010000      CLR    R5
                                .IIF IDN  <DHBCR>,<DHLPR>., HLT    2           ;BREAK CONTROL REGIS
TER ERROR 007766      .IIF IDN  <DHBCR>,<DHBCR>., HLT    3           ;BREAK CONTROL REGIS
TER ERROR 007766 104003      .IIF IDN  EMT    3           ;BREAK CONTROL REGIS
                                .IIF IDN  <DHBCR>,<DHSSR>., HLT    4           ;BREAK CONTROL REGIS
TER ERROR 007770 104400      2#:    SCOPE
007772      020000      BITX=BITX+BITX
                                BITC=BITC+1
                                REGTS6 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC

```

```

;BREAK CONTROL REGISTER DATA TEST
;SET BIT 15 IN BREAK CONTROL TO 1
;VERIFY THAT BIT 15 WAS SET
;CLEAR BIT 15
;VERIFY THAT BIT 15 WAS CLEARED

```

```

007772      007772 012767 000340 167776 TS \XN,4000,2#
010000      012767 000340 010000 010050 T74:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
010006      012767 000340 010006 010036      MOV    #4000,ICOUNT      ;SET UP FOR 4000 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
010014      000075 010014 012777 004000 007766      MOV    #BIT11,@DHSCR     ;MASTER CLEAR INTERFACE

```

```

010022 016703 007776          MOV    DHBCR,R3          ;SET UP POINTER TO BREAK CONTROL
010026 012775 020000          MOV    #20000,R5        ;BIT 15 WILL BE SET IN BREAK CONTROL
010032 010513                    MOV    R5,(R3)          ;SET BIT 15
010034 011304                    MOV    (R3),R4          ;GET CONTENTS OF BREAK CONTROL
                                BIC    #,R4              ;CLEAR UNWANTED BITS
010036 020504                    CMP    R5,R4            ;WAS BIT 15 SET
010040 001401                    BEQ    1$

                                .IIF IDN    <DHBCR>,<DHLPR>,    HLT    2          ;BREAK CONTROL REGIS
                                .IIF IDN    <DHBCR>,<DHBCR>,    HLT    3          ;BREAK CONTROL REGIS
                                .IIF IDN    EMT    3
                                .IIF IDN    <DHBCR>,<DHSSR>,    HLT    4          ;BREAK CONTROL REGIS
010042 010042 104003          1$:    BIC    R5,(R3)          ;CLEAR BIT 15
                                MOV    (R3),R4          ;READ CONTENTS OF BREAK CONTROL
                                .IIF NB <>,    BIC    #,R4          ;CLEAR UNWANTED BITS
                                TST    R4              ;WAS BIT 15 CLEARED
                                BEQ    2$
                                CLR    R5
                                .IIF IDN    <DHBCR>,<DHLPR>,    HLT    2          ;BREAK CONTROL REGIS
                                .IIF IDN    <DHBCR>,<DHBCR>,    HLT    3          ;BREAK CONTROL REGIS
010044 040513                    .IIF IDN    EMT    3
010046 011304                    .IIF IDN    <DHBCR>,<DHSSR>,    HLT    4          ;BREAK CONTROL REGIS
010050 005704                    2$:    SCOPE
010052 001402                    BITX=BITX+BITX
010054 005005                    BITC=BITC+1
                                REGTS6  +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC

                                ;BREAK CONTROL REGISTER DATA TEST
                                ;SET BIT 16 IN BREAK CONTROL TO 1
                                ;VERIFY THAT BIT 16 WAS SET
                                ;CLEAR BIT 16
                                ;VERIFY THAT BIT 16 WAS CLEARED

010062                                TS \XN,4000,2$
010062 012767 000340 167706    T75:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
010070 012767 004000 007760    MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
010076 012767 010150 007746    MOV    #2$,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB    <>
                                MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
010104 000076                    MOV    #BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
010112 016703 007706                    MOV    DHBCR,R3          ;SET UP POINTER TO BREAK CONTROL
010116 012705 040000                    MOV    #40000,R5        ;BIT 16 WILL BE SET IN BREAK CONTROL
010122 010513                    MOV    R5,(R3)          ;SET BIT 16
010124 011304                    MOV    (R3),R4          ;GET CONTENTS OF BREAK CONTROL
                                .IIF NB <>,    BIC    #,R4          ;CLEAR UNWANTED BITS
                                CMP    R5,R4            ;WAS BIT 16 SET
                                BEQ    1$
                                .IIF IDN    <DHBCR>,<DHLPR>,    HLT    2          ;BREAK CONTROL REGIS
                                .IIF IDN    <DHBCR>,<DHBCR>,    HLT    3          ;BREAK CONTROL REGIS
010126 020504                    .IIF IDN    EMT    3
010130 001401                    .IIF IDN    <DHBCR>,<DHSSR>,    HLT    4          ;BREAK CONTROL REGIS
010132 010132 104003          1$:    BIC    R5,(R3)          ;CLEAR BIT 16
                                MOV    (R3),R4          ;READ CONTENTS OF BREAK CONTROL
                                .IIF NB <>,    BIC    #,R4          ;CLEAR UNWANTED BITS
                                TST    R4              ;WAS BIT 16 CLEARED
                                BEQ    2$
010140 005704                    2$:
010142 001402                    ;

```

```

010144 005005          CLR      R5
TER ERROR             .IIF IDN  <DHBCR>,<DHLPR>, HLT  2          ;BREAK CONTROL REGIS
010146               .IIF IDN  <DHBCR>,<DHBCR>, HLT  3          ;BREAK CONTROL REGIS
TER ERROR             010146 104003
010146               EMT      3
TER ERROR             .IIF IDN  <DHBCR>,<DHSSR>, HLT  4          ;BREAK CONTROL REGIS
010150 104400        2$:      SCOPE
010150 100000        BITX=BITX+BITX
010150 000017        BITC=BITC+1
010152               REGTS6  +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC

;BREAK CONTROL REGISTER DATA TEST
;SET BIT 17 IN BREAK CONTROL TO 1
;VERIFY THAT BIT 17 WAS SET
;CLEAR BIT 17
;VERIFY THAT BIT 17 WAS CLEARED

010152               TS \XN,4000,2$
010152 012767 000340 167616 T76:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
010160 012767 004000 007670      MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
010166 012767 010240 007656      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                          .IF NB  <>
                          MOV    @,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                          .ENDC
010174 000077        XN=XN+1
010174 012777 004000 007606      MOV    @BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
010202 016703 007616          MOV    DHBCR,R3        ;SET UP POINTER TO BREAK CONTROL
010206 012705 100000          MOV    #100000,R5     ;BIT 17 WILL BE SET IN BREAK CONTROL
010212 010513          MOV    R5,(R3)        ;SET BIT 17
010214 011304          MOV    (R3),R4        ;GET CONTENTS OF BREAK CONTROL
010216 020504          .IIF NB  <>, BIC    @,R4          ;CLEAR UNWANTED BITS
010220 001401          CMP    R5,R4          ;WAS BIT 17 SET
                          BEQ    1$
TER ERROR             .IIF IDN  <DHBCR>,<DHLPR>, HLT  2          ;BREAK CONTROL REGIS
010222               .IIF IDN  <DHBCR>,<DHBCR>, HLT  3          ;BREAK CONTROL REGIS
TER ERROR             010222 104003
010222               EMT      3
TER ERROR             .IIF IDN  <DHBCR>,<DHSSR>, HLT  4          ;BREAK CONTROL REGIS
010224 040513        1$:      BIC    R5,(R3)        ;CLEAR BIT 17
010226 011304          MOV    (R3),R4        ;READ CONTENTS OF BREAK CONTROL
010230 005704          .IIF NB  <>, BIC    @,R4          ;CLEAR UNWANTED BITS
010232 001402          TST    R4          ;WAS BIT 17 CLEARED
010234 005005          BEQ    2$
010234 005005          CLR    R5
TER ERROR             .IIF IDN  <DHBCR>,<DHLPR>, HLT  2          ;BREAK CONTROL REGIS
010236               .IIF IDN  <DHBCR>,<DHBCR>, HLT  3          ;BREAK CONTROL REGIS
TER ERROR             010236 104003
010236               EMT      3
TER ERROR             .IIF IDN  <DHBCR>,<DHSSR>, HLT  4          ;BREAK CONTROL REGIS
010240 104400        2$:      SCOPE
010240 000000        BITX=BITX+BITX
010240 000020        BITC=BITC+1
281 000000          XBIT=BITX
282 000020          CBIT=BITC
283 000001          BITX=1
284 000000          BITC=0
286 000007          .REPT  7
287               REGTS6  +/DHSSR/,\BITX,+/SIL0 STATUS/,\BITC,700
288               .IF EQ  BITX-40
289               .NLIST

```

290  
291  
292  
293  
294  
295  
296  
297  
298  
299

```

BITX=100000
BITC=17
.LIST
.IFF
.NLIST
BITX=BITX+BITX
BITC=BITC+1
.LIST
.ENDC
.ENDR
REGTS6 †/DHSSR/,\BITX,†/SILO STATUS/,\BITC,700
    
```

```

;SILO STATUS REGISTER DATA TEST
;SET BIT 0 IN SILO STATUS TO 1
;VERIFY THAT BIT 0 WAS SET
;CLEAR BIT 0
;VERIFY THAT BIT 0 WAS CLEARED
    
```

010242

010242 012767 000340 167526  
010250 012767 004000 007600  
010256 012767 010340 007566

TS \XN,4000,2‡

```

T77: MOV #340,PS
      MOV #4000,ICOUNT
      MOV #2‡,ESCAPE
      .IF NB <>
      MOV #,FREEZ1
      .ENDC
      XN=XN+1
    
```

```

;DISABLE ALL INTERRUPTS
;SET UP FOR 4000 ITERATIONS
;SET UP TO ESCAPE TO NEXT TEST
;SET UP TO LOOP WITH DATA ; 3
    
```

010264 012777 004000 007516  
010272 016703 007530  
010276 012705 000001  
010302 010513  
010304 011304  
010306 042704 000700  
010312 020504  
010314 001401

```

MOV #BIT11,@DHSCR
MOV DHSSR,R3
MOV #1,R5
MOV R5,(R3)
MOV (R3),R4
.IIF NB <700>, BIC #700,R4
      CMP R5,R4
      BEQ 1‡
      .IIF IDN <DHSSR>,<DHLPR>,
    
```

```

;MASTER CLEAR INTERFACE
;SET UP POINTER TO SILO STATUS
;BIT 0 WILL BE SET IN SILO STATUS
;SET BIT 0
;GET CONTENTS OF SILO STATUS
;CLEAR UNWANTED BITS
;WAS BIT 0 SET
    
```

R ERROR

HLT 2 ;SILO STATUS REGISTE

R ERROR

.IIF IDN <DHSSR>,<DHBCR>,  
HLT 3 ;SILO STATUS REGISTE

R ERROR

010316

.IIF IDN <DHSSR>,<DHSSR>,  
HLT 4 ;SILO STATUS REGISTE

010316 104004  
010320 040513  
010322 011304  
010324 042704 000700  
010330 005704  
010332 001402  
010334 005005

```

EMT 4
1‡: BIC R5,(R3)
      MOV (R3),R4
      .IIF NB <700>, BIC #700,R4
      TST R4
      BEQ 2‡
      CLR R5
      .IIF IDN <DHSSR>,<DHLPR>,
    
```

```

;CLEAR BIT 0
;READ CONTENTS OF SILO STATUS
;CLEAR UNWANTED BITS
;WAS BIT 0 CLEARED
    
```

R ERROR

HLT 2 ;SILO STATUS REGISTE

R ERROR

.IIF IDN <DHSSR>,<DHBCR>,  
HLT 3 ;SILO STATUS REGISTE

R ERROR

010336

.IIF IDN <DHSSR>,<DHSSR>,  
HLT 4 ;SILO STATUS REGISTE

010336 104004  
010340 104400

```

EMT 4
2‡: SCOPE
      .IF EQ BITX-40
      .NLIST
      BITX=100000
      BITC=17
      .LIST
      .IFF
      BITX=BITX+BITX
      BITC=BITC+1
    
```

000002  
000001

```

010342 .ENDC
REGTS6 †/DHSSR/,\BITX,†/SILO STATUS/,\BITC,700
;SILO STATUS REGISTER DATA TEST
;SET BIT 1 IN SILO STATUS TO 1
;VERIFY THAT BIT 1 WAS SET
;CLEAR BIT 1
;VERIFY THAT BIT 1 WAS CLEARED

010342 010342 012767 000340 167426 TS \XN,4000,2‡
010350 012767 004000 007500 T100: MOV #340,PS ;DISABLE ALL INTERRUPTS
010356 012767 010440 007466 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
MOV #2‡,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
.IF NB ‹›
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
\N=XN+1
010364 000101 012777 004000 007416 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
010372 016703 007430 MOV DHSSR,R3 ;SET UP POINTER TO SILO STATUS
010376 012705 000002 MOV #2,R5 ;BIT 1 WILL BE SET IN SILO STATUS
010402 010513 MOV R5,(R3) ;SET BIT 1
010404 011304 MOV (R3),R4 ;GET CONTENTS OF SILO STATUS
010406 042704 000700 .IIF NB ‹700›, BIC #700,R4 ;CLEAR UNWANTED BITS
010412 020504 CMP R5,R4 ;WAS BIT 1 SET
010414 001401 BEQ 1‡

R ERROR .IIF IDN ‹DHSSR›,‹DHLPR›, HLT 2 ;SILO STATUS REGISTE
R ERROR .IIF IDN ‹DHSSR›,‹DHBCR›, HLT 3 ;SILO STATUS REGISTE
R ERROR 010416 .IIF IDN ‹DHSSR›,‹DHSSR›, HLT 4 ;SILO STATUS REGISTE

010416 104004 EMT 4
010420 040513 1‡: BIC R5,(R3) ;CLEAR BIT 1
010422 011304 MOV (R3),R4 ;READ CONTENTS OF SILO STATUS
010424 042704 000700 .IIF NB ‹700›, BIC #700,R4 ;CLEAR UNWANTED BITS
010430 005704 TST R4 ;WAS BIT 1 CLEARED
010432 001402 BEQ 2‡
010434 005005 CLR R5

R ERROR .IIF IDN ‹DHSSR›,‹DHLPR›, HLT 2 ;SILO STATUS REGISTE
R ERROR .IIF IDN ‹DHSSR›,‹DHBCR›, HLT 3 ;SILO STATUS REGISTE
R ERROR 010436 .IIF IDN ‹DHSSR›,‹DHSSR›, HLT 4 ;SILO STATUS REGISTE

010436 104004 EMT 4
010440 104400 2‡: SCOPE
.IF EQ BITX-40
.NLIST
BITX=100000
BITC=17
.LIST
.IFF
BITX=BITX+BITX
BITC=BITC+1
.ENDC
010442 REGTS6 †/DHSSR/,\BITX,†/SILO STATUS/,\BITC,700
;SILO STATUS REGISTER DATA TEST
;SET BIT 2 IN SILO STATUS TO 1
;VERIFY THAT BIT 2 WAS SET
;CLEAR BIT 2
;VERIFY THAT BIT 2 WAS CLEARED
000004
000002

```

```

010442      TS \XN,4000,2$
010442 012767 000340 167326 T101:  MOV #340,PS ;DISABLE ALL INTERRUPTS
010450 012767 004000 007400      MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
010456 012767 010540 007366      MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
      .ENDC
      XN=XN+1
010464 000102
010472 012777 004000 007316      MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
010476 016703 007330      MOV DHSSR,R3 ;SET UP POINTER TO SILO STATUS
010502 012705 000004      MOV #4,R5 ;BIT 2 WILL BE SET IN SILO STATUS
010504 010513      MOV R5,(R3) ;SET BIT 2
010506 011304      MOV (R3),R4 ;GET CONTENTS OF SILO STATUS
010512 042704 000700      .IIF NB <700>, BIC #700,R4 ;CLEAR UNWANTED BITS
010514 020504      CMP R5,R4 ;WAS BIT 2 SET
010514 001401      BEQ 1$
      .IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
      .IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
      .IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE
R ERROR
R ERROR
R ERROR 010516
010516 104004      EMT 4
010520 040513      1$: BIC R5,(R3) ;CLEAR BIT 2
010522 011304      MOV (R3),R4 ;READ CONTENTS OF SILO STATUS
010524 042704 000700      .IIF NB <700>, BIC #700,R4 ;CLEAR UNWANTED BITS
010530 005704      TST R4 ;WAS BIT 2 CLEARED
010532 001402      BEQ 2$
010534 005005      CLR R5
      .IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
      .IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
      .IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE
R ERROR
R ERROR
R ERROR 010536
010536 104004      EMT 4
010540 104400      2$: SCOPE
      .IF EQ BITX-40
      .NLIST
      BITX=100000
      BITC=17
      .LIST
      .IFF
      BITX=BITX+BITX
      BITC=BITC+1
      .ENDC
010542      REGTS6 †/DHSSR/,\BITX,†/SILO STATUS/,\BITC,700
      ;SILO STATUS REGISTER DATA TEST
      ;SET BIT 3 IN SILO STATUS TO 1
      ;VERIFY THAT BIT 3 WAS SET
      ;CLEAR BIT 3
      ;VERIFY THAT BIT 3 WAS CLEARED
010542      TS \XN,4000,2$
010542 012767 000340 167226 T102:  MOV #340,PS ;DISABLE ALL INTERRUPTS
010550 012767 004000 007300      MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
010556 012767 010640 007266      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
010564 000103
010564 012777 004000 007216      MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE

```

```

010572 016703 007230          MOV    DHSSR,R3          ;SET UP POINTER TO SILO STATUS
010576 012705 000010          MOV    #10,R5          ;BIT 3 WILL BE SET IN SILO STATUS
010602 010513                   MOV    R5,(R3)         ;SET BIT 3
010604 011304                   MOV    (R3),R4         ;GET CONTENTS OF SILO STATUS
010606 042704 000700          .IIF NB <700>, BIC    #700,R4 ;CLEAR UNWANTED BITS
010612 020504                   CMP    R5,R4          ;WAS BIT 3 SET
010614 001401                   BEQ    1$             

R ERROR          .IIF IDN    <DHSSR>,<DHLPR>, HLT    2          ;SILO STATUS REGISTE
                  .IIF IDN    <DHSSR>,<DHBCR>, HLT    3          ;SILO STATUS REGISTE
R ERROR          010616          .IIF IDN    <DHSSR>,<DHSSR>, HLT    4          ;SILO STATUS REGISTE

010616 104004                   EMT    4
010620 040513          1$: BIC    R5,(R3)         ;CLEAR BIT 3
010622 011304                   MOV    (R3),R4         ;READ CONTENTS OF SILO STATUS
010624 042704 000700          .IIF NB <700>, BIC    #700,R4 ;CLEAR UNWANTED BITS
010630 005704                   TST    R4              ;WAS BIT 3 CLEARED
010632 001402                   BEQ    2$
010634 005005                   CLR    R5

R ERROR          .IIF IDN    <DHSSR>,<DHLPR>, HLT    2          ;SILO STATUS REGISTE
                  .IIF IDN    <DHSSR>,<DHBCR>, HLT    3          ;SILO STATUS REGISTE
R ERROR          010636          .IIF IDN    <DHSSR>,<DHSSR>, HLT    4          ;SILO STATUS REGISTE

010636 104004                   EMT    4
010640 104400          2$: SCOPE
                  .IF EQ BITX-40
                  .NLIST
                  BITX=100000
                  BITC=17
                  .LIST
                  .IFF
                  BITX=BITX+BITX
                  BITC=BITC+1
                  .ENDC
                  REGTS6  +/DHSSR/,\BITX,+/SILO STATUS/,\BITC,700

R ERROR          010642          ;SILO STATUS REGISTER DATA TEST
                  ;SET BIT 4 IN SILO STATUS TO 1
                  ;VERIFY THAT BIT 4 WAS SET
                  ;CLEAR BIT 4
                  ;VERIFY THAT BIT 4 WAS CLEARED

010642          TS \XN,4000,2$
010642 012767 000340 167126 T103: MOV    #340,PS          ;DISABLE ALL INTERRUPTS
010650 012767 004000 007200          MOV    #4000,ICOUNT    ;SET UP FOR 4000 ITERATIONS
010656 012767 010740 007166          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

010664 000104          010664 012777 004000 007116          MOV    #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
010672 016703 007130          MOV    DHSSR,R3       ;SET UP POINTER TO SILO STATUS
010676 012705 000020          MOV    #20,R5         ;BIT 4 WILL BE SET IN SILO STATUS
010702 010513                   MOV    R5,(R3)         ;SET BIT 4
010704 011304                   MOV    (R3),R4         ;GET CONTENTS OF SILO STATUS
010706 042704 000700          .IIF NB <700>, BIC    #700,R4 ;CLEAR UNWANTED BITS
010712 020504                   CMP    R5,R4          ;WAS BIT 4 SET
010714 001401                   BEQ    1$

R ERROR          .IIF IDN    <DHSSR>,<DHLPR>, HLT    2          ;SILO STATUS REGISTE
                  .IIF IDN    <DHSSR>,<DHBCR>, HLT    3          ;SILO STATUS REGISTE
R ERROR

```



```

010716          .IIF IDN      <DHSSR>,<DHSSR>,      HLT      4          ;SILO STATUS REGISTE
R ERROR
010716 104004          EMT      4
010720 040513          1$:      BIC      R5,(R3)          ;CLEAR BIT 4
010722 011304          MOV      (R3),R4          ;READ CONTENTS OF SILO STATUS
010724 042704 000700  .IIF NB <700>,      BIC      #700,R4          ;CLEAR UNWANTED BITS
010730 005704          TST      R4          ;WAS BIT 4 CLEARED
010732 001402          BEQ      2$
010734 005005          CLR      R5
          .IIF IDN      <DHSSR>,<DHLPR>,      HLT      2          ;SILO STATUS REGISTE
          .IIF IDN      <DHSSR>,<DHBCR>,      HLT      3          ;SILO STATUS REGISTE
R ERROR
R ERROR 010736          .IIF IDN      <DHSSR>,<DHSSR>,      HLT      4          ;SILO STATUS REGISTE
010736 104004          EMT      4
010740 104400          2$:      SCOPE
          .IF EQ BITX-40
          .NLIST
          BITX=100000
          BITC=17
          .LIST
          .IFF
          BITX=BITX+BITX
          BITC=BITC+1
          .ENDC
          REGTS6  +/DHSSR/,\BITX,+/SILO STATUS/, \BITC,700
          ;SILO STATUS REGISTER DATA TEST
          ;SET BIT 5 IN SILO STATUS TO 1
          ;VERIFY THAT BIT 5 WAS SET
          ;CLEAR BIT 5
          ;VERIFY THAT BIT 5 WAS CLEARED
010742
010742 012767 000340 167026 TS \XN,4000,2$
010750 012767 004000 007100 T104:  MOV      #340,PS          ;DISABLE ALL INTERRUPTS
010756 012767 011040 007066  MOV      #4000,ICOUNT        ;SET UP FOR 4000 ITERATIONS
          MOV      #2$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
          .IF NB <>
          MOV      #,FREEZ1     ;SET UP TO LOOP WITH DATA          : 3
          .ENDC
          XN=XN+1
010764 012777 004000 007016  MOV      #BIT11,@DHSCR        ;MASTER CLEAR INTERFACE
010772 016703 007030  MOV      DHSSR,R3          ;SET UP POINTER TO SILO STATUS
010776 012705 000040  MOV      #40,R5          ;BIT 5 WILL BE SET IN SILO STATUS
011002 010513  MOV      R5,(R3)          ;SET BIT 5
011004 011304  MOV      (R3),R4          ;GET CONTENTS OF SILO STATUS
011006 042704 000700  .IIF NB <700>,      BIC      #700,R4          ;CLEAR UNWANTED BITS
011012 020504  CMP      R5,R4          ;WAS BIT 5 SET
011014 001401  BEQ      1$
          .IIF IDN      <DHSSR>,<DHLPR>,      HLT      2          ;SILO STATUS REGISTE
          .IIF IDN      <DHSSR>,<DHBCR>,      HLT      3          ;SILO STATUS REGISTE
R ERROR
R ERROR 011016          .IIF IDN      <DHSSR>,<DHSSR>,      HLT      4          ;SILO STATUS REGISTE
011016 104004          EMT      4
011020 040513          1$:      BIC      R5,(R3)          ;CLEAR BIT 5
011022 011304          MOV      (R3),R4          ;READ CONTENTS OF SILO STATUS
011024 042704 000700  .IIF NB <700>,      BIC      #700,R4          ;CLEAR UNWANTED BITS
011030 005704          TST      R4          ;WAS BIT 5 CLEARED
011032 001402          BEQ      2$
011034 005005          CLR      R5
          .IIF IDN      <DHSSR>,<DHLPR>,      HLT      2          ;SILO STATUS REGISTE
R ERROR

```

```

R ERROR 011036 .IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
R ERROR 011036 104004 .IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE
011040 104400 EMT 4
100000 2$: SCOPE
000017 .IF EQ BITX-40
BITX=100000
BITC=17
.IFF
.NLIST
BITX=BITX+BITX
BITC=BITC+1
.LIST
.ENDC
011042 REGTS6 †/DHSSR/,\BITX,†/SILO STATUS/,\BITC,700

;SILO STATUS REGISTER DATA TEST
;SET BIT 17 IN SILO STATUS TO 1
;VERIFY THAT BIT 17 WAS SET
;CLEAR BIT 17
;VERIFY THAT BIT 17 WAS CLEARED

011042 TS \XN,4000,2$
011042 012767 000340 166726 T105: MOV #340,PS ;DISABLE ALL INTERRUPTS
011050 012767 004000 007000 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
011056 012767 011140 006766 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
011064 000106 012777 004000 006716 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
011072 016703 006730 MOV DHSSR,R3 ;SET UP POINTER TO SILO STATUS
011076 012705 100000 MOV #100000,R5 ;BIT 17 WILL BE SET IN SILO STATUS
011102 010513 MOV R5,(R3) ;SET BIT 17
011104 011304 MOV (R3),R4 ;GET CONTENTS OF SILO STATUS
011106 042704 000700 .IIF NB <700>, BIC #700,R4 ;CLEAR UNWANTED BITS
011112 020504 CMP R5,R4 ;WAS BIT 17 SET
011114 001401 BEQ 1$

.IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
.IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
R ERROR 011116 .IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE

011116 104004 EMT 4
011120 040513 1$: BIC R5,(R3) ;CLEAR BIT 17
011122 011304 MOV (R3),R4 ;READ CONTENTS OF SILO STATUS
011124 042704 000700 .IIF NB <700>, BIC #700,R4 ;CLEAR UNWANTED BITS
011130 005704 TST R4 ;WAS BIT 17 CLEARED
011132 001402 BEQ 2$
011134 005005 CLR R5

.IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
.IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
R ERROR 011136 .IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE

011136 104004 EMT 4
011140 104400 2$: SCOPE
.IF EQ BITX-40
.NLIST
BITX=100000
BITC=17
.LIST

```

```

000000      .IFF
000020      BITX=BITX+BITX
301          BITC=BITC+1
302          .ENDC
303          XBIT=BITX
304          CBIT=BITC
306          BITX=1
307          BITC=0
308          .REPT 15.
309          .NLIST
310          CCRBIT=-BITX-1
311          CLRBIT=CCRBIT&177767
312          BITCLR=CLRBIT
313          .LIST
314          REGTS7 +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC,177767,\BITCLR
315          .IF EQ BITX-4
316          .NLIST
317          BITX=BITX+BITX
318          BITC=BITC+1
319          .LIST
320          .ENDC
321          .NLIST
322          BITX=BITX+BITX
323          BITC=BITC+1
          .LIST
          .ENDR
          CCRBIT=-BITX-1
          CLRBIT=CCRBIT&177767
          BITCLR=CLRBIT
          REGTS7 +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC,177767,\BITCLR

```

177776  
177766  
177766  
011142

```

;LINE PARAMETER REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 1S
;CLEAR BIT 0
;VERIFY THAT BIT 0 WAS CLEARED
;RESTORE BIT 0
;VERIFY THAT BIT 0 WAS SET

```

```

011142      TS \XN,4000,2$
011142      012767 000340 166626 T106:  MOV #340,PS           ;DISABLE ALL INTERRUPTS
011150      012767 004000 006700      MOV #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
011156      012767 011244 006666      MOV #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
          .IF NB <>
          MOV #,FREEZ1           ;SET UP TO LOOP WITH DATA      ; 3
          .ENDC
          XN=XN+1
011164      000107 012777 004000 006616      MOV #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
011172      016703 006616      MOV DHLPR,R3         ;SET UP POINTER TO LINE PARAMETER
011176      012705 177766      MOV #177766,R5       ;(R5)=EXPECTED DATA
          ;IN LINE PARAMETER REGISTER, 177766
011202      012713 177767      MOV #177767,(R3)    ;SET ALL READ/WRITE BITS
          ;IN LINE PARAMETER REGISTER
011206      042713 000001      BIC #1,(R3)         ;CLEAR BIT 0
011212      011304      MOV (R3),R4         ;GET CONTENTS OF LINE PARAMETER
          .IIF NB <>, BIC #,R4       ;CLEAR UNWANTED BITS
011214      020504      CMP R5,R4           ;WAS BIT 0 CLEARED
011216      001401      BEQ 1$

```

```

011220          .IIF IDN      <DHLPR>,<DHLPR>,      HLT      2          ;LINE PARAMETER REGI
STER ERROR
011220 104002          EMT      2
          .IIF IDN      <DHLPR>,<DHBCR>,      HLT      3          ;LINE PARAMETER REGI
STER ERROR
          .IIF IDN      <DHLPR>,<DHSSR>,      HLT      4          ;LINE PARAMETER REGI
STER ERROR
011222 052713 000001  1$:      BIS      #1,(R3)          ;SET BIT 0
011226 011304          MOV      (R3),R4
          .IIF NB <>,      BIC      #,R4          ;CLEAR UNWANTED BITS
011230 022704 177767          CMP      #177767,R4      ;WAS BIT 0 SET
011234 001403          BEQ      2$
011236 012705 177767          MOV      #177767,R5      ;(R5)=EXPECTED DATA IN
          ;LINE PARAMETER REGISTER, 177767
011242          .IIF IDN      <DHLPR>,<DHLPR>,      HLT      2          ;LINE PARAMETER REGI
STER ERROR
011242 104002          EMT      2
          .IIF IDN      <DHLPR>,<DHBCR>,      HLT      3          ;LINE PARAMETER REGI
STER ERROR
          .IIF IDN      <DHLPR>,<DHSSR>,      HLT      4          ;LINE PARAMETER REGI
STER ERROR
011244 104400          2$:      SCOPE
          .IF EQ BITX-4
          .NLIST
          BITX=BITX+BITX
          BITC=BITC+1
          .LIST
          .ENDC
          BITX=BITX+BITX
          BITC=BITC+1
          CCRBIT=-BITX-1
          CLRBIT=CCRBIT&177767
          BITCLR=CLRBIT
011246          REGTS7 +/DHLPR/,\BITX,+/LINE PARAMETER/, \BITC,177767,\BITCLR

          ;LINE PARAMETER REGISTER DATA TEST
          ;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 1S
          ;CLEAR BIT 1
          ;VERIFY THAT BIT 1 WAS CLEARED
          ;RESTORE BIT 1
          ;VERIFY THAT BIT 1 WAS SET

011246          TS \XN,4000,2$
011246 012767 000340 166522 T107:  MOV      #340,PS          ;DISABLE ALL INTERRUPTS
011254 012767 004000 006574          MOV      #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
011262 012767 011350 006562          MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
          .IF NB <>
          MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
          .ENDC
          XN=XN+1
011270 000110          MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
011276 016703 004000 006512          MOV      DHLPR,R3          ;SET UP POINTER TO LINE PARAMETER
011302 012705 177765          MOV      #177765,R5      ;(R5)=EXPECTED DATA
          ;IN LINE PARAMETER REGISTER, 177765
011306 012713 177767          MOV      #177767,(R3)      ;SET ALL READ/WRITE BITS
          ;IN LINE PARAMETER REGISTER
011312 042713 000002          BIC      #2,(R3)          ;CLEAR BIT 1
011316 011304          MOV      (R3),R4          ;GET CONTENTS OF LINE PARAMETER
          .IIF NB <>,      BIC      #,R4          ;CLEAR UNWANTED BITS
011320 020504          CMP      R5,R4          ;WAS BIT 1 CLEARED
011322 001401          BEQ      1$
011324          .IIF IDN      <DHLPR>,<DHLPR>,      HLT      2          ;LINE PARAMETER REGI
STER ERROR
011324 104002          EMT      2

```

```

STER ERROR          .IIF IDN      <DHLPR>,<DHBCR>,      HLT      3          ;LINE PARAMETER REGI
STER ERROR          .IIF IDN      <DHLPR>,<DHSSR>,      HLT      4          ;LINE PARAMETER REGI
011326 052713 000002 1$:      BIS      #2,(R3)          ;SET BIT 1
011332 011304          MOV      (R3),R4
011334 022704 177767 .IIF NB <>,      BIC      #,R4          ;CLEAR UNWANTED BITS
011340 001403          CMP      #177767,R4      ;WAS BIT 1 SET
011342 012705 177767          BEQ      2$
                                MOV      #177767,R5          ;(R5)=EXPECTED DATA IN
                                ;LINE PARAMETER REGISTER, 177767
STER ERROR          .IIF IDN      <DHLPR>,<DHLPR>,      HLT      2          ;LINE PARAMETER REGI
011346 104002          EMT      2
STER ERROR          .IIF IDN      <DHLPR>,<DHBCR>,      HLT      3          ;LINE PARAMETER REGI
STER ERROR          .IIF IDN      <DHLPR>,<DHSSR>,      HLT      4          ;LINE PARAMETER REGI
011350 104400          2$:      SCOPE
                                .IF EQ BITX-4
                                .NLIST
                                BITX=BITX+BITX
                                BITC=BITC+1
                                .LIST
                                .ENDC
                                BITX=BITX+BITX
                                BITC=BITC+1
                                CCRBIT=-BITX-1
                                CLRBIT=CCRBIT&177767
                                BITCLR=CLRBIT
011352          REGTS7 +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC,177767,\BITCLR
                                ;LINE PARAMETER REGISTER DATA TEST
                                ;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 1S
                                ;CLEAR BIT 2
                                ;VERIFY THAT BIT 2 WAS CLEARED
                                ;RESTORE BIT 2
                                ;VERIFY THAT BIT 2 WAS SET
011352          TS \XN,4000,2$
011352 012767 000340 166416 T110:  MOV      #340,PS          ;DISABLE ALL INTERRUPTS
011360 012767 004000 006470          MOV      #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
011366 012767 011454 006456          MOV      #2$,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
011374 000111          MOV      #BIT11,0DHSCR      ;MASTER CLEAR INTERFACE
011402 016703 006406          MOV      DHLPR,R3          ;SET UP POINTER TO LINE PARAMETER
011406 012705 177763          MOV      #177763,R5        ;(R5)=EXPECTED DATA
                                ;IN LINE PARAMETER REGISTER, 177763
011412 012713 177767          MOV      #177767,(R3)      ;SET ALL READ/WRITE BITS
                                ;IN LINE PARAMETER REGISTER
011416 042713 000004          BIC      #4,(R3)          ;CLEAR BIT 2
011422 011304          MOV      (R3),R4          ;GET CONTENTS OF LINE PARAMETER
                                .IIF NB <>,      BIC      #,R4          ;CLEAR UNWANTED BITS
                                CMP      R5,R4          ;WAS BIT 2 CLEARED
                                BEQ      1$
STER ERROR          .IIF IDN      <DHLPR>,<DHLPR>,      HLT      2          ;LINE PARAMETER REGI
011430 104002          EMT      2
STER ERROR          .IIF IDN      <DHLPR>,<DHBCR>,      HLT      3          ;LINE PARAMETER REGI
STER ERROR          .IIF IDN      <DHLPR>,<DHSSR>,      HLT      4          ;LINE PARAMETER REGI

```

```

011432 052713 000004      1$:   BIS      #4,(R3)           ;SET BIT 2
011436 011304              MOV      (R3),R4
      .IIF NB <>,      BIC      #,R4           ;CLEAR UNWANTED BITS
011440 022704 177767      CMP      #177767,R4       ;WAS BIT 2 SET
011444 001403              BEQ      2$
011446 012705 177767      MOV      #177767,R5       ;(R5)=EXPECTED DATA IN
      ;LINE PARAMETER REGISTER, 177767
011452              .IIF IDN    <DHLPR>,<DHLPR>,      HLT      2           ;LINE PARAMETER REGI
STER ERROR
011452 104002              EMT      2
      .IIF IDN    <DHLPR>,<DHBCR>,      HLT      3           ;LINE PARAMETER REGI
STER ERROR
      .IIF IDN    <DHLPR>,<DHSSR>,      HLT      4           ;LINE PARAMETER REGI
STER ERROR
011454 104400
      000010
      000003
      000020
      000004
      177757
      177747
      177747
011456
2$:   SCOPE
      .IF EQ BITX-4
      BITX=BITX+BITX
      BITC=BITC+1
      .ENDC
      BITX=BITX+BITX
      BITC=BITC+1
      CCRBIT=-BITX-1
      CLRBIT=CCRBIT&177767
      BITCLR=CLRBIT
      REGTS7 ↑/DHLPR/,\BITX,↑/LINE PARAMETER/,\BITC,177767,\BITCLR

      ;LINE PARAMETER REGISTER DATA TEST
      ;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 15
      ;CLEAR BIT 4
      ;VERIFY THAT BIT 4 WAS CLEARED
      ;RESTORE BIT 4
      ;VERIFY THAT BIT 4 WAS SET

011456
011456 012767 000340 166312  TS \XN,4000,2$
011464 012767 004000 006364  T111:  MOV      #340,PS           ;DISABLE ALL INTERRUPTS
011472 012767 011560 006352  MOV      #4000,ICOUNT       ;SET UP FOR 4000 ITERATIONS
      MOV      #2$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV      #,FREEZ1       ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
011500 000112 012777 004000 006302  MOV      #BIT11,@DHSCR       ;MASTER CLEAR INTERFACE
011506 016703 006302          MOV      DHLPR,R3           ;SET UP POINTER TO LINE PARAMETER
011512 012705 177747          MOV      #177747,R5       ;(R5)=EXPECTED DATA
      ;IN LINE PARAMETER REGISTER, 177747
011516 012713 177767          MOV      #177767,(R3)       ;SET ALL READ/WRITE BITS
      ;IN LINE PARAMETER REGISTER
011522 042713 000020          BIC      #20,(R3)           ;CLEAR BIT 4
011526 011304              MOV      (R3),R4           ;GET CONTENTS OF LINE PARAMETER
      .IIF NB <>,      BIC      #,R4           ;CLEAR UNWANTED BITS
011530 020504              CMP      R5,R4             ;WAS BIT 4 CLEARED
011532 001401              BEQ      1$
011534              .IIF IDN    <DHLPR>,<DHLPR>,      HLT      2           ;LINE PARAMETER REGI
STER ERROR
011534 104002              EMT      2
      .IIF IDN    <DHLPR>,<DHBCR>,      HLT      3           ;LINE PARAMETER REGI
STER ERROR
      .IIF IDN    <DHLPR>,<DHSSR>,      HLT      4           ;LINE PARAMETER REGI
STEP ERROR
011536 052713 000020      1$:   BIS      #20,(R3)           ;SET BIT 4
011542 011304              MOV      (R3),R4
      .IIF NB <>,      BIC      #,R4           ;CLEAR UNWANTED BITS
011544 022704 177767      CMP      #177767,R4       ;WAS BIT 4 SET

```

```

011550 001403          BEQ 2$
011552 012705 177767  MOV #177767,R5          ;(R5)=EXPECTED DATA IN
                                ;LINE PARAMETER REGISTER, 177767
011556          .IIF IDN <DHLPR>,<DHLPR>, HLT 2          ;LINE PARAMETER REGI
STER ERROR
011556 104002          EMT 2
                                ;LINE PARAMETER REGI
STER ERROR          .IIF IDN <DHLPR>,<DHBCR>, HLT 3
                                ;LINE PARAMETER REGI
STER ERROR          .IIF IDN <DHLPR>,<DHSSR>, HLT 4
                                ;LINE PARAMETER REGI
011560 104400          2$: SCOPE
                                .IF EQ BITX-4
                                .NLIST
                                BITX=BITX+BITX
                                BITC=BITC+1
                                .LIST
                                .ENDC
                                BITX=BITX+BITX
                                BITC=BITC+1
                                CCRBIT=-BITX-1
                                CLRBIT=CCRBIT&177767
                                BITCLR=CLRBIT
011562          REGTS7 +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC,177767,\BITCLR
                                ;LINE PARAMETER REGISTER DATA TEST
                                ;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 15
                                ;CLEAR BIT 5
                                ;VERIFY THAT BIT 5 WAS CLEARED
                                ;RESTORE BIT 5
                                ;VERIFY THAT BIT 5 WAS SET

011562          TS \XN,4000,2$
011562 012767 000340 166206 T112: MOV #340,PS          ;DISABLE ALL INTERRUPTS
011570 012767 004000 006260 MOV #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
011576 012767 011664 006246 MOV #2$,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
011604 012777 004000 006176 MOV #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
011612 016703 006176          MOV DHLPR,R3          ;SET UP POINTER TO LINE PARAMETER
011616 012705 177727          MOV #177727,R5          ;(R5)=EXPECTED DATA
                                ;IN LINE PARAMETER REGISTER, 177727
011622 012713 177767          MOV #177767,(R3)      ;SET ALL READ/WRITE BITS
                                ;IN LINE PARAMETER REGISTER
011626 042713 000040          BIC #40,(R3)          ;CLEAR BIT 5
011632 011304          MOV (R3),R4          ;GET CONTENTS OF LINE PARAMETER
                                .IIF NB <>, BIC #,R4          ;CLEAR UNWANTED BITS
011634 020504          CMP R5,R4          ;WAS BIT 5 CLEARED
011636 001401          BEQ 1$
011640          .IIF IDN <DHLPR>,<DHLPR>, HLT 2          ;LINE PARAMETER REGI
STER ERROR
011640 104002          EMT 2
                                ;LINE PARAMETER REGI
STER ERROR          .IIF IDN <DHLPR>,<DHBCR>, HLT 3
                                ;LINE PARAMETER REGI
STER ERROR          .IIF IDN <DHLPR>,<DHSSR>, HLT 4
                                ;LINE PARAMETER REGI
011642 052713 000040          1$: BIS #40,(R3)          ;SET BIT 5
011646 011304          MOV (R3),R4
                                .IIF NB <>, BIC #,R4          ;CLEAR UNWANTED BITS
011650 022704 177767          CMP #177767,R4          ;WAS BIT 5 SET
011654 001403          BEQ 2$
011656 012705 177767          MOV #177767,R5          ;(R5)=EXPECTED DATA IN

```

```

011662          .IIF IDN      <DHLPR>, <DHLPR>,          ;LINE PARAMETER REGISTER, 177767
STER ERROR      HLT          2                          ;LINE PARAMETER REGI
011662 104002          EMT          2
011662          .IIF IDN      <DHLPR>, <DHBCR>,          ;LINE PARAMETER REGI
STER ERROR      HLT          3                          ;LINE PARAMETER REGI
011662          .IIF IDN      <DHLPR>, <DHSSR>,          ;LINE PARAMETER REGI
STER ERROR      HLT          4                          ;LINE PARAMETER REGI
011664 104400          2$:      SCOPE
                                .IF EQ BITX-4
                                .NLIST
                                BITX=BITX-BITX
                                BITC=BITC+1
                                .LIST
                                .ENDC
                                BITX=BITX-BITX
                                BITC=BITC+1
                                CCRBIT=-BITX-1
                                CLRBIT=CCRBIT&177767
                                BITCLR=CLRBIT
                                REGTS7  †/DHLPR/, \BITX, †/LINE PARAMETER/, \BITC, 177767, \BITCLR

                                ;LINE PARAMETER REGISTER DATA TEST
                                ;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 15
                                ;CLEAR BIT 6
                                ;VERIFY THAT BIT 6 WAS CLEARED
                                ;RESTORE BIT 6
                                ;VERIFY THAT BIT 6 WAS SET

000100
000006
177677
177667
177667
011666          TS \XN, 4000, 2$
011666 012767 000340 166102 T113:  MOV      #340, PS          ;DISABLE ALL INTERRUPTS
011674 012767 004000 006154      MOV      #4000, ICOUNT      ;SET UP FOR 4000 ITERATIONS
011702 012767 011770 006142      MOV      #2$, ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #, FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
011710 012777 004000 006072      MOV      #BIT11, @DHSCR      ;MASTER CLEAR INTERFACE
011716 016703 006072          MOV      DHLPR, R3          ;SET UP POINTER TO LINE PARAMETER
011722 012705 177667          MOV      #177667, R5      ;(R5)=EXPECTED DATA
                                MOV      #177767, (R3)      ;IN LINE PARAMETER REGISTER, 177667
                                BIC      #100, (R3)          ;SET ALL READ/WRITE BITS
                                MOV      (R3), R4          ;IN LINE PARAMETER REGISTER
                                .IIF NB <>, BIC      #, R4      ;CLEAR BIT 6
                                CMP      R5, R4          ;GET CONTENTS OF LINE PARAMETER
                                BEQ      1$,          ;CLEAR UNWANTED BITS
                                .IIF IDN <DHLPR>, <DHLPR>, HLT      2          ;WAS BIT 6 CLEARED
STER ERROR      ;LINE PARAMETER REGI
011744 104002          EMT          2
011744          .IIF IDN      <DHLPR>, <DHBCR>,          ;LINE PARAMETER REGI
STER ERROR      HLT          3                          ;LINE PARAMETER REGI
011744          .IIF IDN      <DHLPR>, <DHSSR>,          ;LINE PARAMETER REGI
STER ERROR      HLT          4                          ;LINE PARAMETER REGI
011746 052713 000100          1$:      BIS      #100, (R3)          ;SET BIT 6
011752 011304          MOV      (R3), R4
                                .IIF NB <>, BIC      #, R4      ;CLEAR UNWANTED BITS
                                CMP      #177767, R4      ;WAS BIT 6 SET
                                BEQ      2$,          ;(R5)=EXPECTED DATA IN
                                MOV      #177767, R5      ;LINE PARAMETER REGISTER, 177767
011766          .IIF IDN      <DHLPR>, <DHLPR>,          ;LINE PARAMETER REGI
STER ERROR      HLT          2                          ;LINE PARAMETER REGI

```



```

011766 104002          EMT 2
STER ERROR          .IIF IDN <DHLPR>,<DHBCR>, HLT 3 ;LINE PARAMETER REGI
                   .IIF IDN <DHLPR>,<DHSSR>, HLT 4 ;LINE PARAMETER REGI
011770 104400      2$: SCOPE
                   .IF EQ BITX-4
                   .NLIST
                   BITX=BITX+BITX
                   BITC=BITC+1
                   .LIST
                   .ENDC
000200             BITX=BITX+BITX
000007             BITC=BITC+1
177577             CCRBIT=-BITX-1
177567             CLRBIT=CCRBIT&177767
177567             BITCLP=CLRBIT
011772             REGTS7 †/DHLPR/,\BITX,†/LINE PARAMETER/,\BITC,177767,\BITCLR

                   ;LINE PARAMETER REGISTER DATA TEST
                   ;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 15
                   ;CLEAR BIT 7
                   ;VERIFY THAT BIT 7 WAS CLEARED
                   ;RESTORE BIT 7
                   ;VERIFY THAT BIT 7 WAS SET

011772             TS \XN,4000,2$
011772 012767 000340 165776 T114:  MOV #340,PS ;DISABLE ALL INTERRUPTS
012000 012767 004000 006050      MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
012006 012767 012074 006036      MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
                   .IF NB <>
                   MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
                   .ENDC
012014 012777 004000 005766      XN=XN+1
012022 016703 005766             MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
012026 012705 177567             MOV DHLPR,R3 ;SET UP POINTER TO LINE PARAMETER
                   MOV #177567,R5 ;(R5)=EXPECTED DATA
012032 012713 177767             MOV #177767,(R3) ;IN LINE PARAMETER REGISTER, 177567
012036 042713 000200             BIC #200,(R3) ;SET ALL READ/WRITE BITS
012042 011304             MOV (R3),R4 ;IN LINE PARAMETER REGISTER
                   .IIF NB <>, BIC #,R4 ;CLEAR BIT 7
012044 020504             CMP R5,R4 ;GET CONTENTS OF LINE PARAMETER
012046 001401             BEQ 1$ ;CLEAR UNWANTED BITS
012050             .IIF IDN <DHLPR>,<DHLPR>, HLT 2 ;WAS BIT 7 CLEARED
STER ERROR          .IIF IDN <DHLPR>,<DHLPR>, HLT 2 ;LINE PARAMETER REGI
012050 104002          EMT 2
STER ERROR          .IIF IDN <DHLPR>,<DHBCR>, HLT 3 ;LINE PARAMETER REGI
STER ERROR          .IIF IDN <DHLPR>,<DHSSR>, HLT 4 ;LINE PARAMETER REGI
012052 052713 000200      1$: BIS #200,(R3) ;SET BIT 7
012056 011304             MOV (R3),R4
                   .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
012060 022704 177767             CMP #177767,R4 ;WAS BIT 7 SET
012064 001403             BEQ 2$
012066 012705 177767             MOV #177767,R5 ;(R5)=EXPECTED DATA IN
012072             .IIF IDN <DHLPR>,<DHLPR>, HLT 2 ;LINE PARAMETER REGI
STER ERROR          .IIF IDN <DHLPR>,<DHLPR>, HLT 2 ;LINE PARAMETER REGI
012072 104002          EMT 2
STER ERROR          .IIF IDN <DHLPR>,<DHBCR>, HLT 3 ;LINE PARAMETER REGI
STER ERROR          .IIF IDN <DHLPR>,<DHBCR>, HLT 3 ;LINE PARAMETER REGI

```

```

                .IIF IDN      <DHLPR>,<DHSSR> ,      HLT      4      ;LINE PARAMETER REGI
STER ERROR
012074 104400
                2$: SCOPE
                .IF EQ BITX-4
                .NLIST
                BITX=BITX+BITX
                BITC=BITC+1
                .LIST
                .ENDC
                000400
                000010
                177377
                177367
                177367
012076
                BITX=BITX+BITX
                BITC=BITC+1
                CCRBIT=-BITX-1
                CLRBIT=CCRBIT&177767
                BITCLR=CLRBIT
                REGTS7 †/DHLPR/,\BITX,†/LINE PARAMETER/,\BITC,177767,\BITCLR

                ;LINE PARAMETER REGISTER DATA TEST
                ;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 1S
                ;CLEAR BIT 10
                ;VERIFY THAT BIT 10 WAS CLEARED
                ;RESTORE BIT 10
                ;VERIFY THAT BIT 10 WAS SET

012076
012076 012767 000340 165672 TS \XN,4000,2$
012104 012767 004000 005744 T115: MOV #340,PS ;DISABLE ALL INTERRUPTS
012112 012767 012200 005732 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
                MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
                .IF NB <>
                MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
                .ENDC
                XN=XN+1
012120 012777 004000 005662 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
012126 016703 005662 MOV DHLPR,R3 ;SET UP POINTER TO LINE PARAMETER
012132 012705 177367 MOV #177367,R5 ;(R5)=EXPECTED DATA
                ;IN LINE PARAMETER REGISTER, 177367
012136 012713 177767 MOV #177767,(R3) ;SET ALL READ/WRITE BITS
                ;IN LINE PARAMETER REGISTER
012142 042713 000400 BIC #400,(R3) ;CLEAR BIT 10
012146 011304 MOV (R3),R4 ;GET CONTENTS OF LINE PARAMETER
                .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
012150 020504 CMP R5,R4 ;WAS BIT 10 CLEARED
012152 001401 BEQ 1$
012154 .IIF IDN <DHLPR>,<DHLPR> ,      HLT      2      ;LINE PARAMETER REGI
STER ERROR
012154 104002 EMT 2
                .IIF IDN <DHLPR>,<DHBCR> ,      HLT      3      ;LINE PARAMETER REGI
STER ERROR
                .IIF IDN <DHLPR>,<DHSSR> ,      HLT      4      ;LINE PARAMETER REGI
STER ERROR
012156 052713 000400 1$: BIS #400,(R3) ;SET BIT 10
012162 011304 MOV (R3),R4
                .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
012164 022704 177767 CMP #177767,R4 ;WAS BIT 10 SET
012170 001403 BEQ 2$
012172 012705 177767 MOV #177767,R5 ;(R5)=EXPECTED DATA IN
                ;LINE PARAMETER REGISTER, 177767
012176 .IIF IDN <DHLPR>,<DHLPR> ,      HLT      2      ;LINE PARAMETER REGI
STER ERROR
012176 104002 EMT 2
                .IIF IDN <DHLPR>,<DHBCR> ,      HLT      3      ;LINE PARAMETER REGI
STER ERROR
                .IIF IDN <DHLPR>,<DHSSR> ,      HLT      4      ;LINE PARAMETER REGI
STER ERROR
012200 104400 2$: SCOPE

```

```

                                .IF EQ BITX-4
                                .NLIST
                                BITX=BITX+BITX
                                BITC=BITC+1
                                .LIST
                                .ENDC
                                BITX=BITX+BITX
                                BITC=BITC+1
                                CCRBIT=-BITX-1
                                CLRBIT=CCRBIT&177767
                                BITCLR=CLRBIT
012202 REGTS7 +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC,177767,\BITCLR

                                ;LINE PARAMETER REGISTER DATA TEST
                                ;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 15
                                ;CLEAR BIT 11
                                ;VERIFY THAT BIT 11 WAS CLEARED
                                ;RESTORE BIT 11
                                ;VERIFY THAT BIT 11 WAS SET

012202 TS \XN,4000,2$
012202 012767 000340 165566 T116: MOV #340,PS ;DISABLE ALL INTERRUPTS
012210 012767 004000 005640 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
012216 012767 012304 005626 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

012224 000117 012777 004000 005556 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
012232 016703 005556 MOV DHLPR,R3 ;SET UP POINTER TO LINE PARAMETER
012236 012705 176767 MOV #176767,R5 ;(R5)=EXPECTED DATA
                                ;IN LINE PARAMETER REGISTER, 176767
012242 012713 177767 MOV #177767,(R3) ;SET ALL READ/WRITE BITS
                                ;IN LINE PARAMETER REGISTER
012246 042713 001000 BIC #1000,(R3) ;CLEAR BIT 11
012252 011304 MOV (R3),R4 ;GET CONTENTS OF LINE PARAMETER
                                .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
012254 020504 CMP R5,R4 ;WAS BIT 11 CLEARED
012256 001401 BEQ 1$
012260 .IIF IDN <DHLPR>,<DHLPR>, HLT 2 ;LINE PARAMETER REGI
STER ERROR
012260 104002 EMT 2
                                .IIF IDN <DHLPR>,<DHBCR>, HLT 3 ;LINE PARAMETER REGI
STER ERROR
                                .IIF IDN <DHLPR>,<DHSSR>, HLT 4 ;LINE PARAMETER REGI
STER ERROR
012262 052713 001000 1$: BIS #1000,(R3) ;SET BIT 11
012266 011304 MOV (R3),R4
                                .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
012270 022704 177767 CMP #177767,R4 ;WAS BIT 11 SET
012274 001403 BEQ 2$
012276 012705 177767 MOV #177767,R5 ;(R5)=EXPECTED DATA IN
                                ;LINE PARAMETER REGISTER, 177767
012302 .IIF IDN <DHLPR>,<DHLPR>, HLT 2 ;LINE PARAMETER REGI
STER ERROR
012302 104002 EMT 2
                                .IIF IDN <DHLPR>,<DHBCR>, HLT 3 ;LINE PARAMETER REGI
STER ERROR
                                .IIF IDN <DHLPR>,<DHSSR>, HLT 4 ;LINE PARAMETER REGI
STER ERROR
012304 104400 2$: SCOPE
                                .IF EQ BITX-4
                                .NLIST

```

```

                                BITX=BITX+BITX
                                BITC=BITC+1
                                .LIST
                                .ENDC
                                BITX=BITX+BITX
                                BITC=BITC+1
                                CCRBIT=-BITX-1
                                CLRBIT=CCRBIT&177767
                                BITCLR=CLRBIT
012306 REGTS7  +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC,177767,\BITCLR

                                ;LINE PARAMETER REGISTER DATA TEST
                                ;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 15
                                ;CLEAR BIT 12
                                ;VERIFY THAT BIT 12 WAS CLEARED
                                ;RESTORE BIT 12
                                ;VERIFY THAT BIT 12 WAS SET

012306 TS \XN,4000,2$
012306 012767 000340 165462 T117:  MOV  #340,PS          ;DISABLE ALL INTERRUPTS
012314 012767 004000 005534      MOV  #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
012322 012767 012410 005522      MOV  #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV  #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
012330 000120 012777 004000 005452      MOV  #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
012336 016703 005452      MOV  DHLPR,R3        ;SET UP POINTER TO LINE PARAMETER
012342 012705 175767      MOV  #175767,R5     ;(R5)=EXPECTED DATA
                                ;IN LINE PARAMETER REGISTER, 175767
012346 012713 177767      MOV  #177767,(R3)   ;SET ALL READ/WRITE BITS
                                ;IN LINE PARAMETER REGISTER
012352 042713 002000      BIC  #2000,(R3)     ;CLEAR BIT 12
012356 011304      MOV  (R3),R4        ;GET CONTENTS OF LINE PARAMETER
                                .IIF NB  <>,
                                CMP  R5,R4            ;CLEAR UNWANTED BITS
                                BEQ  1$              ;WAS BIT 12 CLEARED
012360 020504      .IIF IDN          <DHLPR>,<DHLPR>,    HLT  2          ;LINE PARAMETER REGI
012362 001401      EMT  2
012364      .IIF IDN          <DHLPR>,<DHBCR>,    HLT  3          ;LINE PARAMETER REGI
STER ERROR 012364 104002
                                .IIF IDN          <DHLPR>,<DHSSR>,    HLT  4          ;LINE PARAMETER REGI
STER ERROR
012366 052713 002000      1$:  BIS  #2000,(R3)   ;SET BIT 12
012372 011304      MOV  (R3),R4
                                .IIF NB  <>,
                                BIC  #,R4            ;CLEAR UNWANTED BITS
012374 022704 177767      CMP  #177767,R4     ;WAS BIT 12 SET
012400 001403      BEQ  2$
012402 012705 177767      MOV  #177767,R5     ;(R5)=EXPECTED DATA IN
                                ;LINE PARAMETER REGISTER, 177767
012406      .IIF IDN          <DHLPR>,<DHLPR>,    HLT  2          ;LINE PARAMETER REGI
STER ERROR 012406 104002
                                EMT  2
012406      .IIF IDN          <DHLPR>,<DHBCR>,    HLT  3          ;LINE PARAMETER REGI
STER ERROR
                                .IIF IDN          <DHLPR>,<DHSSR>,    HLT  4          ;LINE PARAMETER REGI
STER ERROR
012410 104400      2$:  SCOPE
                                .IF EQ  BITX-4
                                .NLIST
                                BITX=BITX+BITX
                                BITC=BITC+1

```

```

004000
000013
173777
173757
173767
012412
    .LIST
    .ENDC
    BITX=BITX+BITX
    BITC=BITC+1
    CCRBIT=-BITX-1
    CLRBIT=CCRBIT&177767
    BITCLR=CLRBIT
    REGTS7 +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC,177767,\BITCLR

    ;LINE PARAMETER REGISTER DATA TEST
    ;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 15
    ;CLEAR BIT 13
    ;VERIFY THAT BIT 13 WAS CLEARED
    ;RESTORE BIT 13
    ;VERIFY THAT BIT 13 WAS SET

012412 012767 000340 165356 TS \XN,4000,2$
012412 012767 004000 005430 T120:  MOV #340,PS ;DISABLE ALL INTERRUPTS
012420 012767 004000 005430      MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
012426 012767 012514 005416      MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
    .IF NB <>
    MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
    .ENDC
    XN=XN+1
012434 012777 004000 005346      MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
012442 016703 005346      MOV DHLPR,R3 ;SET UP POINTER TO LINE PARAMETER
012446 012705 173767      MOV #173767,R5 ;(R5)=EXPECTED DATA
    MOV #177767,(R3) ;IN LINE PARAMETER REGISTER, 173767
    ;SET ALL READ/WRITE BITS
    ;IN LINE PARAMETER REGISTER
012452 012713 177767      MOV #177767,(R3)
012456 042713 004000      BIC #4000,(R3) ;CLEAR BIT 13
012462 011304      MOV (R3),R4 ;GET CONTENTS OF LINE PARAMETER
    .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
012464 020504      CMP R5,R4 ;WAS BIT 13 CLEARED
012466 001401      BEQ 1$
012470      .IIF IDN <DHLPR>,<DHLPR>, HLT 2 ;LINE PARAMETER REGI
STER ERROR
012470 104002      EMT 2
    .IIF IDN <DHLPR>,<DHBCR>, HLT 3 ;LINE PARAMETER REGI
STER ERROR
    .IIF IDN <DHLPR>,<DHSSR>, HLT 4 ;LINE PARAMETER REGI
STER ERROR
012472 052713 004000      1$: BIS #4000,(R3) ;SET BIT 13
012476 011304      MOV (R3),R4
    .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
012500 022704 177767      CMP #177767,R4 ;WAS BIT 13 SET
012504 001403      BEQ 2$
012506 012705 177767      MOV #177767,R5 ;(R5)=EXPECTED DATA IN
    ;LINE PARAMETER REGISTER, 177767
012512      .IIF IDN <DHLPR>,<DHLPR>, HLT 2 ;LINE PARAMETER REGI
STER ERROR
012512 104002      EMT 2
    .IIF IDN <DHLPR>,<DHBCR>, HLT 3 ;LINE PARAMETER REGI
STER ERROR
    .IIF IDN <DHLPR>,<DHSSR>, HLT 4 ;LINE PARAMETER REGI
STER ERROR
012514 104400      2$: SCOPE
    .IF EQ BITX-4
    .NLIST
    BITX=BITX+BITX
    BITC=BITC+1
    .LIST
    .ENDC

```

```

010000 BITX=BITX+BITX
000014 BITC=BITC+1
167777 CCRBIT=-BITX-1
167767 CLRBIT=CCRBIT&177767
167767 BITCLR=CLRBIT
012516 REGTS7 +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC,177767,\BITCLR

;LINE PARAMETER REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 1S
;CLEAR BIT 14
;VERIFY THAT BIT 14 WAS CLEARED
;RESTORE BIT 14
;VERIFY THAT BIT 14 WAS SET

012516 TS \XN,4000,2$
012516 012767 000340 165252 T121: MOV #340,PS ;DISABLE ALL INTERRUPTS
012524 012767 004000 005324 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
012532 012767 012620 005312 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
.IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1
012540 000122 012777 004000 005242 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
012546 016703 005242 MOV DHLPR,R3 ;SET UP POINTER TO LINE PARAMETER
012552 012705 167767 MOV #167767,R5 ;(R5)=EXPECTED DATA
;IN LINE PARAMETER REGISTER, 167767
012556 012713 177767 MOV #177767,(R3) ;SET ALL READ/WRITE BITS
;IN LINE PARAMETER REGISTER
012562 042713 010000 BIC #10000,(R3) ;CLEAR BIT 14
012566 011304 MOV (R3),R4 ;GET CONTENTS OF LINE PARAMETER
.IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
012570 020504 CMP R5,R4 ;WAS BIT 14 CLEARED
012572 001401 BEQ 1$
012574 .IIF IDN <DHLPR>,<DHLPR>, HLT 2 ;LINE PARAMETER REGI
STER ERROR
012574 104002 EMT 2
.IIF IDN <DHLPR>,<DHBCR>, HLT 3 ;LINE PARAMETER REGI
STER ERROR
.IIF IDN <DHLPR>,<DHSSR>, HLT 4 ;LINE PARAMETER REGI
STER ERROR
012576 052713 010000 1$: BIS #10000,(R3) ;SET BIT 14
012602 011304 MOV (R3),R4
.IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
012604 022704 177767 CMP #177767,R4 ;WAS BIT 14 SET
012610 001403 BEQ 2$
012612 012705 177767 MOV #177767,R5 ;(R5)=EXPECTED DATA IN
;LINE PARAMETER REGISTER, 177767
012616 .IIF IDN <DHLPR>,<DHLPR>, HLT 2 ;LINE PARAMETER REGI
STER ERROR
012616 104002 EMT 2
.IIF IDN <DHLPR>,<DHBCR>, HLT 3 ;LINE PARAMETER REGI
STER ERROR
.IIF IDN <DHLPR>,<DHSSR>, HLT 4 ;LINE PARAMETER REGI
STER ERROR
012620 104400 2$: SCOPE
.IF EQ BITX-4
.NLIST
BITX=BITX+BITX
BITC=BITC+1
.LIST
.ENDC
020000 BITX=BITX+BITX
000015 BITC=BITC+1

```

```

157777
157767
157767
012622
                                CCRBIT=-BITX-1
                                CLRBIT=CCRBIT&177767
                                BITCLR=CLRBIT
                                REGTS7 †/DHLPR/,\BITX,†/LINE PARAMETER/,\BITC,177767,\BITCLR

                                ;LINE PARAMETER REGISTER DATA TEST
                                ;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 15
                                ;CLEAR BIT 15
                                ;VERIFY THAT BIT 15 WAS CLEARED
                                ;RESTORE BIT 15
                                ;VERIFY THAT BIT 15 WAS SET

012622      012622      012767      000340      165146      TS \XN,4000,2$
012622      012767      004000      005220      T122:      MOV      #340,PS      ;DISABLE ALL INTERRUPTS
012630      012767      004000      005220      MOV      #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
012636      012767      012724      005206      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

012644      000123      012777      004000      005136      MOV      #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
012652      016703      005136      MOV      DHLPR,R3      ;SET UP POINTER TO LINE PARAMETER
012656      012705      157767      MOV      #157767,R5    ;(R5)=EXPECTED DATA
                                ;IN LINE PARAMETER REGISTER, 157767
012662      012713      177767      MOV      #177767,(R3)  ;SET ALL READ/WRITE BITS
                                ;IN LINE PARAMETER REGISTER
012666      042713      020000      BIC      #20000,(R3)   ;CLEAR BIT 15
012672      011304      MOV      (R3),R4      ;GET CONTENTS OF LINE PARAMETER
                                .IIF NB      <>,      BIC      #,R4      ;CLEAR UNWANTED BITS
012674      020504      CMP      R5,R4      ;WAS BIT 15 CLEARED
012676      001401      BEQ      1$
012700      .IIF IDN      <DHLPR>,<DHLPR>,      HLT      2      ;LINE PARAMETER REGI
STER ERROR
012700      104002      EMT      2
                                .IIF IDN      <DHLPR>,<DHBCR>,      HLT      3      ;LINE PARAMETER REGI
STER ERROR
                                .IIF IDN      <DHLPR>,<DHSSR>,      HLT      4      ;LINE PARAMETER REGI
STER ERROR
012702      052713      020000      1$:      BIS      #20000,(R3) ;SET BIT 15
012706      011304      MOV      (R3),R4
                                .IIF NB      <>,      BIC      #,R4      ;CLEAR UNWANTED BITS
012710      022704      177767      CMP      #177767,R4    ;WAS BIT 15 SET
012714      001403      BEQ      2$
012716      012705      177767      MOV      #177767,R5    ;(R5)=EXPECTED DATA IN
                                ;LINE PARAMETER REGISTER, 177767
012722      .IIF IDN      <DHLPR>,<DHLPR>,      HLT      2      ;LINE PARAMETER REGI
STER ERROR
012722      104002      EMT      2
                                .IIF IDN      <DHLPR>,<DHBCR>,      HLT      3      ;LINE PARAMETER REGI
STER ERROR
                                .IIF IDN      <DHLPR>,<DHSSR>,      HLT      4      ;LINE PARAMETER REGI
STER ERROR
012724      104400      2$:      SCOPE
                                .IF EQ      BITX-4
                                .NLIST
                                BITX=BITX+BITX
                                BITC=BITC+1
                                .LIST
                                .ENDC
                                BITX=BITX+BITX
                                BITC=BITC+1
                                CCRBIT=-BITX-1
                                CLRBIT=CCRBIT&177767
040000
000016
137777
137767

```

```

012726      137767      BITCLR=CLRBIT
012726      REGTS7  +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC,177767,\BITCLR

;LINE PARAMETER REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 1S
;CLEAR BIT 16
;VERIFY THAT BIT 16 WAS CLEARED
;RESTORE BIT 16
;VERIFY THAT BIT 16 WAS SET

012726      012726      000340      165042      TS \XN,4000,2$
012726      012767      004000      005114      T123:  MOV      #340,PS      ;DISABLE ALL INTERRUPTS
012734      012767      004000      005114      MOV      #4000,ICOUNT    ;SET UP FOR 4000 ITERATIONS
012742      012767      013030      005102      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

012750      000124      012777      004000      005032      MOV      #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
012756      016703      005032      MOV      DHLPR,R3      ;SET UP POINTER TO LINE PARAMETER
012762      012705      137767      MOV      #137767,R5     ;(R5)=EXPECTED DATA
;IN LINE PARAMETER REGISTER, 137767
;SET ALL READ/WRITE BITS
;IN LINE PARAMETER REGISTER
;CLEAR BIT 16
;GET CONTENTS OF LINE PARAMETER
;CLEAR UNWANTED BITS
;WAS BIT 16 CLEARED

012766      012713      177767      MOV      #177767,(R3)

012772      042713      040000      BIC      #40000,(R3)
012776      011304      MOV      (R3),R4

.IIF NB      <>,
BIC      #,R4
CMP      R5,R4
BEQ      1$

013000      020504      .IIF IDN      <DHLPR>,<DHLPR>,      HLT      2      ;LINE PARAMETER REGI
013002      001401
013004
STER ERROR
013004      104002      EMT      2
.IIF IDN      <DHLPR>,<DHBCR>,      HLT      3      ;LINE PARAMETER REGI
.IIF IDN      <DHLPR>,<DHSSR>,      HLT      4      ;LINE PARAMETER REGI

STER ERROR
013006      052713      040000      1$:      BIS      #40000,(R3)      ;SET BIT 16
013012      011304      MOV      (R3),R4

.IIF NB      <>,
BIC      #,R4      ;CLEAR UNWANTED BITS
CMP      #177767,R4     ;WAS BIT 16 SET
BEQ      2$
MOV      #177767,R5     ;(R5)=EXPECTED DATA IN
;LINE PARAMETER REGISTER, 177767

013026      .IIF IDN      <DHLPR>,<DHLPR>,      HLT      2      ;LINE PARAMETER REGI
STER ERROR
013026      104002      EMT      2
.IIF IDN      <DHLPR>,<DHBCR>,      HLT      3      ;LINE PARAMETER REGI
STER ERROR
.IIF IDN      <DHLPR>,<DHSSR>,      HLT      4      ;LINE PARAMETER REGI
STER ERROR
013030      104400      2$:      SCOPE
.IIF EQ      BITX-4
.NLIST
BITX=BITX+BITX
BITC=BITC+1
.LIST
.ENDC
100000      BITX=BITX+BITX
000017      BITC=BITC+1
077777      CCRBIT=-BITX-1
077767      CLRBIT=CCRBIT&177767
077767      BITCLR=CLRBIT
013032      REGTS7  +/DHLPR/,\BITX,+/LINE PARAMETER/,\BITC,177767,\BITCLR

```



```

;LINE PARAMETER REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN LINE PARAMETER REGISTER TO 1S
;CLEAR BIT 17
;VERIFY THAT BIT 17 WAS CLEARED
;RESTORE BIT 17
;VERIFY THAT BIT 17 WAS SET
    
```

```

013032      TS \XN,4000,2$
013032 012767 000340 164736 T124:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
013040 012767 004000 005010      MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
013046 012767 013134 004776      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
013054      000125
013054 012777 004000 004726      MOV    #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
013062 016703 004726 004726      MOV    @DHLPR,R3      ;SET UP POINTER TO LINE PARAMETER
013066 012705 077767 077767      MOV    #77767,R5      ;(R5)=EXPECTED DATA
                                ;IN LINE PARAMETER REGISTER, 77767
013072 012713 177767 177767      MOV    #177767,(R3)   ;SET ALL READ/WRITE BITS
                                ;IN LINE PARAMETER REGISTER
013076 042713 100000 100000      BIC    #100000,(R3)   ;CLEAR BIT 17
013102 011304 011304 011304      MOV    (R3),R4        ;GET CONTENTS OF LINE PARAMETER
                                .IIF NB <>, BIC    #,R4        ;CLEAR UNWANTED BITS
013104 020504 020504 020504      CMP    R5,R4          ;WAS BIT 17 CLEARED
013106 001401 001401 001401      BEQ    1$
013110      .IIF IDN    <DHLPR>,<DHLPR>, HLT    2           ;LINE PARAMETER REGI
STER ERROR
013110 104002 104002 104002      EMT    2
                                .IIF IDN    <DHLPR>,<DHBCR>, HLT    3           ;LINE PARAMETER REGI
STER ERROR
                                .IIF IDN    <DHLPR>,<DHSSR>, HLT    4           ;LINE PARAMETER REGI
STER ERROR
013112 052713 100000 100000      1$:   BIS    #100000,(R3) ;SET BIT 17
013116 011304 011304 011304      MOV    (R3),R4
                                .IIF NB <>, BIC    #,R4        ;CLEAR UNWANTED BITS
013120 022704 177767 177767      CMP    #177767,R4    ;WAS BIT 17 SET
013124 001403 001403 001403      BEQ    2$
013126 012705 177767 177767      MOV    #177767,R5    ;(R5)=EXPECTED DATA IN
                                ;LINE PARAMETER REGISTER, 177767
013132      .IIF IDN    <DHLPR>,<DHLPR>, HLT    2           ;LINE PARAMETER REGI
STER ERROR
013132 104002 104002 104002      EMT    2
                                .IIF IDN    <DHLPR>,<DHBCR>, HLT    3           ;LINE PARAMETER REGI
STER ERROR
                                .IIF IDN    <DHLPR>,<DHSSR>, HLT    4           ;LINE PARAMETER REGI
STER ERROR
013134 104400 104400 104400      2$:   SCOPE
                                .IF EQ  BITX-4
                                .NLIST
                                BITX=BITX+BITX
                                BITC=BITC+1
                                .LIST
                                .ENDC
                                BITX=BITX+BITX
                                BITC=BITC+1
0325      000000
0326      000000
0327      000020
0328      000001
0328      000000
0330      000020
0331      .REPT 16.
                                .NLIST
    
```

332  
333  
334  
335  
336  
337  
338  
339  
340

```

CLRBIT=-BITX-1
BITCLR=CLRBIT
.LIST
REGTS7 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC,177777,\BITCLR
.NLIST
BITX=BITX+BITX
BITC=BITC+1
.LIST
.ENDR
CLRBIT=-BITX-1
BITCLR=CLRBIT
REGTS7 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC,177777,\BITCLR
    
```

177776  
177776  
013136

```

;BREAK CONTROL REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 1S
;CLEAR BIT 0
;VERIFY THAT BIT 0 WAS CLEARED
;RESTORE BIT 0
;VERIFY THAT BIT 0 WAS SET
    
```

```

013136 TS \XN,4000,2$
013136 012767 000340 164632 T125: MOV #340,PS ;DISABLE ALL INTERRUPTS
013144 012767 004000 004704 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
013152 012767 013240 004672 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
      .ENDC
013160 000126 XN=XN+1
013166 012777 004000 004622 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
013172 012705 177776 MOV DHBCR,R3 ;SET UP POINTER TO BREAK CONTROL
      MOV #177776,R5 ;(R5)=EXPECTED DATA
013176 012713 177777 MOV #177777,(R3) ;IN BREAK CONTROL REGISTER, 177776
      MOV #177777,(R3) ;SET ALL READ/WRITE BITS
      ;IN BREAK CONTROL REGISTER
013202 042713 000001 BIC #1,(R3) ;CLEAR BIT 0
013206 011304 MOV (R3),R4 ;GET CONTENTS OF BREAK CONTROL
      .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
013210 020504 CMP R5,R4 ;WAS BIT 0 CLEARED
013212 001401 BEQ 1$
      .IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
TER ERROR 013214 .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
TER ERROR 013214 104003 EMT 3
      .IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
TER ERROR 013216 052713 000001 1$: BIS #1,(R3) ;SET BIT 0
013222 011304 MOV (R3),R4
      .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
013224 022704 177777 CMP #177777,R4 ;WAS BIT 0 SET
013230 001403 BEQ 2$
013232 012705 177777 MOV #177777,R5 ;(R5)=EXPECTED DATA IN
      ;BREAK CONTROL REGISTER, 177777
      .IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
TER ERROR 013236 .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
TER ERROR 013236 104003 EMT 3
      .IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
TER ERROR 013240 104400 2$: SCOPE
000002 BITX=BITX+BITX
000001 BITC=BITC+1
    
```

```

177775 CLRBIT=-BITX-1
177775 BITCLR=CLRBIT
013242 REGTS7 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC,177777,\BITCLR

;BREAK CONTROL REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 1S
;CLEAR BIT 1
;VERIFY THAT BIT 1 WAS CLEARED
;RESTORE BIT 1
;VERIFY THAT BIT 1 WAS SET

013242 TS \XN,4000,2$
013242 012767 000340 164526 T126: MOV #340,PS ;DISABLE ALL INTERRUPTS
013250 012767 004000 004600 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
013256 012767 013344 004566 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

013264 000127 012777 004000 004516 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
013272 016703 004526 MOV DHBCR,R3 ;SET UP POINTER TO BREAK CONTROL
013276 012705 177775 MOV #177775,R5 ;(R5)=EXPECTED DATA
;IN BREAK CONTROL REGISTER, 177775
;SET ALL READ/WRITE BITS
;IN BREAK CONTROL REGISTER

013302 012713 177777 MOV #177777,(R3)
013306 042713 000002 BIC #2,(R3) ;CLEAR BIT 1
013312 011304 MOV (R3),R4 ;GET CONTENTS OF BREAK CONTROL
.IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
CMP R5,R4 ;WAS BIT 1 CLEARED
BEQ 1$

.IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
.IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
.IIF IDN EMT 3 <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS

1$: BIS #2,(R3) ;SET BIT 1
MOV (R3),R4
.IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
CMP #177777,R4 ;WAS BIT 1 SET
BEQ 2$
MOV #177777,R5 ;(R5)=EXPECTED DATA IN
;BREAK CONTROL REGISTER, 177777

.IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
.IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
.IIF IDN EMT 3 <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS

013344 104400 2$: SCOPE
000004 BITX=BITX+BITX
000002 BITC=BITC+1
177773 CLRBIT=-BITX-1
177773 BITCLR=CLRBIT
013346 REGTS7 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC,177777,\BITCLR

;BREAK CONTROL REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 1S
;CLEAR BIT 2
;VERIFY THAT BIT 2 WAS CLEARED
;RESTORE BIT 2

```

```

                                ;VERIFY THAT BIT 2 WAS SET
013346                                TS \XN,4000,2$
013346 012767 000340 164422 T127: MOV #340,PS ;DISABLE ALL INTERRUPTS
013354 012767 004000 004474 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
013362 012767 013450 004462 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC
                                XN=XN+1
013370 000130 012777 004000 004412 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
013376 016703 004422 MOV DHBCR,R3 ;SET UP POINTER TO BREAK CONTROL
013402 012705 177773 MOV #177773,R5 ;(R5)=EXPECTED DATA
                                MOV #177777,(R3) ;IN BREAK CONTROL REGISTER, 177773
                                MOV #177777,(R3) ;SET ALL READ/WRITE BITS
                                BIC #4,(R3) ;CLEAR BIT 2 ;IN BREAK CONTROL REGISTER
013412 042713 000004 MOV (R3),R4 ;GET CONTENTS OF BREAK CONTROL
013416 011304 .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
                                CMP R5,R4 ;WAS BIT 2 CLEARED
013420 020504 BEQ 1$
013422 001401 .IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
TER ERROR 013424 .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
TER ERROR 013424 104003 EMT 3 ;BREAK CONTROL REGIS
                                .IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
TER ERROR 013426 052713 000004 1$: BIS #4,(R3) ;SET BIT 2
013432 011304 MOV (R3),R4
                                .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
013434 022704 177777 CMP #177777,R4 ;WAS BIT 2 SET
013440 001403 BEQ 2$
013442 012705 177777 MOV #177777,R5 ;(R5)=EXPECTED DATA IN
                                .IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGISTER, 177777
TER ERROR 013446 .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
TER ERROR 013446 104003 EMT 3 ;BREAK CONTROL REGIS
                                .IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
TER ERROR 013450 104400 2$: SCOPE
000010 BITX=BITX+BITX
000003 BITC=BITC+1
177767 CLRBIT=-BITX-1
177767 BITCLR=CLRBIT
013452 REGTS7 +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC,177777,\BITCLR
                                ;BREAK CONTROL REGISTER DATA TEST
                                ;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 1S
                                ;CLEAR BIT 3
                                ;VERIFY THAT BIT 3 WAS CLEARED
                                ;RESTORE BIT 3
                                ;VERIFY THAT BIT 3 WAS SET
013452                                TS \XN,4000,2$
013452 012767 000340 164316 T130: MOV #340,PS ;DISABLE ALL INTERRUPTS
013460 012767 004000 004370 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
013466 012767 013554 004356 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC

```

```

000131
013474 012777 004000 004306 XN=XN+1
013502 016703 004316
013506 012705 177767
MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
MOV DHBCR,R3 ;SET UP POINTER TO BREAK CONTROL
MOV #177767,R5 ;(R5)=EXPECTED DATA
;IN BREAK CONTROL REGISTER, 177767
013512 012713 177777 MOV #177777,(R3) ;SET ALL READ/WRITE BITS
;IN BREAK CONTROL REGISTER
013516 042713 000010 BIC #10,(R3) ;CLEAR BIT 3
013522 011304 MOV (R3),R4 ;GET CONTENTS OF BREAK CONTROL
.IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
013524 020504 CMP R5,R4 ;WAS BIT 3 CLEARED
013526 001401 BEQ 1$
.IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
.IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
013530 104003 EMT 3
.IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
013532 052713 000010 1$: BIS #10,(R3) ;SET BIT 3
013536 011304 MOV (R3),R4
.IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
013540 022704 177777 CMP #177777,R4 ;WAS BIT 3 SET
013544 001403 BEQ 2$
013546 012705 177777 MOV #177777,R5 ;(R5)=EXPECTED DATA IN
;BREAK CONTROL REGISTER, 177777
.IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
.IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
013552 104003 EMT 3
.IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
013554 104400 2$: SCOPE
000020 BITX=BITX+BITX
000004 BITC=BITC+1
177757 CLRBIT=-BITX-1
177757 BITCLR=CLRBIT
013556 REGIS7 +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC,177777,\BITCLR
;BREAK CONTROL REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 1S
;CLEAR BIT 4
;VERIFY THAT BIT 4 WAS CLEARED
;RESTORE BIT 4
;VERIFY THAT BIT 4 WAS SET
013556 012767 000340 164212 TS \XN,4000,2$
013556 012767 000340 164212 T131: MOV #340,PS ;DISABLE ALL INTERRUPTS
013564 012767 004000 004264 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
013572 012767 013660 004252 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
013600 012777 004000 004202 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
013606 016703 004212 MOV DHBCR,R3 ;SET UP POINTER TO BREAK CONTROL
013612 012705 177757 MOV #177757,R5 ;(R5)=EXPECTED DATA
;IN BREAK CONTROL REGISTER, 177757
013616 012713 177777 MOV #177777,(R3) ;SET ALL READ/WRITE BITS
;IN BREAK CONTROL REGISTER
013622 042713 000020 BIC #20,(R3) ;CLEAR BIT 4
013626 011304 MOV (R3),R4 ;GET CONTENTS OF BREAK CONTROL

```

```

013630 020504      .IIF NB <>,      BIC    #,R4      ;CLEAR UNWANTED BITS
013632 001401      CMP      R5,R4      ;WAS BIT 4 CLEARED
                                BEQ      1$
                                .IIF IDN    <DHBCR>,<DHLPR>,<DHLPR>,      HLT    2      ;BREAK CONTROL REGIS
                                .IIF IDN    <DHBCR>,<DHBCR>,<DHBCR>,      HLT    3      ;BREAK CONTROL REGIS
                                EMT      3
                                .IIF IDN    <DHBCR>,<DHSSR>,<DHSSR>,      HLT    4      ;BREAK CONTROL REGIS
TER ERROR
013634 013634 104003      1$:      BIS      #20,(R3)      ;SET BIT 4
                                MOV      (R3),R4
                                .IIF NB <>,      BIC    #,R4      ;CLEAR UNWANTED BITS
                                CMP      #177777,R4      ;WAS BIT 4 SET
                                BEQ      2$
                                MOV      #177777,R5      ;(R5)=EXPECTED DATA IN
                                ;BREAK CONTROL REGISTER, 177777
                                .IIF IDN    <DHBCR>,<DHLPR>,<DHLPR>,      HLT    2      ;BREAK CONTROL REGIS
                                .IIF IDN    <DHBCR>,<DHBCR>,<DHBCR>,      HLT    3      ;BREAK CONTROL REGIS
                                EMT      3
                                .IIF IDN    <DHBCR>,<DHSSR>,<DHSSR>,      HLT    4      ;BREAK CONTROL REGIS
TER ERROR
013656 013656 104003
TER ERROR
013652 012705 177777
                                2$:      SCOPE
                                BITX=BITX+BITX
                                BITC=BITC+1
                                CLRBIT=-BITX-1
                                BITCLR=CLRBIT
                                REGTS7  +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC,177777,\BITCLR
                                ;BREAK CONTROL REGISTER DATA TEST
                                ;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 15
                                ;CLEAR BIT 5
                                ;VERIFY THAT BIT 5 WAS CLEARED
                                ;RESTORE BIT 5
                                ;VERIFY THAT BIT 5 WAS SET
013662 013662 104400
                                000040
                                000005
                                177737
                                177737
                                013662
013662 012767 000340 164106      TS \XN,4000,2$
013670 012767 004000 004160      T132:  MOV      #340,PS      ;DISABLE ALL INTERRUPTS
013676 012767 01376: 004146      MOV      #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
                                MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
013704 012777 004000 004076      MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
013712 016703 004106      MOV      DHBCR,R3      ;SET UP POINTER TO BREAK CONTROL
013716 012705 177737      MOV      #177737,R5      ;(R5)=EXPECTED DATA
                                ;IN BREAK CONTROL REGISTER, 177737
013722 012713 177777      MOV      #177777,(R3)      ;SET ALL READ/WRITE BITS
                                ;IN BREAK CONTROL REGISTER
013726 042713 000040      BIC      #40,(R3)      ;CLEAR BIT 5
013732 011304      MOV      (R3),R4      ;GET CONTENTS OF BREAK CONTROL
                                .IIF NB <>,      BIC    #,R4      ;CLEAR UNWANTED BITS
                                CMP      R5,R4      ;WAS BIT 5 CLEARED
                                BEQ      1$
                                .IIF IDN    <DHBCR>,<DHLPR>,<DHLPR>,      HLT    2      ;BREAK CONTROL REGIS
                                .IIF IDN    <DHBCR>,<DHBCR>,<DHBCR>,      HLT    3      ;BREAK CONTROL REGIS
                                EMT      3
                                .IIF IDN    <DHBCR>,<DHSSR>,<DHSSR>,      HLT    4      ;BREAK CONTROL REGIS
TER ERROR
013740 013740 104003
TER ERROR
013742 052713 000040      1$:      BIS      #40,(R3)      ;SET BIT 5
013746 011304      MOV      (R3),R4

```

```

013750 022704 177777      .IIF NB <>,      BIC      #,R4      ;CLEAR UNWANTED BITS
013754 001403              CMP      #177777,R4 ;WAS BIT 5 SET
013756 012705 177777      BEQ      2$
                                MOV      #177777,R5      ;(R5)=EXPECTED DATA IN
                                ;BREAK CONTROL REGISTER, 177777
                                HLT      2
TER ERROR 013762          .IIF IDN      <DHBCR>,<DHLPR>, ;BREAK CONTROL REGIS
TER ERROR 013762 104003    .IIF IDN      <DHBCR>,<DHBCR>, HLT      3      ;BREAK CONTROL REGIS
                                EMT      3
TER ERROR 013764 104400    .IIF IDN      <DHBCR>,<DHSSR>, HLT      4      ;BREAK CONTROL REGIS
                                2$: SCOPE
                                BITX=BITX+BITX
                                BITC=BITC+1
                                CLRBIT=-BITX-1
                                BITCLR=CLRBIT
                                REGTS7 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC,177777,\BITCLR
                                ;BREAK CONTROL REGISTER DATA TEST
                                ;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 15
                                ;CLEAR BIT 6
                                ;VERIFY THAT BIT 6 WAS CLEARED
                                ;RESTORE BIT 6
                                ;VERIFY THAT BIT 6 WAS SET

013766 012767 000340 164002 TS \XN,4000,2$
013766 012767 000340 164002 T133: MOV      #340,PS      ;DISABLE ALL INTERRUPTS
013774 012767 004000 004054      MOV      #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
014002 012767 014070 004042      MOV      #2$,ESCAPE   ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
014010 012777 004000 003772      MOV      #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
014016 016703 004002              MOV      DHBCR,R3      ;SET UP POINTER TO BREAK CONTROL
014022 012705 177677              MOV      #177677,R5    ;(R5)=EXPECTED DATA
                                ;IN BREAK CONTROL REGISTER, 177677
014026 012713 177777              MOV      #177777,(R3)  ;SET ALL READ/WRITE BITS
                                ;IN BREAK CONTROL REGISTER
014032 042713 000100              BIC      #100,(R3)    ;CLEAR BIT 6
014036 011304              MOV      (R3),R4      ;GET CONTENTS OF BREAK CONTROL
                                .IIF NB <>,      BIC      #,R4      ;CLEAR UNWANTED BITS
014040 020504              CMP      R5,R4        ;WAS BIT 6 CLEARED
014042 001401              BEQ      1$
                                .IIF IDN      <DHBCR>,<DHLPR>, HLT      2      ;BREAK CONTROL REGIS
TER ERROR 014044          .IIF IDN      <DHBCR>,<DHBCR>, HLT      3      ;BREAK CONTROL REGIS
TER ERROR 014044 104003    .IIF IDN      <DHBCR>,<DHSSR>, HLT      4      ;BREAK CONTROL REGIS
                                EMT      3
TER ERROR 014046 052713 000100    1$: BIS      #100,(R3)  ;SET BIT 6
014052 011304              MOV      (R3),R4
                                .IIF NB <>,      BIC      #,R4      ;CLEAR UNWANTED BITS
014054 022704 177777              CMP      #177777,R4  ;WAS BIT 6 SET
014060 001403              BEQ      2$
014062 012705 177777              MOV      #177777,R5  ;(R5)=EXPECTED DATA IN
                                ;BREAK CONTROL REGISTER, 177777
                                HLT      2
TER ERROR 014066          .IIF IDN      <DHBCR>,<DHLPR>, HLT      2      ;BREAK CONTROL REGIS
TER ERROR 014066          .IIF IDN      <DHBCR>,<DHBCR>, HLT      3      ;BREAK CONTROL REGIS
TER ERROR 014066 104003    .IIF IDN      <DHBCR>,<DHSSR>, HLT      4      ;BREAK CONTROL REGIS
                                EMT      3

```

TER ERROR

```

014070 104400      2$: SCOPE
          000200    BITX=BITX+BITX
          000007    BITC=BITC+1
          177577    CLRBIT=-BITX-1
          177577    BITCLR=CLRBIT
014072                                REGTS7 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC,177777,\BITCLR
    
```

```

;BREAK CONTROL REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 1S
;CLEAR BIT 7
;VERIFY THAT BIT 7 WAS CLEARED
;RESTORE BIT 7
;VERIFY THAT BIT 7 WAS SET
    
```

```

014072                                TS \XN,4000,2$
014072 012767 000340 163676 T134: MOV #340,PS ;DISABLE ALL INTERRUPTS
014100 012767 004000 003750 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
014106 012767 014174 003736 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC
                                XN=XN+1
014114 012777 004000 003666 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
014122 016703 003676 MOV DHBCR,R3 ;SET UP POINTER TO BREAK CONTROL
014126 012705 177577 MOV #177577,R5 ;(R5)=EXPECTED DATA
                                MOV #177777,(R3) ;IN BREAK CONTROL REGISTER, 177577
                                MOV #200,(R3) ;SET ALL READ/WRITE BITS
                                MOV (R3),R4 ;IN BREAK CONTROL REGISTER
                                .IIF NB <>, BIC #,R4 ;CLEAR BIT 7
                                CMP R5,R4 ;GET CONTENTS OF BREAK CONTROL
                                BEQ 1$ ;CLEAR UNWANTED BITS
                                .IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;WAS BIT 7 CLEARED ;BREAK CONTROL REGIS
                                .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
                                EMT 3
                                .IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
014152 052713 000200 1$: BIS #200,(R3) ;SET BIT 7
014156 011304 MOV (R3),R4
                                .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
                                CMP #177777,R4 ;WAS BIT 7 SET
                                BEQ 2$
                                MOV #177777,R5 ;(R5)=EXPECTED DATA IN
                                .IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGISTER, 177777 ;BREAK CONTROL REGIS
                                .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
                                EMT 3
                                .IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
014174 104400      2$: SCOPE
          000400    BITX=BITX+BITX
          000010    BITC=BITC+1
          177377    CLRBIT=-BITX-1
          177377    BITCLR=CLRBIT
014176                                REGTS7 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC,177777,\BITCLR
    
```

```

;BREAK CONTROL REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 1S
    
```



;CLEAR BIT 10  
 ;VERIFY THAT BIT 10 WAS CLEARED  
 ;RESTORE BIT 10  
 ;VERIFY THAT BIT 10 WAS SET

014176				TS \XN,4000,2#		
014176	012767	000340	163572	T135: MOV	#340,PS	;DISABLE ALL INTERRUPTS
014204	012767	004000	003644	MOV	#4000,ICOUNT	;SET UP FOR 4000 ITERATIONS
014212	012767	014300	003632	MOV	#2#,ESCAPE	;SET UP TO ESCAPE TO NEXT TEST
				.IF NB	<>	
				MOV	#,FREEZ1	;SET UP TO LOOP WITH DATA ; 3
		000136		.ENDC		
				XN=XN+1		
014220	012777	004000	003562	MOV	#BIT11,@DHSCR	;MASTER CLEAR INTERFACE
014226	016703	003572		MOV	DHBCR,R3	;SET UP POINTER TO BREAK CONTROL
014232	012705	177377		MOV	#177377,R5	; (R5)=EXPECTED DATA
						;IN BREAK CONTROL REGISTER, 177377
014236	012713	177777		MOV	#177777,(R3)	;SET ALL READ/WRITE BITS
						;IN BREAK CONTROL REGISTER
014242	042713	000400		BIC	#400,(R3)	;CLEAR BIT 10
014246	011304			MOV	(R3),R4	;GET CONTENTS OF BREAK CONTROL
				.IIF NB	<>, BIC #,R4	;CLEAR UNWANTED BITS
014250	020504			CMP	R5,R4	;WAS BIT 10 CLEARED
014252	001401			BEQ	1#	
TER ERROR				.IIF IDN	<DHBCR>,<DHLPR>,	HLT 2 ;BREAK CONTROL REGIS
014254				.IIF IDN	<DHBCR>,<DHBCR>,	HLT 3 ;BREAK CONTROL REGIS
TER ERROR						
014254	104003			EMT	3	
TER ERROR				.IIF IDN	<DHBCR>,<DHSSR>,	HLT 4 ;BREAK CONTROL REGIS
014256	052713	000400		1#:	BIS #400,(R3)	;SET BIT 10
014262	011304			MOV	(R3),R4	
				.IIF NB	<>, BIC #,R4	;CLEAR UNWANTED BITS
014264	022704	177777		CMP	#177777,R4	;WAS BIT 10 SET
014270	001403			BEQ	2#	
014272	012705	177777		MOV	#177777,R5	; (R5)=EXPECTED DATA IN
						;BREAK CONTROL REGISTER, 177777
TER ERROR				.IIF IDN	<DHBCR>,<DHLPR>,	HLT 2 ;BREAK CONTROL REGIS
014276				.IIF IDN	<DHBCR>,<DHBCR>,	HLT 3 ;BREAK CONTROL REGIS
TER ERROR						
014276	104003			EMT	3	
TER ERROR				.IIF IDN	<DHBCR>,<DHSSR>,	HLT 4 ;BREAK CONTROL REGIS
014300	104400			2#:	SCOPE	
	001000				BITX=BITX+BITX	
	000011				BITC=BITC+1	
	176777				CLRBIT=-BITX-1	
	176777				BITCLR=CLRBIT	
014302				REGTS7	↑/DHBCR/, \BITX,↑/BREAK CONTROL/, \BITC,177777,\BITCLR	
						;BREAK CONTROL REGISTER DATA TEST
						;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 1S
						;CLEAR BIT 11
						;VERIFY THAT BIT 11 WAS CLEARED
						;RESTORE BIT 11
						;VERIFY THAT BIT 11 WAS SET
014302				TS \XN,4000,2#		
014302	012767	000340	163466	T136: MOV	#340,PS	;DISABLE ALL INTERRUPTS
014310	012767	004000	003540	MOV	#4000,ICOUNT	;SET UP FOR 4000 ITERATIONS
014316	012767	014404	003526	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
014324 000137 004000 003456 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
014332 016703 003466 MOV DHBCR,R3 ;SET UP POINTER TO BREAK CONTROL
014336 012705 176777 MOV #176777,R5 ;(R5)=EXPECTED DATA
014342 012713 177777 MOV #177777,(R3) ;IN BREAK CONTROL REGISTER, 176777
014346 042713 001000 BIC #1000,(R3) ;SET ALL READ/WRITE BITS
014352 011304 MOV (R3),R4 ;IN BREAK CONTROL REGISTER
;CLEAR BIT 11
014354 020504 .IIF NB <>, BIC #,R4 ;GET CONTENTS OF BREAK CONTROL
014356 001401 CMP R5,R4 ;CLEAR UNWANTED BITS
BEQ 1$ ;WAS BIT 11 CLEARED
      .IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
      .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
      EMT 3
      .IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
1$: BIS #1000,(R3) ;SET BIT 11
MOV (R3),R4
      .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
CMP #177777,R4 ;WAS BIT 11 SET
BEQ 2$
MOV #177777,R5 ;(R5)=EXPECTED DATA IN
;BREAK CONTROL REGISTER, 177777
      .IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
      .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
      EMT 3
      .IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
2$: SCOPE
BITX=BITX+BITX
BITC=BITC+1
CLRBIT=-BITX-1
BITCLR=CLRBIT
REGTS7 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC,177777,\BITCLR
;BREAK CONTROL REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 15
;CLEAR BIT 12
;VERIFY THAT BIT 12 WAS CLEARED
;RESTORE BIT 12
;VERIFY THAT BIT 12 WAS SET
014406 104400 TS \XN,4000,2$
014406 012767 000340 163362 T137: MOV #340,PS ;DISABLE ALL INTERRUPTS
014414 012767 004000 003434 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
014422 012767 014510 003422 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
014430 000140 004000 003352 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
014436 016703 003362 MOV DHBCR,R3 ;SET UP POINTER TO BREAK CONTROL
014442 012705 175777 MOV #175777,R5 ;(R5)=EXPECTED DATA
014446 012713 177777 MOV #177777,(R3) ;IN BREAK CONTROL REGISTER, 175777
;SET ALL READ/WRITE BITS

```

```

014452 042713 002000          BIC    #2000,(R3)          ;IN BREAK CONTROL REGISTER
014456 011304                MOV    (R3),R4          ;CLEAR BIT 12
                                .IIF NB <>, BIC    #,R4          ;GET CONTENTS OF BREAK CONTROL
                                CMP    R5,R4          ;CLEAR UNWANTED BITS
                                BEQ    1$            ;WAS BIT 12 CLEARED

                                .IIF IDN <DHBCR>,<DHLPR>, HLT    2                ;BREAK CONTROL REGIS
                                .IIF IDN <DHBCR>,<DHBCR>, HLT    3                ;BREAK CONTROL REGIS
                                EMT    3
                                .IIF IDN <DHBCR>,<DHSSR>, HLT    4                ;BREAK CONTROL REGIS

TER ERROR 014464
TER ERROR 014464 104003          EMT    3
                                .IIF IDN <DHBCR>,<DHSSR>, HLT    4                ;BREAK CONTROL REGIS

TER ERROR 014466 052713 002000 1$: BIC    #2000,(R3)          ;SET BIT 12
014472 011304                MOV    (R3),R4
                                .IIF NB <>, BIC    #,R4          ;CLEAR UNWANTED BITS
                                CMP    #177777,R4      ;WAS BIT 12 SET
                                BEQ    2$
                                MOV    #177777,R5      ;(R5)=EXPECTED DATA IN
                                                ;BREAK CONTROL REGISTER, 177777
                                .IIF IDN <DHBCR>,<DHLPR>, HLT    2                ;BREAK CONTROL REGIS
                                .IIF IDN <DHBCR>,<DHBCR>, HLT    3                ;BREAK CONTROL REGIS
                                EMT    3
                                .IIF IDN <DHBCR>,<DHSSR>, HLT    4                ;BREAK CONTROL REGIS

TER ERROR 014506
TER ERROR 014506 104003          EMT    3
TER ERROR 014510 104400          2$: SCOPE
                                004000 BITX=BITX+BITX
                                000013 BITC=BITC+1
                                173777 CLRBIT=-BITX-1
                                173777 BITCLR=CLRBIT
014512 REGTS7 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC,177777,\BITCLR

                                ;BREAK CONTROL REGISTER DATA TEST
                                ;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 1S
                                ;CLEAR BIT 13
                                ;VERIFY THAT BIT 13 WAS CLEARED
                                ;RESTORE BIT 13
                                ;VERIFY THAT BIT 13 WAS SET

014512 012767 000340 163256 TS \XN,4000,2$
014512 012767 004000 003330 T140: MOV    #340,PS          ;DISABLE ALL INTERRUPTS
014520 012767 004000 003330      MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
014526 012767 014614 003316      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

014534 000141 004000 003246      MOV    #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
014542 016703 003256          MOV    DHBCR,R3        ;SET UP POINTER TO BREAK CONTROL
014546 012705 173777          MOV    #173777,R5      ;(R5)=EXPECTED DATA
                                MOV    #177777,(R3)    ;IN BREAK CONTROL REGISTER, 173777
                                .IIF NB <>, BIC    #,R4          ;SET ALL READ/WRITE BITS
                                CMP    R5,R4          ;IN BREAK CONTROL REGISTER
                                BEQ    1$            ;CLEAR BIT 13
                                .IIF IDN <DHBCR>,<DHLPR>, HLT    2                ;BREAK CONTROL REGIS
                                .IIF IDN <DHBCR>,<DHBCR>, HLT    3                ;BREAK CONTROL REGIS

TER ERROR 014564 020504          .IIF NB <>, BIC    #,R4          ;GET CONTENTS OF BREAK CONTROL
014566 001401                CMP    R5,R4          ;CLEAR UNWANTED BITS
                                BEQ    1$            ;WAS BIT 13 CLEARED

                                .IIF IDN <DHBCR>,<DHLPR>, HLT    2                ;BREAK CONTROL REGIS
                                .IIF IDN <DHBCR>,<DHBCR>, HLT    3                ;BREAK CONTROL REGIS

TER ERROR 014570
TER ERROR 014570 104003          EMT    3

```

```

TER ERROR          .IIF IDN      <DHBCR>,<DHSSR> ,      HLT      4      ;BREAK CONTROL REGIS
    014572 052713 004000      1$:      BIS      #4000,(R3)      ;SET BIT 13
    014576 011304              MOV      (R3),R4
    014600 022704 177777      .IIF NB <> ,      BIC      #,R4      ;CLEAR UNWANTED BITS
    014604 001403              CMP      #177777,R4      ;WAS BIT 13 SET
    014606 012705 177777      BEQ      2$
    MOV      #177777,R5      ;(R5)=EXPECTED DATA IN
                                ;BREAK CONTROL REGISTER, 177777
TER ERROR          .IIF IDN      <DHBCR>,<DHLPR> ,      HLT      2      ;BREAK CONTROL REGIS
    014612              .IIF IDN      <DHBCR>,<DHBCR> ,      HLT      3      ;BREAK CONTROL REGIS
TER ERROR          EMT      3
    014612 104003      .IIF IDN      <DHBCR>,<DHSSR> ,      HLT      4      ;BREAK CONTROL REGIS
TER ERROR          2$:      SCOPE
    014614 104400      BITX=BITX+BITX
    010000      BITC=BITC+1
    000014      CLRBIT=-BITX-1
    167777      BITCLR=CLRBIT
    167777      REGTS7 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC,177777,\BITCLR
    014616              ;BREAK CONTROL REGISTER DATA TEST
                                ;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 15
                                ;CLEAR BIT 14
                                ;VERIFY THAT BIT 14 WAS CLEARED
                                ;RESTORE BIT 14
                                ;VERIFY THAT BIT 14 WAS SET
    014616              TS \XN,4000,2$
    014616 012767 000340 163152 T141:      MOV      #340,PS      ;DISABLE ALL INTERRUPTS
    014624 012767 004000 003224      MOV      #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
    014632 012767 014720 003212      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
    .IF NB <>
    MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA ; 3
    .ENDC
    XN=XN+1
    014640 000142              MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
    014646 012777 004000 003142      MOV      DHBCR,R3      ;SET UP POINTER TO BREAK CONTROL
    014652 012705 167777      MOV      #167777,R5      ;(R5)=EXPECTED DATA
    014656 012713 177777      MOV      #177777,(R3)      ;IN BREAK CONTROL REGISTER, 167777
    014662 042713 010000      BIC      #10000,(R3)      ;SET ALL READ/WRITE BITS
    014666 011304      MOV      (R3),R4      ;IN BREAK CONTROL REGISTER
    .IIF NB <> ,      BIC      #,R4      ;CLEAR BIT 14
    014670 020504      CMP      R5,R4      ;GET CONTENTS OF BREAK CONTROL
    014672 001401      BEQ      1$      ;CLEAR UNWANTED BITS
                                ;WAS BIT 14 CLEARED
TER ERROR          .IIF IDN      <DHBCR>,<DHLPR> ,      HLT      2      ;BREAK CONTROL REGIS
    014674              .IIF IDN      <DHBCR>,<DHBCR> ,      HLT      3      ;BREAK CONTROL REGIS
TER ERROR          EMT      3
    014674 104003      .IIF IDN      <DHBCR>,<DHSSR> ,      HLT      4      ;BREAK CONTROL REGIS
TER ERROR          1$:      BIS      #10000,(R3)      ;SET BIT 14
    014702 011304      MOV      (R3),R4
    014704 022704 177777      .IIF NB <> ,      BIC      #,R4      ;CLEAR UNWANTED BITS
    014710 001403      CMP      #177777,R4      ;WAS BIT 14 SET
    014712 012705 177777      BEQ      2$
    MOV      #177777,R5      ;(R5)=EXPECTED DATA IN
                                ;BREAK CONTROL REGISTER, 177777
TER ERROR          .IIF IDN      <DHBCR>,<DHLPR> ,      HLT      2      ;BREAK CONTROL REGIS

```

```

014716          .IIF IDN      <DHBCR>,<DHBCR>.,      HLT      3          ;BREAK CONTROL REGIS
TER ERROR
014716 104003          EMT      3
.IIF IDN      <DHBCR>,<DHSSR>.,      HLT      4          ;BREAK CONTROL REGIS
TER ERROR
014720 104400          2$:      SCOPE
020000          BITX=BITX+BITX
000015          BITC=BITC+1
157777          CLRBIT=-BITX-1
157777          BITCLR=CLRBIT
014722          REGTS7  +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC,177777,\BITCLR

;BREAK CONTROL REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 15
;CLEAR BIT 15
;VERIFY THAT BIT 15 WAS CLEARED
;RESTORE BIT 15
;VERIFY THAT BIT 15 WAS SET

014722          TS \XN,4000,2$
014722 012767 000340 163046 T142:  MOV      #340,PS          ;DISABLE ALL INTERRUPTS
014730 012767 004000 003120      MOV      #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
014736 012767 015024 003106      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
014744 000143          012777 004000 003036      MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
014752 016703 003046      MOV      DHBCR,R3          ;SET UP POINTER TO BREAK CONTROL
014756 012705 157777      MOV      #157777,R5      ;(R5)=EXPECTED DATA
;IN BREAK CONTROL REGISTER, 157777
014762 012713 177777      MOV      #177777,(R3)      ;SET ALL READ/WRITE BITS
;IN BREAK CONTROL REGISTER
014766 042713 020000      BIC      #20000,(R3)      ;CLEAR BIT 15
014772 011304          MOV      (R3),R4          ;GET CONTENTS OF BREAK CONTROL
.IIF NB      <>.,      BIC      #,R4          ;CLEAR UNWANTED BITS
CMP      R5,R4          ;WAS BIT 15 CLEARED
BEQ      1$
014774 020504          .IIF IDN      <DHBCR>,<DHLPR>.,      HLT      2          ;BREAK CONTROL REGIS
014776 001401          .IIF IDN      <DHBCR>,<DHBCR>.,      HLT      3          ;BREAK CONTROL REGIS
TER ERROR
015000          .IIF IDN      <DHBCR>,<DHSSR>.,      HLT      4          ;BREAK CONTROL REGIS
TER ERROR
015000 104003          EMT      3
.IIF IDN      <DHBCR>,<DHSSR>.,      HLT      4          ;BREAK CONTROL REGIS
TER ERROR
015002 052713 020000      1$:      BIS      #20000,(R3)      ;SET BIT 15
015006 011304          MOV      (R3),R4
.IIF NB      <>.,      BIC      #,R4          ;CLEAR UNWANTED BITS
CMP      #177777,R4      ;WAS BIT 15 SET
BEQ      2$
015010 022704 177777      MOV      #177777,R5      ;(R5)=EXPECTED DATA IN
015014 001403          .IIF IDN      <DHBCR>,<DHLPR>.,      HLT      2          ;BREAK CONTROL REGIS
015016 012705 177777      .IIF IDN      <DHBCR>,<DHBCR>.,      HLT      3          ;BREAK CONTROL REGIS
TER ERROR
015022          .IIF IDN      <DHBCR>,<DHSSR>.,      HLT      4          ;BREAK CONTROL REGIS
TER ERROR
015022 104003          EMT      3
.IIF IDN      <DHBCR>,<DHSSR>.,      HLT      4          ;BREAK CONTROL REGIS
TER ERROR
015024 104400          2$:      SCOPE
040000          BITX=BITX+BITX
000016          BITC=BITC+1
137777          CLRBIT=-BITX-1
137777          BITCLR=CLRBIT
015026          REGTS7  +/DHBCR/,\BITX,+/BREAK CONTROL/,\BITC,177777,\BITCLR

```

```

;BREAK CONTROL REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 15
;CLEAR BIT 16
;VERIFY THAT BIT 16 WAS CLEARED
;RESTORE BIT 16
;VERIFY THAT BIT 16 WAS SET
    
```

```

015026 TS \XN,4000,2$
015026 012767 000340 162742 T143: MOV #340,PS ;DISABLE ALL INTERRUPTS
015034 012767 004000 003014 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
015042 012767 015130 003002 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
015050 000144 012777 004000 002732 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
015056 016703 002742 MOV DHBCR,R3 ;SET UP POINTER TO BREAK CONTROL
015062 012705 137777 MOV #137777,R5 ;(R5)=EXPECTED DATA
;IN BREAK CONTROL REGISTER, 137777
015066 012713 177777 MOV #177777,(R3) ;SET ALL READ/WRITE BITS
;IN BREAK CONTROL REGISTER
015072 042713 040000 BIC #40000,(R3) ;CLEAR BIT 16
015076 011304 MOV (R3),R4 ;GET CONTENTS OF BREAK CONTROL
.IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
015100 020504 CMP R5,R4 ;WAS BIT 16 CLEARED
015102 001401 BEQ 1$

.IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
TER ERROR 015104 .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
TER ERROR 015104 104003 EMT 3
.IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
TER ERROR 015106 052713 040000 1$: BIS #40000,(R3) ;SET BIT 16
015112 011304 MOV (R3),R4
.IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
015114 022704 177777 CMP #177777,R4 ;WAS BIT 16 SET
015120 001403 BEQ 2$
015122 012705 177777 MOV #177777,R5 ;(R5)=EXPECTED DATA IN
;BREAK CONTROL REGISTER, 177777
.IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
TER ERROR 015126 .IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
TER ERROR 015126 104003 EMT 3
.IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
TER ERROR 015130 104400 2$: SCOPE
100000 BITX=BITX+BITX
000017 BITC=BITC+1
077777 CLRBIT=-BITX-1
077777 BITCLR=CLRBIT
015132 REGTS7 †/DHBCR/,\BITX,†/BREAK CONTROL/,\BITC,177777,\BITCLR

;BREAK CONTROL REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN BREAK CONTROL REGISTER TO 15
;CLEAR BIT 17
;VERIFY THAT BIT 17 WAS CLEARED
;RESTORE BIT 17
;VERIFY THAT BIT 17 WAS SET
    
```

```

015132 012767 000340 162636 T144: MOV #340,PS ;DISABLE ALL INTERRUPTS
015140 012767 004000 002710 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
015146 012767 015234 002676 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1
015154 012777 004000 002626 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
015162 016703 002636 MOV DHBCR,R3 ;SET UP POINTER TO BREAK CONTROL
015166 012705 077777 MOV #77777,R5 ;(R5)=EXPECTED DATA
;IN BREAK CONTROL REGISTER, 77777
015172 012713 177777 MOV #177777,(R3) ;SET ALL READ/WRITE BITS
;IN BREAK CONTROL REGISTER
015176 042713 100000 BIC #100000,(R3) ;CLEAR BIT 17
015202 011304 MOV (R3),R4 ;GET CONTENTS OF BREAK CONTROL
;CLEAR UNWANTED BITS
015204 020504 .IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
015206 001401 CMP R5,R4 ;WAS BIT 17 CLEARED
BEQ 1$
.IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
.IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
EMT 3
.IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
1$: BIS #100000,(R3) ;SET BIT 17
MOV (R3),R4
.IIF NB <>, BIC #,R4 ;CLEAR UNWANTED BITS
CMP #177777,R4 ;WAS BIT 17 SET
BEQ 2$
MOV #177777,R5 ;(R5)=EXPECTED DATA IN
;BREAK CONTROL REGISTER, 177777
.IIF IDN <DHBCR>,<DHLPR>, HLT 2 ;BREAK CONTROL REGIS
.IIF IDN <DHBCR>,<DHBCR>, HLT 3 ;BREAK CONTROL REGIS
EMT 3
.IIF IDN <DHBCR>,<DHSSR>, HLT 4 ;BREAK CONTROL REGIS
2$: SCOPE
BITX=BITX+BITX
BITC=BITC+1
XBIT=BITX
CBIT=BITC
BITX=1
BITC=0
.REPT 7
.NLIST
CCRBIT=-BITX-1
CLRBIT=CCRBIT&100077
BITCLR=CLRBIT
.LIST
REGTS7 +/DHSSR/,\BITX,+/SILO STATUS/,\BITC,100077,CLRBIT,77700
.IF EQ BITX-40
.NLIST
BITX=100000
BITC=17
.LIST
.IFF
.NLIST
BITX=BITX+BITX
BITC=BITC+1
.LIST

```

364  
365

177776  
100076  
100076

015236

```
.ENDC
.ENDR
CCRBIT=-BITX-1
CLRBIT=CCRBIT&100077
BITCLR=CLRBIT
REGTS7 +/DHSSR/,\BITX,+/SILO STATUS/,\BITC,100077,CLRBIT,77700
```

```
;SILO STATUS REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN SILO STATUS REGISTER TO 1S
;CLEAR BIT 0
;VERIFY THAT BIT 0 WAS CLEARED
;RESTORE BIT 0
;VERIFY THAT BIT 0 WAS SET
```

015236  
015236 012767 000340 162532  
015244 012767 004000 002604  
015252 012767 015350 002572

```
TS \XN,4000,2$
T145: MOV #340,PS ;DISABLE ALL INTERRUPTS
MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
.IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
```

015260 000146  
015266 012777 004000 002522  
015272 016703 002534  
015272 012705 100076

```
.ENDC
XN=XN+1
MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
MOV DHSSR,R3 ;SET UP POINTER TO SILO STATUS
MOV #CLRBIT,R5 ;(R5)=EXPECTED DATA
MOV #100077,(R3) ;IN SILO STATUS REGISTER, CLRBIT
;SET ALL READ/WRITE BITS
;IN SILO STATUS REGISTER
```

015276 012713 100077  
015302 042713 000001  
015306 011304  
015310 042704 077700  
015314 020504  
015316 001401

```
BIC #1,(R3) ;CLEAR BIT 0
MOV (R3),R4 ;GET CONTENTS OF SILO STATUS
.IIF NB <77700>, BIC #77700,R4 ;CLEAR UNWANTED BITS
CMP R5,R4 ;WAS BIT 0 CLEARED
BEQ 1$
```

R ERROR

```
.IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
```

R ERROR

```
.IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
```

R ERROR

015320

```
.IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE
```

015320 104004  
015322 052713 000001  
015326 011304  
015330 042704 077700  
015334 022704 100077  
015340 001403  
015342 012705 100077

```
EMT 4 ;SET BIT 0
1$: BIS #1,(R3)
MOV (R3),R4
.IIF NB <77700>, BIC #77700,R4 ;CLEAR UNWANTED BITS
CMP #100077,R4 ;WAS BIT 0 SET
BEQ 2$
MOV #100077,R5 ;(R5)=EXPECTED DATA IN
;SILO STATUS REGISTER, 100077
```

R ERROR

```
.IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
```

R ERROR

```
.IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
```

R ERROR

015346

```
.IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE
```

015346 104004  
015350 104400

```
EMT 4
2$: SCOPE
.IF EQ BITX-40
.NLIST
BITX=100000
BITC=17
.LIST
.IFF
BITX=BITX+BITX
BITC=BITC+1
```

000002  
000001



177775  
100075  
100075  
015352

```
.ENDC
CCRBIT=-BITX-1
CLRBIT=CCRBIT&100077
BITCLR=CLRBIT
REGTS7 +/DHSSR/,\BITX,+/SILO STATUS/,\BITC,100077,CLRBIT,77700
```

```
;SILO STATUS REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN SILO STATUS REGISTER TO 1S
;CLEAR BIT 1
;VERIFY THAT BIT 1 WAS CLEARED
;RESTORE BIT 1
;VERIFY THAT BIT 1 WAS SET
```

015352  
015352 012767 000340 162416  
015360 012767 004000 002470  
015366 012767 015464 002456

```
TS \XN,4000,2$
T146: MOV #340,PS ;DISABLE ALL INTERRUPTS
MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
.IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
```

000147  
015374 012777 004000 002406  
015402 016703 002420  
015406 012705 100075

```
.ENDC
XN=XN+1
MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
MOV DHSSR,R3 ;SET UP POINTER TO SILO STATUS
MOV #CLRBIT,R5 ;(R5)=EXPECTED DATA
MOV #100077,(R3) ;IN SILO STATUS REGISTER, CLRBIT
;SET ALL READ/WRITE BITS
;IN SILO STATUS REGISTER
```

015412 012713 100077  
015416 042713 000002  
015422 011304  
015424 042704 077700  
015430 020504  
015432 001401

```
BIC #2,(R3) ;CLEAR BIT 1
MOV (R3),R4 ;GET CONTENTS OF SILO STATUS
.IIF NB <77700>, BIC #77700,R4 ;CLEAR UNWANTED BITS
CMP R5,R4 ;WAS BIT 1 CLEARED
BEQ 1$
.IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
.IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
.IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE
```

R ERROR

R ERROR

R ERROR 015434

015434 104004  
015436 052713 000002  
015442 011304  
015444 042704 077700  
015450 022704 100077  
015454 001403  
015456 012705 100077

```
1$: EMT 4
BIS #2,(R3) ;SET BIT 1
MOV (R3),R4
.IIF NB <77700>, BIC #77700,R4 ;CLEAR UNWANTED BITS
CMP #100077,R4 ;WAS BIT 1 SET
BEQ 2$
MOV #100077,R5 ;(R5)=EXPECTED DATA IN
;SILO STATUS REGISTER, 100077
```

R ERROR

R ERROR

R ERROR 015462

015462 104004  
015464 104400

```
.IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
.IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
.IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE
```

000004  
000002

```
2$: EMT 4
SCOPE
.IF EQ BITX-40
.NLIST
BITX=100000
BITC=17
.LIST
.IFF
BITX=BITX+BITX
BITC=BITC+1
.ENDC
```

```

177773
100073
100073
015466
CCRBIT=-BITX-1
CLRBIT=CCRBIT&100077
BITCLR=CLRBIT
REGTS7 †/DHSSR/,\BITX,†/SILO STATUS/,\BITC,100077,CLRBIT,77700

;SILO STATUS REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN SILO STATUS REGISTER TO 15
;CLEAR BIT 2
;VERIFY THAT BIT 2 WAS CLEARED
;RESTORE BIT 2
;VERIFY THAT BIT 2 WAS SET

015466 012767 000340 162302 TS \XN,4000,2‡
015466 012767 004000 002354 T147: MOV †340,PS ;DISABLE ALL INTERRUPTS
015474 012767 004000 002354 MOV †4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
015502 012767 015600 002342 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
015510 000150 012777 004000 002272 MOV †BIT11,‡DHSCR ;MASTER CLEAR INTERFACE
015516 016703 002304 MOV DHSSR,R3 ;SET UP POINTER TO SILO STATUS
015522 012705 100073 MOV †CLRBIT,R5 ;(R5)=EXPECTED DATA
;IN SILO STATUS REGISTER, CLRBIT
015526 012713 100077 MOV †100077,(R3) ;SET ALL READ/WRITE BITS
;IN SILO STATUS REGISTER
015532 042713 000004 BIC †4,(R3) ;CLEAR BIT 2
015536 011304 MOV (R3),R4 ;GET CONTENTS OF SILO STATUS
015540 042704 077700 .IIF NB ‹77700>, BIC †77700,R4 ;CLEAR UNWANTED BITS
015544 020504 CMP R5,R4 ;WAS BIT 2 CLEARED
015546 001401 BEQ 1‡

.IIF IDN ‹DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
.IIF IDN ‹DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
.IIF IDN ‹DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE

015550
015550 104004 EMT 4
015552 052713 000004 1‡: BIS †4,(R3) ;SET BIT 2
015556 011304 MOV (R3),R4
015560 042704 077700 .IIF NB ‹77700>, BIC †77700,R4 ;CLEAR UNWANTED BITS
015564 022704 100077 CMP †100077,R4 ;WAS BIT 2 SET
015570 001403 BEQ 2‡
015572 012705 100077 MOV †100077,R5 ;(R5)=EXPECTED DATA IN
;SILO STATUS REGISTER, 100077
.IIF IDN ‹DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
.IIF IDN ‹DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
.IIF IDN ‹DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE

015576
015576 104004 EMT 4
015600 104400 2‡: SCOPE
.IF EQ BITX-40
.NLIST
BITX=100000
BITC=17
.LIST
.IFF
BITX=BITX+BITX
BITC=BITC+1
.ENDC
000010
000003
177767
CCRBIT=-BITX-1

```

```

100067
100067
015602 CLRBIT=CCRBIT&100077
        BITCLR=CLRBIT
        REGTS7 +/DHSSR/,\BITX,+/SILO STATUS/,\BITC,100077,CLRBIT,77700

        ;SILO STATUS REGISTER DATA TEST
        ;SET ALL READ/WRITE BITS IN SILO STATUS REGISTER TO 15
        ;CLEAR BIT 3
        ;VERIFY THAT BIT 3 WAS CLEARED
        ;RESTORE BIT 3
        ;VERIFY THAT BIT 3 WAS SET

015602 TS \XN,4000,2$
015602 012767 000340 162166 T150: MOV #340,PS ;DISABLE ALL INTERRUPTS
015610 012767 004000 002240 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
015616 012767 015714 002226 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

015624 000151 012777 004000 002156 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
015632 016703 002170 MOV DHSSR,R3 ;SET UP POINTER TO SILO STATUS
015636 012705 100067 MOV #CLRBIT,R5 ;(R5)=EXPECTED DATA
                                ;IN SILO STATUS REGISTER, CLRBIT
015642 012713 100077 MOV #100077,(R3) ;SET ALL READ/WRITE BITS
                                ;IN SILO STATUS REGISTER
015646 042713 000010 BIC #10,(R3) ;CLEAR BIT 3
015652 011304 MOV (R3),R4 ;GET CONTENTS OF SILO STATUS
015654 042704 077700 .IIF NB <77700>, BIC #77700,R4 ;CLEAR UNWANTED BITS
015660 020504 CMP R5,R4 ;WAS BIT 3 CLEARED
015662 001401 BEQ 1$

        .IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
        .IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
        .IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE

015664 104004 EMT 4
015666 052713 000010 1$: BIS #10,(R3) ;SET BIT 3
015672 011304 MOV (R3),R4
015674 042704 077700 .IIF NB <77700>, BIC #77700,R4 ;CLEAR UNWANTED BITS
015700 022704 100077 CMP #100077,R4 ;WAS BIT 3 SET
015704 001403 BEQ 2$
015706 012705 100077 MOV #100077,R5 ;(R5)=EXPECTED DATA IN
                                ;SILO STATUS REGISTER, 100077

        .IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
        .IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
        .IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE

015712 104004 EMT 4
015714 104400 2$: SCOPE
        .IF EQ BITX-40
        .NLIST
        BITX=100000
        BITC=17
        .LIST
        .IFF
        BITX=BITX+BITX
        BITC=BITC+1
        .ENDC
        CCRBIT=-BITX-1
        CLRBIT=CCRBIT&100077

000020
000004
177757
100057

```

015716 100057

BITCLR=CLRBIT  
REGTS7 †/DHSSR/,\BITX,†/SILO STATUS/,\BITC,100077,CLRBIT,77700

;SILO STATUS REGISTER DATA TEST  
;SET ALL READ/WRITE BITS IN SILO STATUS REGISTER TO 15  
;CLEAR BIT 4  
;VERIFY THAT BIT 4 WAS CLEARED  
;RESTORE BIT 4  
;VERIFY THAT BIT 4 WAS SET

```

015716 012767 000340 162052 TS \XN,4000,2$
015716 012767 004000 002124 T151: MOV #340,PS ;DISABLE ALL INTERRUPTS
015724 012767 004000 002124 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
015732 012767 016030 002112 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1
015740 012777 004000 002042 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
015746 016703 002054 MOV DHSSR,R3 ;SET UP POINTER TO SILO STATUS
015752 012705 100057 MOV #CLRBIT,R5 ;(R5)=EXPECTED DATA
;IN SILO STATUS REGISTER, CLRBIT
015756 012713 100077 MOV #100077,(R3) ;SET ALL READ/WRITE BITS
;IN SILO STATUS REGISTER
015762 042713 000020 BIC #20,(R3) ;CLEAR BIT 4
015766 011304 MOV (R3),R4 ;GET CONTENTS OF SILO STATUS
015770 042704 077700 .IIF NB <77700>, BIC #77700,R4 ;CLEAR UNWANTED BITS
015774 020504 CMP R5,R4 ;WAS BIT 4 CLEARED
015776 001401 BEQ 1$
.IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
.IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
.IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE
016000 016000 104004 EMT 4
016002 052713 000020 1$: BIS #20,(R3) ;SET BIT 4
016006 011304 MOV (R3),R4
016010 042704 077700 .IIF NB <77700>, BIC #77700,R4 ;CLEAR UNWANTED BITS
016014 022704 100077 CMP #100077,R4 ;WAS BIT 4 SET
016020 001403 BEQ 2$
016022 012705 100077 MOV #100077,R5 ;(R5)=EXPECTED DATA IN
;SILO STATUS REGISTER, 100077
R ERROR .IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
R ERROR .IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
R ERROR 016026 .IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE
016026 016026 104004 EMT 4
016030 016030 104400 2$: SCOPE
;IF EQ BITX-40
.NLIST
BITX=100000
BITC=17
.LIST
.IFF
BITX=BITX+BITX
BITC=BITC+1
.ENDC
000040 CCRBIT=-BITX-1
000005 CLRBIT=CCRBIT&100077
177737 BITCLR=CLRBIT
100037
100037
    
```

016032

REGTS7 †/DHSSR/, \BITX, †/SILO STATUS/, \BITC, 100077, CLRBIT, 77700

```

;SILO STATUS REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN SILO STATUS REGISTER TO 15
;CLEAR BIT 5
;VERIFY THAT BIT 5 WAS CLEARED
;RESTORE BIT 5
;VERIFY THAT BIT 5 WAS SET

```

```

016032 012767 000340 161736 TS \XN,4000,2$
016032 012767 004000 002010 T152: MOV #340,PS ;DISABLE ALL INTERRUPTS
016040 012767 004000 002010 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
016046 012767 016144 001776 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

016054 012777 004000 001726 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
016062 016703 001740 MOV DHSSR,R3 ;SET UP POINTER TO SILO STATUS
016066 012705 100037 MOV #CLRBIT,R5 ;(R5)=EXPECTED DATA
;IN SILO STATUS REGISTER, CLRBIT
;SET ALL READ/WRITE BITS
;IN SILO STATUS REGISTER
;CLEAR BIT 5
;GET CONTENTS OF SILO STATUS
;CLEAR UNWANTED BITS

016072 012713 100077 MOV #100077,(R3) ;WAS BIT 5 CLEARED

016076 042713 000040 BIC #40,(R3)
016102 011304 MOV (R3),R4
016104 042704 077700 .IIF NB <77700>, BIC #77700,R4
016110 020504 CMP R5,R4
016112 001401 BEQ 1$

.IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
.IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
.IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE

016114 104004 EMT 4
016116 052713 000040 1$: BIS #40,(R3) ;SET BIT 5
016122 011304 MOV (R3),R4
016124 042704 077700 .IIF NB <77700>, BIC #77700,R4 ;CLEAR UNWANTED BITS
016130 022704 100077 CMP #100077,R4 ;WAS BIT 5 SET
016134 001403 BEQ 2$
016136 012705 100077 MOV #100077,R5 ;(R5)=EXPECTED DATA IN
;SILO STATUS REGISTER, 100077

.IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
.IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
.IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE

016142 104004 EMT 4
016144 104400 2$: SCOPE
;IF EQ BITX-40
BITX=100000
BITC=17
.IFF
.NLIST
BITX=BITX+BITX
BITC=BITC+1
.LIST
.ENDC
077777 CCRBIT=-BITX-1
000077 CLRBIT=CCRBIT&100077
000077 BITCLR=CLRBIT
016146 REGTS7 †/DHSSR/, \BITX, †/SILO STATUS/, \BITC, 100077, CLRBIT, 77700

```

```

;SILO STATUS REGISTER DATA TEST
;SET ALL READ/WRITE BITS IN SILO STATUS REGISTER TO 1S
;CLEAR BIT 17
;VERIFY THAT BIT 17 WAS CLEARED
;RESTORE BIT 17
;VERIFY THAT BIT 17 WAS SET

016146          TS \XN,4000,2$
016146 012767 000340 161622 T153: MOV #340,PS ;DISABLE ALL INTERRUPTS
016154 012767 004000 001674 MOV #4000,ICOUNT ;SET UP FOR 4000 ITERATIONS
016162 012767 016260 001662 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

016170 000154 012777 004000 001612 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
016176 016703 001624 MOV DHSSR,R3 ;SET UP POINTER TO SILO STATUS
016202 012705 000077 MOV #CLRBIT,R5 ;(R5)=EXPECTED DATA
;IN SILO STATUS REGISTER, CLRBIT
016206 012713 100077 MOV #100077,(R3) ;SET ALL READ/WRITE BITS
;IN SILO STATUS REGISTER
016212 042713 100000 BIC #100000,(R3) ;CLEAR BIT 17
016216 011304 MOV (R3),R4 ;GET CONTENTS OF SILO STATUS
016220 042704 077700 .IIF NB <77700>, BIC #77700,R4 ;CLEAR UNWANTED BITS
016224 020504 CMP R5,R4 ;WAS BIT 17 CLEARED
016226 001401 BEQ 1$

;IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
;IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
;IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE

R ERROR
R ERROR
R ERROR 016230
016230 104004 EMT 4
016232 052713 100000 1$: BIS #100000,(R3) ;SET BIT 17
016236 011304 MOV (R3),R4
016240 042704 077700 .IIF NB <77700>, BIC #77700,R4 ;CLEAR UNWANTED BITS
016244 022704 100077 CMP #100077,R4 ;WAS BIT 17 SET
016250 001403 BEQ 2$
016252 012705 100077 MOV #100077,R5 ;(R5)=EXPECTED DATA IN
;SILO STATUS REGISTER, 100077

;IIF IDN <DHSSR>,<DHLPR>, HLT 2 ;SILO STATUS REGISTE
;IIF IDN <DHSSR>,<DHBCR>, HLT 3 ;SILO STATUS REGISTE
;IIF IDN <DHSSR>,<DHSSR>, HLT 4 ;SILO STATUS REGISTE

R ERROR
R ERROR
R ERROR 016256
016256 104004 EMT 4
016260 104400 2$: SCOPE

;IF EQ BITX-40
.NLIST
BITX=100000
BITC=17
.LIST
.IFF
BITX=BITX+BITX
BITC=BITC+1
.ENDC

000000
000020

356
367
368
369
370

;SYSTEM CONTROL REGISTER MOVE-BYTE TEST (PART 1)
;ISSUE A MASTER CLEAR
;MOVE A BYTE TO SET INTERRUPT ENABLE BITS
; IN UPPER BYTE OF SYSTEM CONTROL

```

```

371 ;VERIFY BITS SET CORRECTLY
372 ;MOVE A BYTE TO SET INTERRUPT ENABLE BIT
373 ; IN LOWER BYTE OF SYSTEM CONTROL
374 ;VERIFY UPPER BYTE WAS NOT AFFECTED
375
376 016262 012706 021544 T154: MOV #STACK, SP ;SET UP STACK
377 016266 012767 000340 161502 MOV #340, PS ;LOCK OUT INTERRUPTS
378 016274 012767 016404 001550 MOV #99#, ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
379 016302 012767 004000 001546 MOV #4000, ICOUNT ;SET UP FOR 4000 ITERATIONS
380 016310 052777 004000 001472 BIS #BIT11, @DHSCR ;ISSUE MASTER CLEAR
381 016316 016703 001466 MOV DHSCR, R3 ;(R3)=SYSTEM CONTROL REGISTER ADDRESS
382 016322 005203 INC R3 ; UPPER BYTE
383 016324 112713 000060 MOV# #60, (R3) ;SET INTERRUPT ENABLE BITS
384 ;IN UPPER BYTE OF SYSTEM CONTROL
385 016330 022777 030000 001452 CMP #BIT13+BIT12,@DHSCR ;VERIFY BITS SET CORRECTLY
386 016336 001406 BEQ 2# ;BRANCH IF OK
387 016340 012705 030000 MOV #BIT13+BIT12,R5 ;(R5)=REGISTER EXPECTED DATA
388 016344 017704 001440 MOV @DHSCR, R4 ;(R4)=REGISTER ACTUAL DATA
389 016350 HLT 1 ;BITS DID NOT SET CORRECTLY
016350 104001 EMT 1
390 016352 000414 BR 99# ;BRANCH TO SCOPE
391 016354 112777 000100 001426 2#: MOV# #BIT06, @DHSCR ;SET INTERRUPT ENABLE BIT
392 ;IN LOWER BYTE OF SYSTEM CONTROL
393 016362 022777 030100 001420 CMP #BIT13+BIT12+BIT06,@DHSCR ;VERIFY BITS SET CORRECTLY
394 016370 001405 BEQ 99# ;BRANCH IF OK
395 016372 012705 030100 MOV #BIT13+BIT12+BIT06,R5 ;(R5)=REGISTER EXPECTED DATA
396 016376 017704 001406 MOV @DHSCR, R4 ;(R4)=REGISTER ACTUAL DATA
397 016402 HLT 1 ;UPPER BYTE WAS AFFECTED?
016402 104001 EMT 1
398 016404 104400 99#: SCOPE ;CHECK FOR ITERATIONS,LOOP
399
400 ;SYSTEM CONTROL REGISTER MOVE-BYTE TEST (PART 2)
401 ;ISSUE MASTER CLEAR
402 ;MOVE A BYTE TO SET LINE SELECT BITS
403 ; IN LOWER BYTE OF SYSTEM CONTROL
404 ;VERIFY BITS SET CORRECTLY
405 ;MOVE A BYTE TO SET THE MAINTENANCE BIT
406 ; IN UPPER BYTE OF SYSTEM CONTROL
407 ;VERIFY LOWER BYTE WAS NOT AFFECTED
408
409 016406 012706 021544 T155: MOV #STACK, SP ;SET UP STACK
410 016412 012767 000340 161356 MOV #340, PS ;LOCK OUT INTERRUPTS
411 016420 012767 016530 001424 MOV #99#, ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
412 016426 012767 004000 001422 MOV #4000, ICOUNT ;SET UP FOR 4000 ITERATIONS
413 016434 052777 004000 001346 BIS #BIT11, @DHSCR ;ISSUE MASTER CLEAR
414 016442 016703 001342 MOV DHSCR, R3 ;(R3)=SYSTEM CONTROL REGISTER ADDRESS
415 016446 005203 INC R3 ; UPPER BYTE
416 016450 112777 000017 001332 MOV# #17, @DHSCR ;SET LINE SELECT BITS
417 ;IN LOWER BYTE OF SYSTEM CONTROL
418 016456 022777 000017 001324 CMP #17, @DHSCR ;VERIFY BITS SET CORRECTLY
419 016464 001406 BEQ 2# ;BRANCH IF OK
420 016466 012705 000017 MOV #17, R5 ;(R5)=REGISTER EXPECTED DATA
421 016472 017704 001312 MOV @DHSCR, R4 ;(R4)=REGISTER ACTUAL DATA
422 016476 HLT 1 ;BITS DID NOT SET CORRECTLY
016476 104001 EMT 1
423 016500 000413 BR 99# ;BRANCH TO SCOPE
424 016502 112713 000002 2#: MOV# #2, (R3) ;SET MAINTENANCE BIT

```

```

425
426 016506 022777 001017 001274      CMP    #1017, @DHSCR      ; IN UPPER BYTE OF SYSTEM CONTROL
427 016514 001405                      BEQ    99$                ; VERIFY BITS SET CORRECTLY
428 016516 012705 001017              MOV    #1017, R5         ; BRANCH IF OK
429 016522 017704 001262              MOV    @DHSCR, R4        ; (R5)=REGISTER EXPECTED DATA
430 016526                      HLT    1                 ; (R4)=REGISTER ACTUAL DATA
    016526 104001                      EMT    1                 ; LOWER BYTE WAS AFFECTED?
431 016530 104400                      99$:  SCOPE              ; CHECK FOR ITERATIONS, LOOP
432
433 016532      .EOP      †/BEGIN/

;END OF PASS
;TYPE NAME OF TEST
;UPDATE PASS COUNT
;CHECK FOR EXIT TO ACT-11
;RESTART TEST

016532 104401      EOP:      TYPE                ;TYPE NAME OF TEST
016534 020467      MEPASS
016536 005067 001344  CLR    LAST                ;CLEAR LAST ERROR PC
016542 005067 001274  CLR    ERRFLG             ;CLEAR ERROR FLAG
016546 005267 001272  INC    PASCNT            ;UPDATE PASS COUNT
016552 005767 162224  TST    LIGHTS           ; ARE WE USING LIGHTS?           ; 4
016556 001005      BNE    2$                ; BRANCH IF WE ARE               ; 6
016560 104401      TYPE
016562 020502      PASTXT
016564 104402      OCTASC
016566 016624      PASARG
016570 000403      BR    3$                ; PRINT PASSCOUNT               ; 4
016572                      ; CONTINUE                       ; 4
016572 016767 001246 162202 2$:  MOV    PASCNT,LIGHTS      ;DISPLAY PASS COUNT               ; 4
016600                      3$:
016600 013701 000042  MOV    @#42,R1           ;CHECK FOR ACT-11 OR DDP
016604 001405      BEQ    RESTRT          ;IF NOT, CONTINUE TESTING
016606 000005      RESET
016610 004711      LOGICAL: JSR    PC,(R1)
016612 000240      NOP
016614 000240      NOP
016616 000240      NOP
016620 000167 162462  RESTRT: JMP    BEGIN
016624 000001      PASARG: .WORD 1                ; PARAMETERS TO PRINT PASSCOUNT ; 5
016626 006 002      .BYTE 6,2
016630 020044      .WORD PASCNT
434 016632      .SCOPE
;CHECK FOR LOOP ON CURRENT TEST ; 3
;CHECK FOR ITERATION SUPPRESSION

016632 032777 002000 162140 SCOPER: BIT    @SW10,@SWR           ; 4
016640 001030      BNE    4$
016642 032777 040000 162130 1$:  BIT    @SW14,@SWR           ; 4
016650 001021      BNE    3$
016652 032777 004000 162120  BIT    @SW11,@SWR           ; 4
016660 001006      BNE    2$
016662 005267 001172  INC    LPCNT
016666 026767 001166 001162  CMP    LPCNT,ICOUNT
016674 001007      BNE    3$

```



```

016676 005067 001156      2$: CLR      LPCNT
016702 005067 001134      CLR      ERRFLG
016706 011667 001136      MOV      (SP),RETRN
016712 000002      RTI
016714 016716 001130      3$: MOV      RETRN,(SP)
016720 000002      RTI
016722 005767 001114      4$: TST      ERRFLG
016726 001745      BEQ      1$
016730 000762      BR       2$
435 016732      .SCOP1
    
```

;CHECK FOR FREEZE ON CURRENT DATA

```

016732 032777 001000 162040 SCOP1R: BIT      #SW09,@SWR
016740 001402      BEQ      1$
016742 016716 001106      MOV      FREEZ1,(SP)
016746 000002      1$: RTI
436 016750      .ERROR
    
```

;ERROR HANDLER

```

016750 032777 020000 162022 ERRORS: BIT      #SW13,@SWR
016756 001055      BNE      HALTS
016760 021667 001122      CMP      (SP),LAST
016764 001404      BEQ      1$
016766 011667 001114      MOV      (SP),LAST
016772 005067 001044      CLR      ERRFLG
016776 104406      1$: SAVOSP
017000 011605      MOV      (SP),R5
017002 162705 000002      SUB      #2,R5
017006 011504      MOV      (R5),R4
017010 006304      ASL      R4
017012 006304      ASL      R4
017014 042704 177001      BIC      #177001,R4
017020 062704 020622      ADD      #ERRTAB,R4
017024 012467 000040      MOV      (R4)+,ERRMSG
017030 011467 000052      MOV      (R4),DATABP
017034 005767 001002      TST      ERRFLG
017040 001403      BEQ      TYPMSG
017042 005767 000040      TST      DATABP
017046 001011      BNE      TYPDAT
017050 104401      TYPMSG: TYPE
017052 020377      MCRLF
017054 104402      OCTASC
017056 017154      ERTABO
017060 012767 000001 000754      MOV      #1,ERRFLG
017066 104401      TYPE
017070 000000      ERRMSG: 0
017072 005767 000010      TYPDAT: TST      DATABP
017076 001404      BEQ      RESREG
017100 104401      TYPE
017102 020377      MCRLF
017104 104402      OCTASC
017106 000000      DATABP: 0
017110 104407      RESREG: RES05
017112 005777 161662      HALTS: TST      @SWR
017116 100005      BPL      EXITER
    
```

; 4

; 4

: 3  
: 5  
: 5

: 5  
: 5

: 4

```

017120 010046          PUSHRO
017122 016600 000002  MOV      2(SP),R0
017126 000000          HALT
017130 012600          POPRO
017132 005267 000710  EXITER:  INC      ERRCNT
017136 032777 002000 161634  BIT      #SW10,@SWR
017144 001402          BEQ      1$
017146 016716 000700  MOV      ESCAPE,(SP)
017152 000002          1$: RTI
017154 000001          ERTAB0: 1
017156      006      002  .BYTE   6,2
017160 020100          SAVPC
437 017162          .TRPSRV
                        ;TRAP DISPATCH SERVICE
                        ;ARGUMENT OF TRAP IS EXTRACTED
                        ;AND USED AS OFFSET TO OBTAIN POINTER
                        ;TO SELECTED SUBROUTINE
; 3

017162 011646          TRPSRV:  MOV      (SP),-(SP)      ;GET PC OF RETURN
017164 162716 000002  SUB      #2,(SP)          ;=PC OF TRAP
017170 017616 000000  MOV      @SP,(SP)        ;GET TRP
017174 006316          TRPOK:  ASL      (SP)          ;MULTIPLY TRAP ARG BY 2
017176 042716 177001  BIC      #177001,(SP)    ;CLEAR UNWANTED BITS
017202 062716 020542  ADD      #TRPTAB,(SP)    ;POINTER TO SUBROUTINE ADDRESS
017206 017616 000000  MOV      @SP,(SP)        ;SUBROUTINE ADDRESS
017212 000136          JMP      @SP+            ;GO TO SUBROUTINE
438 017214          .TYPER
                        ;TELETYPE OUTPUT ROUTINE

017214 017605 000000  TYPER:  MOV      @SP,R5
017220 062716 000002  ADD      #2,(SP)
017224 105777 000554  1$:    TSTB     @TPCSR
017230 100375          BPL      1$
017232 105715          TSTB     (R5)
017234 001001          BNE      2$
017236 000002          RTI
017240 112577 000542  2$:    MOVB     (R5)+,@TPDBR
017244 000767          BR       1$
439 017246          .INSTRG
                        ;ASCII STRING INPUT ROUTINE

017246 017667 000000 000006  INSTRG:  MOV      @SP,MSG
017254 062716 000002  ADD      #2,(SP)
017260 104401          INSTR1:  TYPE
017262 000000          MSG:    0
017264 012704 020564  MOV      #INBUF,R4
017270 012703 000007  MOV      #7,R3
017274 105777 000500  1$:    TSTB     @TKCSR
017300 100375          BPL      1$
017302 117714 000474  MOVB     @TKDBR,(R4)
017306 142714 000200  BICB     #200,(R4)
017312 122427 000015  CMPB     (R4)+,#15
017316 001413          BEQ      INSTR2
017320 117777 000456 000460  MOVB     @TKDBR,@TPDBR
017326 105777 000452  2$:    TSTB     @TPCSR

```

017332 100375  
 017334 005303  
 017336 001356  
 017340 104401  
 017342 020373  
 017344 000745  
 017346 000002  
 440 017350

BPL 2\$  
 DEC R3  
 BNE 1\$  
 INSTRE: TYPE  
 MQM  
 BR INSTR1  
 INSTR2: RTI  
 .PARAMS

;CONVERT ASCII STRING TO OCTAL

; 3

017350 011605  
 017352 012567 000146  
 017356 012567 000144  
 017362 012567 000142  
 017366 112567 000140  
 017372 112567 000135  
 017376 010516  
 017400 005005  
 017402 012704 020564  
 017406 122714 000015  
 017412 001420  
 017414 121427 000060  
 017420 002415  
 017422 121427 000067  
 017426 003012  
 017430 142714 000060  
 017434 152405  
 017436 122714 000015  
 017442 001406  
 017444 006305  
 017446 006305  
 017450 006305  
 017452 000760  
 017454 104404  
 017456 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

; 3

017460 020567 000042  
 017464 101373  
 017466 020567 000032  
 017472 103770  
 017474 136705 000032  
 017500 001365

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

;STORE NUMBER AT SPECIFIED ADDRESS

017502 016704 000022  
 017506 010524  
 017510 062705 000002  
 017514 105367 000013  
 017520 001372  
 017522 000002  
 017524 000000  
 017526 000000  
 017530 000000

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

017532 000000  
 017533  
 441 017534

LOBITS: 0  
 ADRCNT=LOBITS+1  
 .OCTASC

;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER

017534 017601 000000  
 017540 062716 000002  
 017544 012167 000130  
 017550 112167 000126  
 017554 112167 000123  
 017560 013167 000120  
 017564 016704 000114  
 017570 116705 000106  
 017574 012700 020576  
 017600 010403  
 017602 042703 177770  
 017606 062703 000260  
 017612 110320  
 017614 006204  
 017616 006204  
 017620 006204  
 017622 005305  
 017624 001365  
 017626 012703 020610  
 017632 114023  
 017634 105367 000042  
 017640 001374  
 017642 105767 000035  
 017646 001405  
 017650 112723 000240  
 017654 105367 000023  
 017660 001373  
 017662 105013  
 017664 104401  
 017666 020610  
 017670 005367 000004  
 017674 001325  
 017676 000002  
 017700 000000  
 017702 000000  
 017703 017703  
 017704 000000  
 442 017706

OCTASN: MOV 8(SP),R1  
 ADD #2,(SP)  
 1\$: MOV (R1)+,WRDCNT  
 MOV (R1)+,CHRCNT  
 MOV (R1)+,SPACNT  
 2\$: MOV 8(R1)+,BINWRD  
 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\$  
 4\$: MOV #MDATA,R3  
 MOVB -(R0),(R3)+  
 DECB CHRCNT  
 BNE 4\$  
 TSTB SPACNT  
 BEQ 6\$  
 5\$: MOVB #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  
 .SAVREG

; 5

; 3

;SAVE PC OF TEST THAT FAILED AND RO-R5

017706 016667 000004 000164 SV05P: MOV 4(SP),SAVPC

;SAVE RO-R5

017714 010567 000154 SV05: MOV R5,SAVR5  
 017720 010467 000146 MOV R4,SAVR4  
 017724 010367 000140 MOV R3,SAVR3  
 017730 010267 000132 MOV R2,SAVR2  
 017734 010167 000124 MOV R1,SAVR1  
 017740 010067 000116 MOV R0,SAVR0

; 3

017744 000002  
443 017746

.RESREG RTI  
;RESTORE R0-R5

017746 016700 000110  
017752 016701 000106  
017756 016702 000104  
017762 016703 000102  
017766 016704 000100  
017772 016705 000076  
017776 000002

RS05: MOV SAVR0,R0  
MOV SAVR1,R1  
MOV SAVR2,R2  
MOV SAVR3,R3  
MOV SAVR4,R4  
MOV SAVR5,R5  
RTI

444 020000

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

020000 177560  
020002 177562  
020004 177564  
020006 177566

TKCSR: 177560  
TKDBR: 177562  
TPCSR: 177564  
TPDBR: 177566  
.IRP A

<DHSCR,DHNRC,DHLPR,DHBA,DHBC,DHBAR,DHBCR,DHSSR,DHSLR,DHRVEC,DHRLVL,DHTVEC,DH

TLVL>

A: 0

020010 000000  
020012 000000  
020014 000000  
020016 000000  
020020 000000  
020022 000000  
020024 000000  
020026 000000  
020030 000000  
020032 000000  
020034 000000  
020036 000000  
020040 000000

.ENDM  
DHSCR: 0  
DHNRC: 0  
DHLPR: 0  
DHBA: 0  
DHBC: 0  
DHBAR: 0  
DHBCR: 0  
DHSSR: 0  
DHSLR: 0  
DHRVEC: 0  
DHRLVL: 0  
DHTVEC: 0  
DHTLVL: 0

445 020042

.VARIA  
;PROGRAM VARIABLES

020042 000000  
020044 000000  
020046 000000  
020050 000000  
020052 000000  
020054 000000  
020056 000000  
020060 000000  
020062 000000  
020064 000000  
020066 000000  
020070 000000  
020072 000000  
020074 000000  
020076 000000  
020100 000000  
020102 000000  
020104 000000  
020106 000000

ERRFLG: 0 ;ERROR FLAG  
PASCNT: 0 ;PASS COUNT  
ERRCNT: 0 ;ERROR COUNT  
RETRN: 0 ;SCOPE RETURN ADDRESS FOR TEST LOOPING  
ESCAPE: 0 ;ADDRESS FOR ERROR ESCAPE  
FREEZ1: 0 ;DATA LOOPING RETURN ADDRESS  
ICOUNT: 0 ;ITERATION COUNT FOR TEST IN PROGRESS  
LPCNT: 0 ;NUMBER OF ITERATIONS THIS TEST  
SAVR0: 0 ;R0 SAVE AREA  
SAVR1: 0 ;R1 SAVE AREA  
SAVR2: 0 ;R2 SAVE AREA  
SAVR3: 0 ;R3 SAVE ARE  
SAVR4: 0 ;R4 SAVE AREA  
SAVR5: 0 ;R5 SAVE AREA  
SAVSP: 0 ;STACK POINTER SAVE AREA  
SAVPC: 0 ;CALLING ROUTINE SAVE AREA  
INIFLG: 0 ;PROGRAM INITIALIZATION FLAG  
STFLG: 0 ;PROGRAM START FLAG  
LAST: 0 ;LAST ERROR PC  
.IRP A <>

; 3

446 020110 A: 0  
.ENDM  
.PFAIL ;ENTER HERE ON POWER FAILURE

020110 010046 PFAIL: MOV R0,-(SP) ;SAVE R0-R5 ON PROCESSOR STACK  
020112 010146 MOV R1,-(SP)  
020114 010246 MOV R2,-(SP)  
020116 010346 MOV R3,-(SP)  
020120 010446 MOV R4,-(SP)  
020122 010546 MOV R5,-(SP)  
020124 016746 157674 MOV 24,-(SP)  
020130 010667 177742 MOV SP,SAVSP ;SAVE STACK POINTER  
020134 012767 020146 157662 MOV #RESTART,24 ;SET UP FOR POWER UP TRAP  
020142 000000 HALT ;HALT ON POWER DOWN NORMAL  
020144 000777 BR . ;3

;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

020146 016706 177724 RESTAR: MOV SAVSP,SP ;RESTORE STACK POINTER  
020152 012605 MOV (SP)+,R5 ;RESTORE R0-R5  
020154 012604 MOV (SP)+,R4  
020156 012603 MOV (SP)+,R3  
020160 012602 MOV (SP)+,R2  
020162 012601 MOV (SP)+,R1  
020164 012600 MOV (SP)+,R0  
020166 012767 020110 157630 MOV #PFAIL,24 ;SET UP FOR POWER FAILURE  
020174 012767 000340 157574 MOV #340,PS  
020202 012706 021544 MOV #STACK,SP  
020206 005067 000364 CLR TEMP  
020212 005267 000360 INC TEMP  
020216 001375 BNE .-4  
020220 104401 TYPE ;5  
020222 020377 MCRLF ;5  
020224 104402 OCTASC  
020226 020250 PFTAB  
020230 104401 TYPE  
020232 020402 MPFAIL  
020234 005067 177602 CLR ERRFLG  
020240 005067 177642 CLR LAST  
020244 000177 177600 JMP @RETRN  
020250 000001 PFTAB: 1  
020252 000006 000002 6.2  
020256 020050 RETRN

447 020260 .MSG †/DH11 STATIC LOGIC TEST/,†/CZDHA-DO/  
020260 015 012 012 MTITLE: .ASCIZ <15><12><12>/DH11 STATIC LOGIC TEST /<15><12>  
020263 104 110 061  
020266 061 040 123  
020271 124 101 124  
020274 111 103 040  
020277 114 117 107  
020302 111 103 040  
020305 124 105 123  
020310 124 040 015  
020313 012 000  
020315 015 012 126 MVECTO: .ASCIZ <15><12>/VECTOR ADDRESS-/  
012 061 123 124 040 107 040 123 015

020320	105	103	124	
020323	117	122	040	
020326	101	104	104	
020331	122	105	123	
020334	123	055	000	
020337	015	012	103	MREGAD: .ASCIZ <15><12>/CONTROL REGISTER ADDRESS- /
020342	117	116	124	
020345	122	117	114	
020350	040	122	105	
020353	107	111	123	
020356	124	105	122	
020361	040	101	104	
020364	104	122	105	
020367	123	123	055	
020372	000			
020373	040	040	077	MQM: .ASCIZ / ? /
020376	000			
020377	015	012	000	MCRLF: .ASCIZ <15><12>
020402	040	040	120	MPFAIL: .ASCIZ / POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS /
020405	117	127	105	
020410	122	040	106	
020413	101	111	114	
020416	125	122	105	
020421	054	040	120	
020424	122	117	107	
020427	122	101	115	
020432	040	122	105	
020435	123	124	101	
020440	122	124	040	
020443	101	124	040	
020446	124	105	123	
020451	124	040	111	
020454	116	040	120	
020457	122	117	107	
020462	122	105	123	
020465	123	000		
020467	015	012	103	MEPASS: .ASCIZ <15><12>/CZDHA-D0 /
020472	132	104	110	
020475	101	055	104	
020500	060	000		
020502	015	012	120	PASTXT: .ASCIZ <15><12>/PASS COUNT = /
020505	101	123	123	
020510	040	103	117	
020513	125	116	124	
020516	040	075	040	
020521	000			
020522	015	012	122	MR: .ASCIZ <15><12>/R /
020525	000			
020526	015	012	124	MTSTPC: .ASCIZ <15><12>/TEST PC - /
020531	105	123	124	
020534	040	120	103	
020537	055	000		

448 020542

.EVEN  
.TRPTAB

;TABLE OF POINTERS FOR TRAP DECODING

020542 016632  
 020544 017214  
 020546 017534  
 020550 017246  
 020552 017340  
 020554 017350  
 020556 017706  
 020560 017746  
 020562 016732  
 449 020564

TRPTAB: SCOPER  
 TYPER  
 OCTASN  
 INSTRG  
 INSTRE  
 PARAMS  
 SV05P  
 RS05  
 SCOP1R

.BUFFER

;BUFFERS FOR INPUT-OUTPUT

020564 000000  
 020576 000000  
 020610 000000  
 450 020622 020622

INBUF: 0  
 .=.+10  
 TEMP: 0  
 .=.+10  
 MDATA: 0  
 .=.+10  
 .ERRTAB

;TABLE OF POINTERS TO ERROR MESSAGES AND DATA

020622  
 451 020622 020652  
 452 020624 021320  
 453 020626 020714  
 454 020630 021326  
 455 020632 021003  
 456 020634 021326  
 457 020636 021072  
 458 020640 021326  
 459 020642 021160  
 460 020644 021326  
 461 020646 021244  
 462 020650 021326  
 463 020652 122 105 107  
 020655 111 123 124  
 020660 105 122 040  
 020663 104 111 104  
 020666 040 116 117  
 020671 124 040 122  
 020674 105 123 120  
 020677 117 116 104  
 464 020702 015 012 101  
 020705 104 104 122  
 020710 105 123 123  
 020713 000  
 455 020714 123 131 123  
 020717 124 105 115  
 020722 040 103 117  
 020725 116 124 122  
 020730 117 114 040  
 020733 122 105 107  
 020736 111 123 124

ERRTAB:

EM0  
 DT0  
 EM1  
 DT1  
 EM2  
 DT1  
 EM3  
 DT1  
 EM4  
 DT1  
 EM5  
 DT1

EM0: .ASCII /REGISTER DID NOT RESPOND/

.ASCIZ <15><12>/ADDRESS/

EM1: .ASCII /SYSTEM CONTROL REGISTER ERROR/



	020741	105	122	040			
	020744	105	122	122			
	020747	117	122				
466	020751	015	012	105	.ASCIZ	<15><12>/EXP	REC ADDRESS/
	020754	130	120	040			
	020757	040	040	040			
	020762	040	122	105			
	020765	103	040	040			
	020770	040	040	040			
	020773	101	104	104			
	020776	122	105	123			
	021001	123	000				
467	021003	114	111	116	EM2:	.ASCII	/LINE PARAMETER REGISTER ERROR/
	021006	105	040	120			
	021011	101	122	101			
	021014	115	105	124			
	021017	105	122	040			
	021022	122	105	107			
	021025	111	123	124			
	021030	105	122	040			
	021033	105	122	122			
	021036	117	122				
468	021040	015	012	105	.ASCIZ	<15><12>/EXP	REC ADDRESS/
	021043	130	120	040			
	021046	040	040	040			
	021051	040	122	105			
	021054	103	040	040			
	021057	040	040	040			
	021062	101	104	104			
	021065	122	105	123			
	021070	123	000				
469	021072	102	122	105	EM3:	.ASCII	/BREAK CONTROL REGISTER ERROR/
	021075	101	113	040			
	021100	103	117	116			
	021103	124	122	117			
	021106	114	040	122			
	021111	105	107	111			
	021114	123	124	105			
	021117	122	040	105			
	021122	122	122	117			
	021125	122					
470	021126	015	012	105	.ASCIZ	<15><12>/EXP	REC ADDRESS/
	021131	130	120	040			
	021134	040	040	040			
	021137	040	122	105			
	021142	103	040	040			
	021145	040	040	040			
	021150	101	104	104			
	021153	122	105	123			
	021156	123	000				
471	021160	123	111	114	EM4:	.ASCII	/SILO STATUS REGISTER ERROR/
	021163	117	040	123			
	021166	124	101	124			
	021171	125	123	040			
	021174	122	105	107			
	021177	111	123	124			
	021202	105	122	040			

	021205	105	122	122			
	021210	117	122				
472	021212	015	012	105	.ASCIZ	<15><12>/EXP	REC ADDRESS/
	021215	130	120	040			
	021220	040	040	040			
	021223	040	122	105			
	021226	103	040	040			
	021231	040	040	040			
	021234	101	104	104			
	021237	122	105	123			
	021242	123	000				
473	021244	115	101	123	EMS:	.ASCII	/MASTER CLEAR ERROR/
	021247	124	105	122			
	021252	040	103	114			
	021255	105	101	122			
	021260	040	105	122			
	021263	122	117	122			
474	021266	015	012	105	.ASCIZ	<15><12>/EXP	REC ADDRESS/
	021271	130	120	040			
	021274	040	040	040			
	021277	040	122	105			
	021302	103	040	040			
	021305	040	040	040			
	021310	101	104	104			
	021313	122	105	123			
	021316	123	000				
475					.EVEN		
476	021320	000001			DT0:	1	
477	021322	006	000		.BYTE	6.0	
478	021324	020074				SAVR5	
479	021326	000003			DT1:	3	
480	021330	006	002		.BYTE	6.2	
481	021332	020074				SAVR5	
482	021334	006	002		.BYTE	6.2	
483	021336	020072				SAVR4	
484	021340	006	000		.BYTE	6.0	
485	021342	020070				SAVR3	
486	021344				.ENDCOD		
	021344	000000			ENDCOD:	0	
487		000001			.END		

ADRCNT= 017533	ERTABO 017154	RETRN 020050	T104 010742	T25 003524
BEGIN 001306	ESCAPE 020052	RS05 017746	T105 011042	T26 003612
BINWRD 017704	EXITER 017132	SAVPC 020100	T106 011142	T27 003700
BITC = 000020	FREEZ1 020054	SAVRO 020062	T107 011246	T3 001570
BITCLR= 000077	HALTS 017112	SAVR1 020064	T11 002340	T30 003766
BITX = 000000	HILIM 017526	SAVR2 020066	T110 011352	T31 004042
BIT00 = 000001	ICOUNT 020056	SAVR3 020070	T111 011456	T32 004116
BIT01 = 000002	INBUF 020564	SAVR4 020072	T112 011562	T33 004172
BIT02 = 000004	INIFLG 020102	SAVR5 020074	T113 011666	T34 004246
BIT03 = 000010	INSTER= 104404	SAVSP 020076	T114 011772	T35 004322
BIT04 = 000020	INSTR = 104403	SAV05P= 104406	T115 012076	T36 004452
BIT05 = 000040	INSTRE 017340	SCOPE = 104400	T116 012202	T37 004602
BIT06 = 000100	INSTRG 017246	SCOPE1= 104410	T117 012306	T4 001664
BIT07 = 000200	INSTR1 017260	SCOPE1R 016732	T12 002420	T40 004732
BIT08 = 000400	INSTR2 017346	SCOP1R 016732	T120 012412	T41 005022
BIT09 = 001000	LAST 020106	SPACNT= 017703	T121 012516	T42 005112
BIT10 = 002000	LIGHTS 001002	STACK = 021544	T122 012622	T43 005202
BIT11 = 004000	LIMITS 017460	START 001004	T123 012726	T44 005272
BIT12 = 010000	LOBITS 017532	STFLG 020104	T124 013032	T45 005362
BIT13 = 020000	LOGICA 016610	SV05 017714	T125 013136	T46 005452
BIT14 = 040000	LOLIM 017524	SV05P 017706	T126 013242	T47 005542
BIT15 = 100000	LPCNT 020060	SWR 001000	T127 013346	T5 001760
CBIT = 000020	MCRLF 020377	SW00 = 000001	T13 002500	T50 005632
CCRBIT= 077777	MDATA 020610	SW01 = 000002	T130 013452	T51 005722
CHRCNT 017702	MEPASS 020467	SW02 = 000004	T131 013556	T52 006012
CLRBIT= 000077	MPFAIL 020402	SW03 = 000010	T132 013662	T53 006102
DATABP 017106	MQM 020373	SW04 = 000020	T133 013766	T54 006172
DEVADR 017530	MR 020522	SW05 = 000040	T134 014072	T55 006262
DHBA 020016	MREGAD 020337	SW06 = 000100	T135 014176	T56 006352
DHBAR 020022	MSG 017262	SW08 = 000400	T136 014302	T57 006442
DHBC 020020	MTITLE 020260	SW09 = 001000	T137 014406	T6 002054
DHBCR 020024	MTSTPC 020526	SW10 = 002000	T14 002560	T60 006532
DHLPR 020014	MVECTO 020315	SW11 = 004000	T140 014512	T61 006622
DHNRC 020012	N = 000001	SW12 = 010000	T141 014616	T62 006712
DHRLVL 020034	OCTASC= 104402	SW13 = 020000	T142 014722	T63 007002
DHRVEC 020032	OCTASN 017534	SW14 = 040000	T143 015026	T64 007072
DHSCR 020010	PARAM = 104405	SW15 = 100000	T144 015132	T65 007162
DHSLR 020030	PARAMS 017350	TEMP 020576	T145 015236	T66 007252
DHSSR 020026	PARAM1 017400	TKCSR 020000	T146 015352	T67 007342
DHTLVL 020040	PARERR 017454	TKDBR 020002	T147 015466	T7 002150
DHTVEC 020036	PASARG 016624	TPCSR 020004	T15 002644	T70 007432
DT0 021320	PASCNT 020044	TPDBR 020006	T150 015602	T71 007522
DT1 021326	PASTXT 020502	TRPOK 017174	T151 015716	T72 007612
EM0 020652	PFAIL 020110	TRPSRV 017162	T152 016032	T73 007702
EM1 020714	PFTAB 020250	TRPTAB 020542	T153 016146	T74 007772
EM2 021003	POPPO = 012600	TYPDAT 017072	T154 016262	T75 010062
EM3 021072	POP1SP= 005726	TYPE = 104401	T155 016406	T76 010152
EM4 021160	POP2SP= 022626	TYPER 017214	T16 002732	T77 010242
EM5 021244	PS = 177776	TYPMSG 017050	T17 003020	VEC1 001164
ENDCOD 021344	PUSHRO= 010046	T1 001400	T2 001474	VEC2 001174
EOP 016532	PUSH1S= 005746	T10 002244	T20 003106	WRDCNT 017700
ERPCNT 020046	PUSH2S= 024646	T100 010342	T21 003174	X = 000000
EFRFLG 020042	RESREG 017110	T101 010442	T22 003262	XBIT = 000000
ERRMSG 017070	RESTAR 020146	T102 010542	T23 003350	XN = 000154
ERRORS 016750	RESTRT 016620	T103 010642	T24 003436	Y = 000011
ERRTAB 020622	RES05 = 104407			

. ABS. 021346 000  
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 20736 WORDS ( 81 PAGES)  
DYNAMIC MEMORY AVAILABLE FOR 71 PAGES  
CZDHAD.BIN,CZDHAD.SEQ=CZDHAD.DOC,DHMACA.MAC,CZDHAD.P11