

.REM !

IDENTIFICATION

PRODUCT CODE: AC-8452C-MC
PRODUCT NAME: CZDHCCO DH11 TRANSMITTER/RECEIVER 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) 1985 BY DIGITAL EQUIPMENT CORPORATION

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

1
2
3
4
5
6
7

1. ABSTRACT

THE DM11 TRANSMITTER AND RECEIVER LOGIC TEST CHECKS
THE BASIC TRANSMITTER AND RECEIVER FUNCTIONS.
FUNCTIONS TESTED INCLUDE INTERRUPTS, OPERATION OF
TRANSMITTER NPS LOGIC, AND OPERATION OF RECEIVER SILO LOGIC.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

2. REQUIREMENTS
- 2.1 EQUIPMENT
 - PDP-11 FAMILY STANDARD COMPUTER WITH 4KW OF MEMORY
 - ASR-33 TELETYPE OR EQUIVALENT
 - DH11 ASYNCHRONOUS MULTIPLEXER
 - DM11 MAINTENANCE CARD INSTALLED
- 2.2 STORAGE
 - THE PROGRAM LOADS INTO 4KW OF MEMORY
3. LOADING PROCEDURE
 - THE STANDARD PROCEDURE FOR LOADING ABSOLUTE BINARY TAPES IS TO BE USED
4. STARTING PROCEDURE
 - 4.1 CONTROL SWITCH SETTINGS
 - 4.1.1 AFTER PROGRAM LOAD (INITIAL PROGRAM START)
 - ALL CONSOLE SWITCHES DOWN
 - 4.1.2 TO MODIFY DEVICE VECTOR AND CONTROL REGISTER ADDRESSES AFTER PROGRAM RESTART
 - SW00=1
 - 4.1.3 TO START PROGRAM AT SELECTED TEST AFTER PROGRAM RESTART
 - SW01=1
 - 4.2 STARTING ADDRESS
 - THE STARTING ADDRESS FOR ALL TESTS IS 000200
 - THE RESTART ADDRESS FOR ALL TESTS 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 XXXX AND WILL THEN TYPE "VECTOR ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

4.3 (CONT'D)

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

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

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

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

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

4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN

4.3.2.1 PERFORM 4.3.1.2 TO 4.3.1.5
4.3.2.2 THE PROGRAM WILL TYPE "DH11 XXXX" 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 XXXX" AND WILL THEN TYPE "TEST PC-" AND WILL WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD
4.3.4.5 TYPE IN THE ADDRESS OF THE TEST AT WHICH THE PROGRAM IS TO BE STARTED FOLLOWED BY <CARRIAGE RETURN>
4.3.4.6 THE PROGRAM WILL TYPE R TO INDICATE THAT IT HAS STARTED AND WILL START TESTING AT THE SELECTED TEST.

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

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

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

5.2 SUBROUTINE ABSTRACTS

5.2.1 TRAPCATCHER (LOCATIONS 000000-000776)

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

2
0
4
0
...
772
0
776
0

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

5.2.2 START (PROGRAM INITIALIZATION)

THIS ROUTINE INITIALIZES ALL PROGRAM FLAGS AND COUNTERS, TYPES THE PROGRAM TITLE MESSAGE, AND INPUTS THE VECTOR AND CONTROL REGISTER ADDRESSES OF THE DM11 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. THE 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 IMMEDIATLY AFTER THE 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 "CZDHC-C" AND THEN RESTART TESTING AT TEST 1 (LOCATION T1 IN THE PROGRAM).

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

5.3.2 PROGRAM START WITH SW00=1

THE PROGRAM WILL PERFORM AS DESCRIBED IN 4.3.1 AND 5.3.1

5.3.3 PROGRAM START WITH SW01=1

5.3.3.1 REFER TO SECTION 4.3.4 FOR INITIAL PROGRAM BEHAVIOR

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

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

5.3.4 PROGRAM OPERATION WITH SW15=1

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

5.3.5 PROGRAM OPERATION WITH SW13=1

SAME AS 5.3.1 EXCEPT THAT NO ERROR TYPEOUTS WILL OCCUR

5.3.6 PROGRAM OPERATION WITH SW11=1

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

5.3.7 PROGRAM OPERATION WITH SW10=1

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

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 TYPE 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

6.2.3 ILLEGAL INTERRUPTS

IF AN INTERRUPT OCCURS TO A VECTOR ADDRESS NOT
SELECTED DURING PROGRAM INITIALIZATION, THE PROGRAM WILL
HALT IN THE TRAPCATCHER. THE ADDRESS AT WHICH
THE PROGRAM HALTS IS 2 GREATER THAN THE ADDRESS
TO WHICH THE INTERRUPT OCCURED. THE PROGRAM MUST BE
RESTARTED AT 200 TO RECOVER FROM THIS ERROR.

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

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)

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 CZDHC-C TO END OF TYPEOUT OF CZDHC-C)
IS GIVEN FOR VARIOUS PROCESSORS IN THE TABLE BELOW

PROCESSOR	TIME
PDP-11/05,10	
PDP-11/20	
PDP-11/40	
PDP-11/45	

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

9. PROGRAM DESCRIPTION

THE FIRST GROUP OF TESTS VERIFIES THAT NO INTERRUPTS OCCUR WITH INTERRUPT ENABLES SET FOR EACH INTERRUPTING FUNCTION AND THE ASSOCIATED DONE BIT OR FLAG FOR THAT FUNCTION CLEARED.

THE NEXT GROUP OF TESTS VERIFIES THAT AN INTERRUPT DOES OCCUR IF A SPECIFIC DONE BIT OR FLAG IS SET, ALONG WITH ITS CORRESPONDING INTERRUPT ENABLE. AT THIS TIME, INTERRUPTS TO THE CORRECT VECTOR ARE TESTED. IF AN INTERRUPT OCCURS TO ANY VECTOR OTHER THAN THOSE SELECTED AT PROGRAM START, THE PROGRAM WILL HALT IN THE TRAPCATCHER IN A LOCATION 2 GREATER THAN THE ADDRESS OF THE VECTOR TO WHICH THE INTERRUPT OCCURED.

THE NEXT GROUP OF TESTS CHECKS TRANSMITTER NPR AND INTERRUPT LOGIC OPERATION FOR EACH LINE, IN AN INDIVIDUAL TEST FOR EACH LINE. THE BYTE COUNT FOR THE SELECTED LINE IS SET TO -1 (FOR 1 CHARACTER TRANSMISSION) AND THE BUS ADDRESS MEMORY LOCATION FOR THAT LIN IS SET TO 0. THE BAR BIT FOR THE SELECTED LINE IS SET, TRANSMITTER INTERRUPT ENABLE IS SET, AND THE PROCESSOR STATUS WORD IS CLEARED TO ALLOW INTERRUPTS TO OCCUR. A DELAY LOOP IS THEN ENTERED, AND IF THE DELAY TIMES OUT BEFORE AN INTERRUPT OCCURS, AN ERROR MESSAGE IS TYPED. IF AN INTERRUPT DOES OCCUR, THE CONTROL REGISTER IS TESTED TO SEE IF THE TRANSMITTER DONE BIT HAS BEEN SET, THE BAR REGISTER IS CHECKED TO SEE THAT THE BAR BIT FOR THE SELECTED LIN HAS CLEARED, BYTE COUNT IS CHECKED TO SEE THAT IT WENT TO 0 AND THE BUS ADDRESS REGISTER FOR THE SELECTED LINE IS CHECKED TO SEE THAT IT INCREMENTED TO 1.

THE NEXT GROUP OF TESTS VERIFIES THAT A SINGLE SELECTED LINE WILL PERFORM TRANSMIT FUNCTIONS WITHOUT AFFECTING ANY OTHER LINE. THIS IS DONE BY SETTING ALL BYTE COUNTS TO -1, AND VERIFYING THAT ONLY THE LINE SELECTED FOR TRANSMISSION HAD CHANGES MADE IN BYTE COUNT AND BUS ADDRESS WHEN TRANSMISSION HAS BEEN COMPLETED.

THE NEXT TEST VERIFIES THAT A SINGLE CHARACTER CAN BE LOADED INTO THE SILO (IN MAINTENANCE MODE). THE TEST IS MADE WITH INTERRUPTS ENABLED AND CHECKS ARE MADE TO DETERMINE IF RECEIVER DONE WAS SET, IF THE SILO FILL LEVEL REGISTER WAS INCREMENTED, AND IF THE DATA RECEIVED IN MAINTENANCE MODE WAS CORRECT.

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

9. (CONT'D)

THE NEXT TEST VERIFIES THAT FROM 1 TO 63 CHARACTERS CAN BE LOADED INTO THE SILO, AND THAT THE SILO FILL LEVEL REGISTER INDICATES THE CORRECT NUMBER OF CHARACTERS IN THE SILO.

THE NEXT TEST VERIFIES THAT 64 CHARACTERS CAN BE LOADED INTO THE SILO, FROM 1 TO 64 CHARACTERS CAN BE READ OUT OF THE SILO, AND THAT THE SILO FILL REGISTER INDICATES THE NUMBER OF CHARACTERS REMAINING IN THE SILO.

THE FINAL TEST VERIFIES THAT THE CHARACTER AVAILABLE FLAG WILL NOT BE SET UNTIL THE SILO FILL LEVEL EXCEEDS THE SILO ALARM LEVEL, FOR ALL ALARM LEVELS 0-63.

10. LISTING

!

1 ; DMMAC-A - DM11 MACRO LIBRARY
2 ; COPYRIGHT 1985, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
3
4

5 .LIST ME
6 .MLIST MC,MD,CMD
7

104

119

131

148

158

167

303

339

373

520

563

595

607

652

664

691

712

743

744

745

746

747

748

TRAPS."

749

750

CLEAN END OF PASS
MESSAGE.

751

752

753

TCH REGISTER"

754

; CMS REPLACEMENT HISTORY

; #9 SKONETSKI 26-APR-1985 16:23:08 "FIXED TYPO CAUSING ASSEMBLY ERRORS"
; #8 SKONETSKI 22-APR-1985 16:48:03 "TYPO ERROR IN VECTOR CHANGE CODE SOURCE FIXED"
; #7 SKONETSKI 22-APR-1985 16:26:04 "ADDED CODE TO SET VECTORS FOR PWR FAIL, ERRORS, AND EMT
; #6 SKONETSKI 22-APR-1985 14:22:35 "FIXED BRANCH ERROR IN END OF PASS ROUTINE"
; #5 SKONETSKI 22-APR-1985 08:28:54 "FIXED BUG (AN OCTASC MACRO CALL WAS WRONG) AND ADDED A
; #4 SKONETSKI 18-APR-1985 14:20:15 "ADDED SOFTWARE SWITCH REG SUPPORT, BUT UNTESTED"
; #3 SKONETSKI 12-APR-1985 10:34:52 "FIXED PROBLEMS WITH SPURIOUS CR/LFS"
; #2 SKONETSKI 11-APR-1985 16:00:24 "ADDED MACRO FROM SYSMAC.SML THAT SIZES FOR SOFTWARE SWI
; #1 SKONETSKI 11-APR-1985 15:49:05 "LIBRARY FOR DM11 DIAGNOSTICS"

2
3
5 000000

.LIST ME
.NLIST MC,MD,CND
.HEADER †/1972,1976,1985/,†/DH11 TRANSMITTER AND RECEIVER BASIC LOGIC TEST/,†/CZDHC-CO/

;STARTING PROCEDURE
;LOAD PROGRAM
;LOAD ADDRESS 000200
;PRESS START
;PROGRAM WILL TYPE DH11 TRANSMITTER AND RECEIVER BASIC 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> : 3
;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
;AT THE END OF A PASS, PROGRAM WILL TYPE " CZDHC-CO "
;AND THEN RESUM TESTING

000000

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

6 000000

;SWITCH REGISTER OPTIONS

100000	SW15=100000	;=1,HALT ON ERROR	
040000	SW14=40000	;=1,LOOP ON CURRENT TEST	
020000	SW13=20000	;=1,INHIBIT ERROR TYPEOUT	
010000	SW12=10000		
004000	SW11=4000	;=1,INHIBIT ITERATIONS	
002000	SW10=2000	;=1,ESCAPE TO NEXT TEST ON ERROR	: 3
001000	SW09=1000	;=1,LOOP WITH CURRENT DATA	
000400	SW08=400		
000100	SW06=100		
000040	SW05=40		
000020	SW04=20		
000010	SW03=10		
000004	SW02=4		
000002	SW01=2		
000001	SW00=1	;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

			; 3
	;SWR=177570	;CONSOLE SWITCH REGISTER	; 4
	;LIGHTS=177570	;PDP-11/45 DISPLAY REGISTER	; 4
177776	PS=177776	;PROCESSOR STATUS WORD	; 4
017270	STACK=ENDCOD+200	;START OF PROCESSOR STACK	; 3

;INSTRUCTION DEFINITIONS

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

```

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

```

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

```

0
000000 000000 ;TRAPCATCAER FOR ILLEGAL INTERRUPTS
000200 .=U
.REPT 200
    .+2 ;UNEXPECTED TRAP TO THIS LOCATION
    HALT ;EXAMINE STACK TO FIND CAUSE
.ENDR
    .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000002 HALT ;EXAMINE STACK TO FIND CAUSE
000004 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000006 HALT ;EXAMINE STACK TO FIND CAUSE
000010 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000012 HALT ;EXAMINE STACK TO FIND CAUSE
000014 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000016 HALT ;EXAMINE STACK TO FIND CAUSE
000020 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000022 HALT ;EXAMINE STACK TO FIND CAUSE
000024 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000026 HALT ;EXAMINE STACK TO FIND CAUSE
000030 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000032 HALT ;EXAMINE STACK TO FIND CAUSE
000034 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000036 HALT ;EXAMINE STACK TO FIND CAUSE
000040 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000042 HALT ;EXAMINE STACK TO FIND CAUSE
000044 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000046 HALT ;EXAMINE STACK TO FIND CAUSE
000050 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000052 HALT ;EXAMINE STACK TO FIND CAUSE
000054 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000056 HALT ;EXAMINE STACK TO FIND CAUSE
000060 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000062 HALT ;EXAMINE STACK TO FIND CAUSE
000064 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000066 HALT ;EXAMINE STACK TO FIND CAUSE
000070 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000072 HALT ;EXAMINE STACK TO FIND CAUSE
000074 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000076 HALT ;EXAMINE STACK TO FIND CAUSE
000100 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000102 HALT ;EXAMINE STACK TO FIND CAUSE
000104 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000106 HALT ;EXAMINE STACK TO FIND CAUSE
000110 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000112 HALT ;EXAMINE STACK TO FIND CAUSE
000114 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000116 HALT ;EXAMINE STACK TO FIND CAUSE
000120 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000122 HALT ;EXAMINE STACK TO FIND CAUSE
000124 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000126 HALT ;EXAMINE STACK TO FIND CAUSE
000130 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000132 HALT ;EXAMINE STACK TO FIND CAUSE
000134 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000136 HALT ;EXAMINE STACK TO FIND CAUSE
000140 .+2 ;UNEXPECTED TRAP TO THIS LOCATION
000142 HALT ;EXAMINE STACK TO FIND CAUSE
000144 .+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
000152	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 0C0204    .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

0C0204    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,0,17776,7,10,1,1

```

```

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 <1>, ;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 017270 MOV #STACK,SP ;SET UP PROCESSOR STACK
001016 012702 000024 MOV #24,R2 ; POINT TO VECTOR AREA ; 7
001022 012722 015616 MOV #PFFAIL,(R2)+ ;SET UP POWER FAIL TRAP ; 7
001026 012722 000340 MOV #340,(R2)+ ;SERVICE AT LEVEL 7 ; 7
001032 012722 014372 MOV #ERRORS,(R2)+ ;ERROR HANDLER ; 7
001036 012722 000340 MOV #340,(R2)+ ;SERVICE AT LEVEL 7 ; 7
001042 012722 014604 MOV #TRPSRV,(R2)+ ;GENERAL HANDLER DISPATCH SERVICE ; 7
001046 012712 000340 MOV #340,(R2) ;SERVICE AT LEVEL 7 ; 8
001052 005067 014532 CLR STFLG ;CLEAR TEST START FLAG
001056 005067 014466 CLR PASCNT ;CLEAR PASS COUNT
001062 005067 014464 CLR ERRCNT ;CLEAR ERROR COUNT
001066 005067 014454 CLR ERRFLG ;CLEAR ERROR FLAG
001072 005067 014450 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 ; ; ; 4
001122 012767 000176 177650 1#: 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 ; ; ; 5
001134 005726 2#: 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 015766 TYPE #TITLE ;TYPE TITLE MESSAGE
001154 005767 014426 TST INIFLG ;CHECK INITIALIZATION FLAG
                                .IF NB <DHRVEC>
001160 001021 BNE VEC1 ;IF NOT 0, CHECK SWITCHES
                                ;FOR REINITIALIZATION
                                .IFF
                                BNE BEGIN ;IF NOT 0, START TEST
                                .ENDC
                                .IF NB <1>
001162 005000 SIZE: CLR R0
001164 012737 001176 000004 MOV #2#,R0 ;SET UP TIME OUT RETURN
001172 005720 1#: TST (R0)+ ;WILL TRAP WHEN NO MEMORY ; 9
001174 000776 BR 1# ;LOCATION RESPONDED, CONTINUE
001176 010067 014412 2#: MOV R0,HCORE ;R0 CONTAINS ADDRESS OF
001202 162767 000002 014404 SUB #2,HCORE ;NON EXISTANT MEMORY ; 9
001210 012737 000006 000004 MOV #6#,R0 ;RESTORE TRAPCATCHER

```

```

.ENDC
.IF NB
TRACER: MOV    #1$,#0#10           ;SET UP ILLEGAL INSTRUCTION TRAP RETURN
        SXT    RO                 ;DO 11/40, 11/45 INSTRUCTION
        MOV    #RTT,TRTRET        ;11/40,45 RTT RETURN FROM TRACE TRAP
        BR     2$
1$:     MOV    #RTI,TRTRET        ;1105,10,20 RTI RETURN FROM TRACE TRAP
        MOV    #12,#0#10         ;RESTORE TRAPCATCHER
        MOV    #TRTRET,#0#16     ;SET UP TRACE TRAP VECTOR

.ENDC
.IF NB <DHRVEC>
.IF B <1>
        BR     VEC2
.IFF
001216 005767 014364
001222 001404
        TST    INIFLG             ;IF INITIALIZE FLAG=0
        BEQ    VEC2             ;GET VECTOR AND CSR ADDRESS

.ENDC
VEC1:   BIT    #SW00,#SWR        ;IF SW00=1, GET NEW VECTOR
        BEQ    BEGIN           ;AND CSR
VEC2:
001234 032777 000001 177546
001232 001445
001234 012701 000300
001240 012702 000302
001244 012703 000004
001250 010211
001252 005012
001254 060301
001256 060302
001260 020127 001000
001264 001371
001266 104403
001270 016045
001272 104405
001274 000300
001276 000770
001300 015536
001302 003
001303 004
001304 104403
001306 016067
001310 104405
001312 000000
001314 177776
001316 015514
001320 007
001321 010
        MOV    #300,R1
        MOV    #302,R2
        MOV    #4,R3
1$:     MOV    R2,(R1)           ;RESTORE TRAPCATCHER
        CLR    (R2)            ;IN FLOATING VECTOR APEA
        ADD    R3,R1
        ADD    R3,R2
        CMP    R1,#1000
        BNE    1$
        INSTR MVECTOR         ;INPUT ADDRESS OF DEVICE VECTOR
        PARAM 300             ;MESSAGE "VECTOR ADDRESS-"
        PARAM 770             ;CONVERT STRING TO OCTAL
        DHRVEC                ;LOW LIMIT
        3                     ;HIGH LIMIT
        4                     ;LOCATIONS TO BE FILLED
        INSTR MREGAD          ;NUMBER OF LOCATIONS
        PARAM 0               ;LSB MASK
        177776                ;INPUT ADDRESS OF DEVICE CSR
        DHSCR                 ;MESSAGE "CONTROL REGISTER ADDRESS-"
        7                     ;CONVERT STRING TO OCTAL
        10                    ;LOW LIMIT
        7                     ;HIGH LIMIT
        10                    ;LOCATIONS TO BE FILLED
        10                    ;NUMBER OF LOCATIONS
        10                    ;LSB MASK

.ENDC
.IF NB <1>
CODEM1
001322 016767 014204 014204
001330 005267 014200
        MOV    DHSSR,DHSLR      ;SET UP ADDRESS OF SILO
        INC    DHSLR           ;STATUS REGISTER HIGH BYTE

.ENDC
001334 005767 014246
001340 001002
001342 005167 014240
        TST    INIFLG             ;IF INITIALIZATION FLAG
        BNE    BEGIN           ;IS CLEARED
        COM    INIFLG          ;SET IT

;PROGRAM START
;CHECK FOR PROGRAM START AT SELECTED ADDRESS

```

```

001346 012767 000340 176422 BEGIN: MOV #340,PS ;LOCK OUT INTERRUPTS
001354 012706 017270 MOV #STACK,SP ;SET UP PROCESSOR STACK
001360 032777 000002 177412 BIT #SW01,@SWR ;IF SW01=1 ; 4
001366 001410 BEQ 1# ;GET PC FOR PROGRAM START
001370 104403 INSTR ;GET PC
001372 016256 MTSTPC ;MESSAGE "TEST PC"
001374 104405 PARAM ;CONVERT STRING TO OCTAL
001376 000000 0
001400 017500 17500
001402 015554 RETRN
001404 001 .BYTE 1
001405 001 .BYTE 1
001406 000410 BR 2#
001410 012767 001440 014136 1#: MOV #T1,RETRN ;NORMAL START, TEST 1
001416 005767 014166 TST STFLG ;IF LOOPING, BYPASS TYPEOUT
001422 001004 BNE 3#
001424 005167 014160 COM STFLG
001430 104401 016252 2#: TYPE ,MR ;TYPE "R" TO INDICATE START
001434 000177 014114 3#: JMP @RETRN ;START TESTING ; 3

```



```

001440          INTST1  ↑/CHARACTER AVAILABLE/,BIT06
                   ;INTERRUPT LOGIC TEST
                   ;SET CHARACTER AVAILABLE INTERRUPT ENABLE
                   ;VERIFY THAT NO INTERRUPTS OCCUR

001440          TS \XN,4000,3#
001440 012767 000340 176330 T1:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
001446 012767 004000 014106      MOV    #4000,ICOUNT       ;SET UP FOR 4000 ITERATIONS
001454 012767 001574 014074      MOV    #3#,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST

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

001462 000002          MOV    #BIT11,@DHSCR       ;MASTER CLEAR INTERFACE
001470 012777 004000 014024      MOV    #1#,@DHRVEC       ;SET UP FOR POSSIBLE
                                ;RECEIVER INTERRUPT

001476 012777 000340 014034      MOV    #340,@DHRLVL      ;
001504 012777 001560 014030      MOV    #2#,@DHTVEC       ;SET UP FOR POSSIBLE
                                ;TRANSMITTER INTERRUPT

001512 012777 000340 014024      MOV    #340,@DHTLVL     ;
001520 012777 000100 013766      MOV    #BIT06,@DHSCR    ;SET CHARACTER AVAILABLE
                                ;INTERRUPT ENABLE
                                ;ALLOW INTERRUPTS
                                ;WINDOW FOR INTERRUPTS
                                ;NO INTERRUPT OCCURED
                                ;CONTINUE
001526 005067 176244          CLR    PS                ;GET CONTENTS OF SYSTEM CONTROL REGISTER
001532 000240          NOP                                ;UNEXPECTED INTERRUPT
001534 012767 000340 176234      MOV    #340,PS          ;SET UP TO
001542 000414          BR    3#                                ;RETURN FROM INTERRUPT
001544 017705 013744          1#:  MOV    @DHSCR,R5        ;GET CONTENTS OF SYSTEM CONTROL REGISTER
001550          HLT    0                                ;UNEXPECTED INTERRUPT
001550 104000          EMT    0
001552 012716 001574          MOV    #3#,(SP)          ;SET UP TO
001556 000002          RTI                                ;RETURN FROM INTERRUPT
001560 017705 013730          2#:  MOV    @DHSCR,R5        ;GET CONTENTS OF SYSTEM CONTROL REGISTER
001564          HLT    0                                ;UNEXPECTED INTERRUPT
001564 104000          EMT    0
001566 012716 001574          MOV    #3#,(SP)          ;SET UP TO
001572 000002          RTI                                ;RETURN FROM INTERRUPT
001574 016777 013740 013734 3#:  MOV    DHRLVL,@DHRVEC    ;RESTORE TRAPCATCHER
001602 005077 013732          CLR    @DHRLVL
001606 016777 013732 013726      MOV    DHTLVL,@DHTVEC
001614 005077 013724          CLR    @DHTLVL
001620 104400          SCOPE                                ;CHECK FOR ITERATIONS. LOOP
2 001622          INTST1  ↑/SILO OVERFLOW/,BIT12
                   ;INTERRUPT LOGIC TEST
                   ;SET SILO OVERFLOW INTERRUPT ENABLE
                   ;VERIFY THAT NO INTERRUPTS OCCUR

001622          TS \XN,4000,3#
001622 012767 000340 176146 T2:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
001630 012767 004000 013724      MOV    #4000,ICOUNT       ;SET UP FOR 4000 ITERATIONS
001636 012767 001756 013712      MOV    #3#,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST

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

001644 000003          MOV    #BIT11,@DHSCR       ;MASTER CLEAR INTERFACE
001652 012777 004000 013642      MOV    #1#,@DHRVEC       ;SET UP FOR POSSIBLE
                                ;RECEIVER INTERRUPT

```

```

001660 012777 000340 013652      MOV    #340, @DHRLVL
001666 012777 001742 013646      MOV    #21, @DHTVEC          ;SET UP FOR POSSIBLE
                                ;TRANSMITTER INTERRUPT

001674 012777 000340 013642      MOV    #340, @DHTLVL
001702 012777 010000 013604      MOV    #BIT12, @DHSCR       ;SET SILO OVERFLOW
                                ;INTERRUPT ENABLE
                                ;ALLOW INTERRUPTS
                                ;WINDOW FOR INTERRUPTS
                                ;NO INTERRUPT OCCURED
                                ;CONTINUE
001710 005067 176062                CLR    PS                    ;GET CONTENTS OF SYSTEM CONTROL REGISTER
001714 000240                NOP                            ;UNEXPECTED INTERRUPT
001716 012767 000340 176052      MOV    #340, PS
001724 000414                BR    3#
001726 017705 013562      1#:  MOV    @DHSCR, R5
001732 001732                HLT    0
001732 104000                EMT    0
001734 012716 001756                MOV    #31, (SP)            ;SET UP TO
001740 000002                RTI                            ;RETURN FROM INTERRUPT
001742 017705 013546      2#:  MOV    @DHSCR, R5
001746 001746                HLT    0                    ;GET CONTENTS OF SYSTEM CONTROL REGISTER
001746 104000                EMT    0                    ;UNEXPECTED INTERRUPT
001750 012716 001756                MOV    #31, (SP)            ;SET UP TO
001754 000002                RTI                            ;RETURN FROM INTERRUPT
001756 016777 013556 013552 3#:  MOV    DHRLVL, @DHRVEC       ;RESTORE TRAPCATCHER
001764 005077 013550                CLR    @DHRLVL
001770 016777 013550 013544      MOV    DHTLVL, @DHTVEC
001776 005077 013542                CLR    @DHTLVL
002002 104400                SCOPE
3 002004      INTST1  +/TRANSMITTER DONE/, BIT13 ;CHECK FOR ITERATIONS, LOOP
                                ;INTERRUPT LOGIC TEST
                                ;SET TRANSMITTER DONE INTERRUPT ENABLE
                                ;VERIFY THAT NO INTERRUPTS OCCUR

002004      TS \XN, 4000, 3#
002004 012767 000340 175764      T3:  MOV    #340, PS          ;DISABLE ALL INTERRUPTS
002012 012767 004000 013542      MOV    #4000, ICOUNT        ;SET UP FOR 4000 ITERATIONS
002020 012767 002140 013530      MOV    #31, ESCAPE          ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV    #, FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC
                                XN=XN+1

002026 012777 004000 013460      MOV    #BIT11, @DHSCR       ;MASTER CLEAR INTERFACE
002034 012777 002110 013474      MOV    #11, @DHRVEC        ;SET UP FOR POSSIBLE
                                ;RECEIVER INTERRUPT

002042 012777 000340 013470      MOV    #340, @DHRLVL
002050 012777 002124 013464      MOV    #21, @DHTVEC        ;SET UP FOR POSSIBLE
                                ;TRANSMITTER INTERRUPT

002056 012777 000340 013460      MOV    #340, @DHTLVL
002064 012777 020000 013422      MOV    #BIT13, @DHSCR       ;SET TRANSMITTER DONE
                                ;INTERRUPT ENABLE
                                ;ALLOW INTERRUPTS
                                ;WINDOW FOR INTERRUPTS
                                ;NO INTERRUPT OCCURED
                                ;CONTINUE
002072 005067 175700                CLR    PS                    ;GET CONTENTS OF SYSTEM CONTROL REGISTER
002076 000240                NOP                            ;UNEXPECTED INTERRUPT
002100 012767 000340 175670      MOV    #340, PS
002106 000414                BR    3#
002110 017705 013400      1#:  MOV    @DHSCR, R5
002114 002114                HLT    0
002114 104000                EMT    0
002116 012716 002140                MOV    #31, (SP)            ;SET UP TO
002122 000002                RTI                            ;RETURN FROM INTERRUPT
002124 017705 013364      2#:  MOV    @DHSCR, R5 ;GET CONTENTS OF SYSTEM CONTROL REGISTER

```



```

;SET SILO OVERFLOW INTERRUPT ENABLE
;SET SILO OVERFLOW (MAINTENANCE MODE IS ENABLED)
;VERIFY THAT AN INTERRUPT OCCURS

002360      002360 012767 000340 175110 TS \XN,4000,3#
002366      002366 012767 004000 013166 T5:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
002374      002374 012767 002524 013154      MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
                                MOV    #3#,ESCAPE           ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1             ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1

002402      002402 012777 004000 013104      MOV    @BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
002410      002410 012777 002502 013120      MOV    #1,@DHRVEC       ;SET UP FOR RECEIVER INTERRUPT
002416      002416 012777 000340 013114      MOV    #340,@DHRLVL
002424      002424 012777 002510 013110      MOV    #2,@DHTVEC       ;SET UP FOR TRANSMITTER INTERRUPT
002432      002432 012777 000340 013104      MOV    #340,@DHTLVL
002440      002440 012777 001000 013046      MOV    @BIT09,@DHSCR    ;SET MAINTENANCE MODE
002446      002446 052777 040000 013040      BIS    @BIT14,@DHSCR    ;SET SILO OVERFLOW
                                ;INTERRUPT ENABLE
                                ;FORCE INTERRUPT BY
                                ;SETTING SILO OVERFLOW
                                ;ALLOW INTERRUPTS
002454      002454 052777 010000 013032      BIS    @BIT12,@DHSCR    ;WINDOW FOR INTERRUPTS
                                ;NO INTERRUPT OCCURED
                                ;WITH SILO OVERFLOW INTERRUPT
002462      002462 005067 175310      CLR    PS
002466      002466 000240      NOP
002470      002470 012767 000340 175300      MOV    #340,PS
002476      002476      104001      HLT    1
                                EMT    1
                                ;ENABLE AND SILO OVERFLOW SET
                                ;ERROR

002500      002500 000411      BR     3#
002502      002502 012716 002524      1#:  MOV    #3#,(SP)       ;SET UP TO RETURN
002506      002506 000002      RTI
                                ;FROM VALID INTERRUPT
002510      002510 017705 013000      2#:  MOV    @DHSCR,R5      ;GET CONTENTS OF SYSTEM CONTROL REGISTER
002514      002514      104000      HLT    0
                                EMT    0
                                ;UNEXPECTED INTERRUPT
002516      002516 012716 002524      MOV    #3#,(SP)       ;SET UP TO RETURN
002522      002522 000002      RTI
                                ;FROM UNEXPECTED INTERRUPT
002524      002524 016777 013010 013004      3#:  MOV    @DHRLVL,@DHRVEC ;RESTORE TRAPCATCHER
002532      002532 005077 013002      CLR    @DHRLVL
002536      002536 016777 013002 012776      MOV    @DHTLVL,@DHTVEC
002544      002544 005077 012774      CLR    @DHTLVL
002550      002550 104400      SCOPE
6 002552      002552      INTST2  ↑/NON EXISTATN MEMORY/,BIT10,↑/TRANSMITTER/,BIT13,2#.1#

;INTERRUPT LOGIC TEST
;SET TRANSMITTER INTERRUPT ENABLE
;SET NON EXISTATN MEMORY (MAINTENANCE MODE IS ENABLED)
;VERIFY THAT AN INTERRUPT OCCURS

002552      002552 012767 000340 175216 TS \XN,4000,3#
002560      002560 012767 004000 012774 T6:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
002566      002566 012767 002716 012762      MOV    #4000,ICOUNT      ;SET UP FOR 4000 ITERATIONS
                                MOV    #3#,ESCAPE           ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1             ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
000007

```

```

002574 012777 004000 012712      MOV      @BIT11,@DHSCR      ;MASTER CLEAR INTERFAC
002602 012777 002702 012726      MOV      @2,@DHRVEC       ;SET JP FOR RECEIVER INTERRUPT
002610 012777 000340 012722      MOV      @340,@DHRLVL     ;
002616 012777 002674 012716      MOV      @1,@DHTVEC       ;SET UP FOR TRANSMITTER INTERRUPT
002624 012777 000340 012712      MOV      @340,@DHTLVL     ;
002632 012777 001000 012654      MOV      @BIT09,@DHSCR    ;SET MAINTENANCE MODE
002640 052777 020000 012646      BIS      @BIT13,@DHSCR    ;SET TRANSMITTER
                                ;INTERRUPT ENABLE
                                ;FORCE INTERRUPT BY
                                ;SETTING NON EXISTATN MEMORY
002646 052777 002000 012640      BIS      @BIT10,@DHSCR    ;ALLOW INTERRUPTS
                                ;WINDOW FOR INTERRUPTS
002654 005067 175116                CLR      PS                ;NO INTERRUPT OCCURED
002660 000240                NOP                        ;WITH TRANSMITTER INTERRUPT
002662 012767 000340 175106      MOV      @340,PS
002670                HLT      1
002670                EMT      1
                                ;ENABLE AND NON EXISTATN MEMORY SET
                                ;ERROR
002672 000411                BR       3$
002674 012716 002716      1$:  MOV      @3$, (SP)       ;SET UP TO RETURN
002700 000002                RTI                       ;FROM VALID INTEPRUPT
002702 017705 012606      2$:  MOV      @DHSCR,R5       ;GET CONTENTS OF SYSTEM CONTROL REGISTER
002706                HLT      0
                                ;UNEXPECTED INTERRUPT
002706 104000                EMT      0
002710 012716 002716      MOV      @3$, (SP)       ;SET UP TO RETURN
002714 000002                RTI                       ;FROM UNEXPECTED INTERRUPT
002716 016777 012616 012612 3$:  MOV      @DHRLVL,@DHRVEC  ;RESTORE TRAPCATCHER
002724 005077 012610                CLR      @DHRLVL
002730 016777 012610 012604      MOV      @DHTLVL,@DHTVEC
002736 005077 012602      CLR      @DHTLVL
002742 104400                SCOPE
7 002744      INTST2  +/TRANSMITTER DONE/,BIT15,+/TRANSMITTER DONE/,BIT13.2$,1$
                                ;INTERRUPT LOGIC TEST
                                ;SET TRANSMITTER DONE INTERRUPT ENABLE
                                ;SET TRANSMITTER DONE (MAINTENANCE MODE IS ENABLED
                                ;VERIFY THAT AN INTERRUPT OCCURS

002744      TS \XN,4000,3$
002744 012767 000340 175024 T7:  MOV      @340,PS          ;DISABLE ALL INTERRUPTS
002752 012767 004000 012602      MOV      @4000,ICOUNT    ;SET UP FOR 4000 ITERATIONS
002760 012767 003110 012570      MOV      @3$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      @,FREEZ1          ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
002766 000010                MOV      @BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
002774 012777 003074 012534      MOV      @2,@DHRVEC       ;SET UP FOR RECEIVER INTERRUPT
003002 012777 000340 012530      MOV      @340,@DHRLVL     ;
003010 012777 003066 012524      MOV      @1,@DHTVEC       ;SET UP FOR TRANSMITTER INTERRUPT
003016 012777 000340 012520      MOV      @340,@DHTLVL     ;
003024 012777 001000 012462      MOV      @BIT09,@DHSCR    ;SET MAINTENANCE MODE
003032 052777 020000 012454      BIS      @BIT13,@DHSCR    ;SET TRANSMITTER DONE
                                ;INTERRUPT ENABLE
                                ;FORCE INTERRUPT BY
003040 052777 100000 012446      BIS      @BIT15,@DHSCR    ;SETTING TRANSMITTER DONE
                                ;ALLOW INTERRUPTS
003046 005067 174724                CLR      PS                ;WINDOW FOR INTERRUPTS
003052 000240                NOP

```

```

003054 012767 000340 174714      MOV    #340,PS      ;NO INTERRUPT OCCURED
003062                                HLT    1            ;WITH TRANSMITTER DONE INTERRUPT
003062 104001                                EMT    1            ;ENABLE AND TRANSMITTER DONE SET
                                           ;ERROR
003064 000411                                BR     3#           ;SET UP TO RETURN
003066 012716 003110      1#:    MOV    #3#,(SP) ;FROM VALID INTERRUPT
003072 000002                                RTI                                ;SET CONTENTS OF SYSTEM CONTROL REGISTER
003074 017705 012414      2#:    MOV    @DHSCR,R5 ;UNEXPECTED INTERRUPT
003100                                HLT    0
003100 104000                                EMT    0
003102 012716 003110                                MOV    #3#,(SP)      ;SET UP TO RETURN
003106 000002                                RTI                                ;FROM UNEXPECTED INTERRUPT
003110 016777 012424 012420 3#:    MOV    DHRLVL,@DHVVEC ;RESTORE TRAPCATCHER
003116 0C5077 012416                                CLR    @DHRLVL
003122 016777 012416 012412      MOV    DHTLVL,@DHTVEC
003130 005077 012410                                CLR    @DHTLVL
003134 104400                                SCOPE
9                                000000      LINE=0
10                               000000      XLINE='LINE
11                               000001      BITX=1
12                               000001      XBIT=BITX
14                               000020      .REPT 20
15                               NPRTS \LINE,\BITX
16                               .NLIST
17                               LINE=LINE+1
18                               BITX=BITX+BITX
19                               .LIST
20                               .ENDR
003136      NPRTS \LINE,\BITX

;NPR LOGIC TEST
;SET BYTE COUNT TO 1 FOR LINE 0
;SET BAR BIT FOR LINE 0
;DCLAY FOR NPR
;VERIFY THAT BAR BIT FOR LINE 0 CLEARS
;VERIFY THAT TRANSMITTER DONE IS SET

003136      TS \XN,20,6#
003136 012767 000340 174632      T10:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
003144 012767 000020 012410      MOV    #20,ICOUNT    ;SET UP FOR 20 ITERATIONS
003152 012767 003366 012376      MOV    #6#,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV    #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
003160 000011                                MOV    #BIT11,@DHSCR ;ISSUE MASTER CLEAR
003166 004767 012230                                JSR    PC,CLEAR      ;CLEAR ALL BUS ADDRESS
                                           ;AND BYTE COUNT MEMORY LOCATIONS
003172 012777 003254 012342      MOV    #2#,@DHTJEC   ;SET UP TRANSMITTER
003200 012777 000340 012336      MOV    #340,@DHTLVL ;INTERRUPT VECTOR
003206 012777 000000 012300      MOV    #0,@DHSCR     ;SELECT LINE 0
003214 012777 177777 012302      MOV    #-1,@DHBC     ;SET BYTE COUNT TO 1
003222 012700 001000                                MOV    #1000,R0
003226 012777 000001 012272      MOV    #1,@DHBAR     ;SET BAR BIT FOR
                                           ;LINE 0
003234 052777 020000 012252      BIS    #BIT13,@DHSCR ;SET TRANSMITTER INTERRUPT ENADLE

```

```

003242 005067 174530          CLR    PS          ;ALLOW INTERRUPTS
003246 005300          1$: DEC    RO          ;DELAY FOR NPR
003250 001376          BNE    1$
003252          HLT    1          ;NO INTERRUPT OCCURED, ERROR
003252 104001          EMT    1
003254 005777 012234          2$: TST    @DHSCR          ;VERIFY THAT TRANSMITTER
                                ;DONE IS SET
003260 100401          BMI    3$
003262          HLT    2          ;TRANSMITTER DONE NOT SET, ERROR
003262 104002          EMT    2
003264 005777 012236          3$: TST    @DHBAR          ;WAS BAR BIT CLEARED FOR LINE 0
003270 001404          BEQ    4$
003272 005005          CLR    R5          ;(R5)=EXPECTED DATA IN
                                ;BUFFER ACTIVE REGISTER, 0
003274 017704 012226          MOV    @DHBAR,R4          ;(R4)=ACTUAL CONTENTS OF
                                ;BUFFER ACTIVE REGISTER
003300          HLT    0          ;BUS ACTIVE BIT NOT CLEARED, ERROR
003300 104000          EMT    0
003302 022777 000001 012212 4$: CMP    #1,@DHBA          ;WAS BUS ADDRESS INCREMENTED
003310 001405          BEQ    5$
003312 017704 012204          MOV    @DHBA,R4          ;(R4)=ACTUAL CONTENTS
                                ;OF BUS ADDRESS MEMORY FOR
                                ;LINE 0
003316 012705 000001          MOV    #1,R5          ;(R5)=EXPECTED VALUE OF
                                ;BUS ADDRESS MEMORY FOR
                                ;LINE 0, 1
003322          HLT    3          ;BUS ADDRESS NOT UPDATED
003322 104003          EMT    3
                                ;CORRECTLY, ERROR
003324 005777 012174          5$: TST    @DHBC          ;DID BYTE COUNT DECREMENT TO 0
003330 001416          BEQ    6$
003332 017704 012166          MOV    @DHBC,R4          ;(R4)=ACTUAL VALUE OF BYTE
                                ;COUNT FOR LINE 0
003336 005005          CLR    R5          ;(R5)=EXPECTED VALUE OF BYTE
                                ;COUNT FOR LINE 0, 0
003340          HLT    4          ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR
003340 104004          EMT    4
003342 016777 012172 012166          MOV    @DHLVL,@DHRVEC ;RESTORE TRAPCATCHER
003350 005077 012164          CLR    @DHLVL
003354 016777 012164 012160          MOV    @DHTLVL,@DHTVEC
003362 005077 012156          CLR    @DHTLVL
003366 012706 017270          6$: MOV    @STACK,SP          ;RESTORE STACK
003372 104400          SCOPE          ;CHECK FOR ITERATIONS, LOOP
000001          LINE=LINE+1
000002          BITX=BITX+BITX
003374          NFRTS \LINE,\BITX

                                ;NPR LOGIC TEST
                                ;SET BYTE COUNT TO J FOR LINE 1
                                ;SET BAR BIT FOR LINE 1
                                ;DELAY FOR NPR
                                ;VERIFY THAT BAR BIT FOR LINE 1 CLEARS
                                ;VERIFY THAT TRANSMITTER DONE IS SET

003374          TS \XN,20,6$
003374 012767 000340 174374 T11: MOV    #340,PS          ;DISABLE ALL INTERRUPTS
003402 012767 000020 012152          MOV    #20,ICOUNT          ;SET UP FOR 20 ITERATIONS

```

```

003410 012767 003624 012140      MOV    #6,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV    #,FREEZ1    ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
003416 012777 004000 012070      MOV    #BIT11,@DHSCR   ;ISSUE MASTER CLEAR
0C3424 004767 011772                JSR    PC,CLEAR        ;CLEAR ALL BUS ADDRESS
                                ;AND BYTE COUNT MEMORY LOCATIONS
003430 012777 003512 012104      MOV    #2,@DHTVEC     ;SET UP TRANSMITTER
003436 012777 000340 012100      MOV    #340,@DHTLVL   ;INTERRUPT VECTOR
003444 012777 000001 012002      MOV    #1,@DHSCR      ;SELECT LINE 1
003452 012777 177777 012044      MOV    #-1,@DHBC      ;SET BYTE COUNT TO 1
003460 012700 001000                MOV    #1000,R0
003464 012777 000002 012034      MOV    #2,@DHBAR      ;SET BAR BIT FOR
                                ;LINE 1
003472 052777 020000 012014      BIS    #BIT13,@DHSCR   ;SET TRANSMITTER INTERRUPT ENABLE
003500 005067 174272                CLR    PS              ;ALLOW INTERRUPTS
003504 005300                1#:   DEC    RO        ;DELAY FOR NPR
003506 001376                BNE    1#
003510                HLT    1              ;NO INTERRUPT OCCUPED, ERROR
003510 104001                EMT    1
003512 005777 011776                2#:   TST    @DHSCR    ;VERIFY THAT TRANSMITTER
                                ;DONE IS SET
003516 100401                BMI    3#
003520                HLT    2              ;TRANSMITTER DONE NOT SET, ERROR
003520 104002                EMT    2
003522 005777 012000                3#:   TST    @DHBAR    ;WAS BAR BIT CLEARED FOR LINE 1
003526 001404                BEQ    4#
003530 005005                CLR    R5              ;(R5)=EXPECTED DATA IN
                                ;BUFFER ACTIVE REGISTER, 0
003532 017704 011770                MOV    @DHBAR,R4      ;(R4)=ACTUAL CONTENTS OF
                                ;BUFFER ACTIVE REGISTER
003536                HLT    0              ;BUS ACTIVE BIT NOT CLEARED, ERROR
003536 104000                EMT    0
003540 022777 000001 011754        4#:   CMP    #1,@DHBA    ;WAS BUS ADDRESS INCREMENTED
003546 001405                BEQ    5#
003550 017704 011746                MOV    @DHBA,R4      ;(R4)=ACTUAL CONTENTS
                                ;OF BUS ADDRESS MEMORY FOR
                                ;LINE 1
003554 012705 000001                MOV    #1,R5         ;(R5)=EXPECTED VALUE OF
                                ;BUS ADDRESS MEMORY FOR
                                ;LINE 1, 1
003560                HLT    3              ;BUS ADDRESS NOT UPDATED
003560 104003                EMT    3
003562 005777 011736                5#:   TST    @DHBC      ;CORRECTLY, ERROR
003566 001416                BEQ    6#             ;DID BYTE COUNT DECREMENT TO 0
003570 017704 011730                MOV    @DHBC,R4      ;(R4)=ACTUAL VALUE OF BYTE
                                ;COUNT FOR LINE 1
003574 005005                CLR    R5             ;(R5)=EXPECTED VALUE OF BYTE
                                ;COUNT FOR LINE 1, 0
003576                HLT    4              ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR
003576 104004                EMT    4
003600 016777 011734 011730        MOV    DHRLVL,@DHRVEC ;RESTORE TRAPCATCHER
003606 005077 011726                CLR    @DHRLVL
003612 016777 011726 011722        MOV    DHTLVL,@DHTVEC
003620 005077 011720                CLR    @DHTLVL

```



```

003624 012706 017270      6#:  MOV    #STACK,SP      ;RESTCRE STACK
003630 104400                SCOPE                      ;CHECK FOR ITERATIONS, LOOP
      000002                LINE=LINE+1
      000004                BITX=BITX+BITX
003632                NPRTS  \LINE,\BITX

                        ;NPR LOGIC TEST
                        ;SET BYTE COUNT TO 1 FOR LINE 2
                        ;SET BAR BIT FOR LINE 2
                        ;DELAY FOR NPR
                        ;VERIFY THAT BAR BIT FOR LINE 2 CLEARS
                        ;VERIFY THAT TRANSMITTER DONE IS SET

003632                TS  \XN,20,6#
003632 012767 000340 174136 T12:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
003640 012767 000020 011714      MOV    #20,ICOUNT      ;SET UP FOR 20 ITERATIONS
003646 012767 004062 011702      MOV    #6#,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST

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

003654 012777 004000 011632      MOV    #BIT11,@DHSCR  ;ISSUE MASTER CLEAR
003662 004767 011534                JSR    PC,CLEAR        ;CLEAR ALL BUS ADDRESS
                        ;AND BYTE COUNT MEMORY LOCATIONS
003666 012777 003750 011646      MOV    #2,@DHTVEC     ;SET UP TRANSMITTER
003674 012777 000340 011642      MOV    #340,@DHTLVL  ;INTERRUPT VECTOR
003702 012777 000002 011604      MOV    #2,@DHSCR     ;SELECT LINE 2
003710 012777 177777 011606      MOV    #-1,@DHBC     ;SET BYTE COUNT TO 1
003716 012700 001000                MOV    #1000,RO
003722 012777 000004 011576      MOV    #4,@DHBAR     ;SET BAR BIT FOR
                        ;LINE 2
003730 052777 020000 011556      BIS    #BIT13,@DHSCR ;SET TRANSMITTER INTERRUPT ENABLE
003736 005067 174034                CLR    PS             ;ALLOW INTERRUPTS
003742 005300                1#:  DEC    RO          ;DELAY FOR NPR
003744 001376                BNE   1#
003746                HLT   1#          ;NO INTERRUPT OCCURED, ERROR
003746 104001                EMT   1#
003750 005777 011540                2#:  TST   @DHSCR     ;VERIFY THAT TRANSMITTER
                        ;DONE IS SET

003754 100401                BMI   3#
003756                HLT   2#          ;TRANSMITTER DONE NOT SET, ERROR
003756 104002                EMT   2#
003760 005777 011542                3#:  TST   @DHBAR     ;WAS BAR BIT CLEARED FOR LINE 2
003764 001404                BEQ   4#
003766 005005                CLR   R5
                        ;(R5)=EXPECTED DATA IN
                        ;BUFFER ACTIVE REGISTER, 0
003770 017704 011532                MOV   @DHBAR,R4      ;(R4)=ACTUAL CONTENTS OF
                        ;BUFFER ACTIVE REGISTER
003774                HLT   0#          ;BUS ACTIVE BIT NOT CLEARED, ERROR
003774 104000                EMT   0#
003776 022777 000001 011516      4#:  CMP   #1,@DHBA     ;WAS BUS ADDRESS INCREMENTED
004004 001405                BEQ   5#
004006 017704 011510                MOV   @DHBA,R4      ;(R4)=ACTUAL CONTENTS
                        ;OF BUS ADDRESS MEMORY FOR
                        ;LINE 2
004012 012705 000001                MOV   #1,R5         ;(R5)=EXPECTED VALUE OF
                        ;BUS ADDRESS MEMORY FOR

```

```

004016          HLT      3          ;LINE 2, 1
004016 104003   EMT      3          ;BUS ADDRESS NOT UPDATED

004020 005777 011500   5#: TST      @DHBC          ;CORRECTLY, ERROR
004024 001416          BEQ      6#          ;DID BYTE COUNT DECREMENT TO 0
004026 017704 011472   MOV      @DHBC,R4          ;(R4)=ACTUAL VALUE OF BYTE
                                ;COUNT FOR LINE 2
004032 005005          CLR      R5          ;(R5)=EXPECTED VALUE OF BYTE
                                ;COUNT FOR LINE 2, 0
004034          HLT      4          ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR
004034 104004   EMT      4
004036 016777 011476 011472   MOV      DHRLVL,@DHRVEC ;RESTORE TRAPCATCHER
004044 005077 011470          CLR      @DHRLVL
004050 016777 011470 011464   MOV      DHTLVL,@DHTVEC
004056 005077 011462          CLR      @DPTLVL
004062 012706 017270   6#: MOV      @STACK,SP          ;RESTORE STACK
004066 104400          SCOPE          ;CHECK FOR ITERATIONS, LOOP
                                LINE=LINE+1
                                BITX=BITX+BITX
004070          NPRTS  \LINE,\BITX

                                ;NPR LOGIC TEST
                                ;SET BYTE COUNT TO 1 FOR LINE 3
                                ;SET BAR BIT FOR LINE 3
                                ;DELAY FOR NPR
                                ;VERIFY THAT BAR BIT FOR LINE 3 CLEARS
                                ;VERIFY THAT TRANSMITTER DONE IS SET

004070          TS \XN,20,6#
004070 012767 000340 173700 T13: MOV      @340,PS          ;DISABLE ALL INTERRUPTS
004076 012767 000020 011456   MOV      @20,ICOUNT        ;SET UP FOR 20 ITERATIONS
004104 012767 004320 011444   MOV      @6#,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST

                                .IF NB <>
004112          MOV      @,FREEZ1          ;SET UP TO LO. WITH DATA ; 3
004120 004767 011276   .ENDC
                                XN=XN+1

004112 012777 004000 011374   MOV      @BIT11,@DHSCR      ;ISSUE MASTER CLEAR
004120 004767 011276   JSR      PC,CLEAR          ;CLEAR ALL BUS ADDRESS
                                ;AND BYTE COUNT MEMORY LOCATIONS
004124 012777 004206 011410   MOV      @2,@DHTVEC        ;SET UP TRANSMITTER
004132 012777 000340 011404   MOV      @340,@DHTLVL      ;INTERRUPT VECTOR
004140 012777 000003 011346   MOV      @3,@DHSCR         ;SELECT LINE 3
004146 012777 177777 011350   MOV      @-1,@DHBC         ;SET BYTE COUNT TO 1
004154 012700 001000          MOV      @1000,R0
004160 012777 000010 011340   MOV      @10,@DHBAR        ;SET BAR BIT FOR
                                ;LINE 3
004166 052777 020000 011320   BIS      @BIT13,@DHSCR      ;SET TRANSMITTER INTERRUPT ENABLE
004174 005067 173576          CLR      PS                ;ALLOW INTERRUPTS
004200 005300          1#: DEC      R0            ;DELAY FOR NPR
004202 001376          BNE     1#
004204          HLT      1          ;NO INTERRUPT OCCURED, ERROR
004204 104001          EMT      1
004206 005777 011302   2#: TST      @DHSCR         ;VERIFY THAT TRANSMITTER
                                ;DONE IS SET

004212 100401          BMI     3#
004214          HLT      2          ;TRANSMITTER DONE NOT SET, ERROR

```

```

004214 104002          EMT      2
004216 005777 011304  3$:   TST      @DHBAR      ;WAS BAR BIT CLEARED FOR LINE 3
004222 001404          BEQ      4$
004224 005005          CLR      R5              ;(R5)=EXPECTED DATA IN
                          ;BUFFER ACTIVE REGISTER, 0
004226 017704 011274          MOV      @DHBAR,R4      ;(R4)=ACTUAL CONTENTS OF
                          ;BUFFER ACTIVE REGISTER
004232          HLT      0              ;BUS ACTIVE BIT NOT CLEARED, ERROR
004232 104000          EMT      0
004234 022777 000001 011260 4$:   CMP      @1,@DHBA      ;HAS BUS ADDRESS INCREMENTED
004242 001405          BEQ      5$
004244 017704 011252          MOV      @DHBA,R4      ;(R4)=ACTUAL CONTENTS
                          ;OF BUS ADDRESS MEMORY FOR
                          ;LINE 3
004250 012705 000001          MOV      @1,R5          ;(R5)=EXPECTED VALUE OF
                          ;BUS ADDRESS MEMORY FOR
                          ;LINE 3, 1
004254          HLT      3              ;BUS ADDRESS NOT UPDATED
004254 104003          EMT      3
004256 005777 011242          5$:   TST      @DHBC
004262 001416          BEQ      6$
004264 017704 011234          MOV      @DHBC,R4      ;(R4)=ACTUAL VALUE OF BYTE
                          ;COUNT FOR LINE 3
004270 005005          CLR      R5              ;(R5)=EXPECTED VALUE OF BYTE
                          ;COUNT FOR LINE 3, 0
004272          HLT      4              ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR
004272 104004          EMT      4
004274 016777 011240 011234          MOV      @HRLVL,@DHRVEC ;RESTORE TRAPCATCHER
004302 005077 011232          CLR      @HRLVL
004304 016777 011232 011226          MOV      @HTLVL,@DHTVEC
004314 005077 011224          CLR      @HTLVL
004320 012706 017270          6$:   MOV      @STACK,SP      ;RESTORE STACK
004324 104400          SCOPE
004324 000004          LINE=LINE+1
004324 000020          BITX=BITX+BITX
004326          NPRTS  \LINE,\BITX

                          ;NPR LOGIC TEST
                          ;SET BYTE COUNT TO 1 FOR LINE 4
                          ;SET BAR BIT FOR LINE 4
                          ;DELAY FOR NPR
                          ;VERIFY THAT BAR BIT FOR LINE 4 CLEARS
                          ;VERIFY THAT TRANSMITTER DONE IS SET

004326          TS  \XN,20,6$
004326 012767 000340 173442 T14:  MOV      @340,PS      ;DISABLE ALL INTERRUPTS
004334 012767 000020 011220          MOV      @20,ICOUNT    ;SET UP FOR 20 ITERATIONS
004342 012767 004556 011206          MOV      @6$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST

                          .IF NB  <>
004350          MOV      @,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                          .ENDC
004350 000015          XN=XN+1
004350 012777 004000 011136          MOV      @BIT11,@DHSCR ;ISSUE MASTER CLEAR
004356 004767 011040          JSR      PC,CLEAR      ;CLEAR ALL BUS ADDRESS
                          ;AND BYTE COUNT MEMORY LOCATIONS
004362 012777 004444 011152          MOV      @2$,@DHTVEC   ;SET UP TRANSMITTER

```



```

;DELAY FOR NPR
;VERIFY THAT BAR BIT FOR LINE 5 CLEARS
;VERIFY THAT TRANSMITTER DONE IS SET

004564 TS \XN,20,6#
004564 012767 000340 173204 T15: MOV #340,PS ;DISABLE ALL INTERRUPTS
004572 012767 000020 010762 MOV #20,ICOUNT ;SET UP FOR 20 ITERATIONS
004600 012767 005014 010750 MOV #6#,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST

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

004606 000016
004614 012777 004000 010700 MOV #BIT11,@DHSCR ;ISSUE MASTER CLEAR
;CLEAR ALL BUS ADDRESS
;AND BYTE COUNT MEMORY LOCATIONS
004620 012777 004702 010714 MOV #2#,@DHTEC ;SET UP TRANSMITTER
004626 012777 000340 010710 MOV #340,@DHTEC ;INTERRUPT VECTOR
004634 012777 000005 010652 MOV #5,@DHSCR ;SELECT LINE 5
004642 012777 177777 010654 MOV #-1,@DHBC ;SET BYTE COUNT TO 1
004650 012700 001000
004654 012777 000040 010644 MOV #40,@DHBAR ;SET BAR BIT FOR
;LINE 5
004662 052777 020000 010624 BIS #BIT13,@DHSCR ;SET TRANSMITTER INTERRUPT ENABLE
004670 005067 173102 CLR PS ;ALLOW INTERRUPTS
004674 005300 1# : DEC RO ;DELAY FOR NPR
004676 001376 BNE 1#
004700 HLT 1 ;NO INTERRUPT OCCURED, ERROR
004700 104001 EMT 1
004702 005777 010606 2# : TST @DHSCR ;VERIFY THAT TRANSMITTER
;DONE IS SET

004706 100401 BMI 3#
004710 HLT 2 ;TRANSMITTER DONE NOT SET, ERROR
004710 104002 EMT 2
004712 005777 010610 3# : TST @DHBAR ;WAS BAR BIT CLEARED FOR LINE 5
004716 001404 BEQ 4#
004720 005005 CLR R5 ;(R5)=EXPECTED DATA IN
;BUFFER ACTIVE REGISTER, 0
004722 017704 010600 MOV @DHBAR,R4 ;(R4)=ACTUAL CONTENTS OF
;BUFFER ACTIVE REGISTER
;BUS ACTIVE BIT NOT CLEARED, ERROR

004726 HLT 0
004726 104000 EMT 0
004730 022777 000001 010564 4# : CMP #1,@DHBA ;WAS BUS ADDRESS INCREMENTED
004736 001405 BEQ 5#
004740 017704 010556 MOV @DHBA,R4 ;(R4)=ACTUAL CONTENTS
;OF BUS ADDRESS MEMORY FOR
;LINE 5
004744 012705 000001 MOV #1,R5 ;(R5)=EXPECTED VALUE OF
;BUS ADDRESS MEMORY FOR
;LINE 5, 1
;BUS ADDRESS NOT UPDATED

004750 HLT 3
004750 104003 EMT 3 ;CORRECTLY, ERROR
;DID BYTE COUNT DECREMENT TO 0

004752 005777 010546 5# : TST @DHBC
004756 001416 BEQ 6#
004760 017704 010540 MOV @DHBC,R4 ;(R4)=ACTUAL VALUE OF BYTE
;COUNT FOR LINE 5
004764 005005 CLR R5 ;(R5)=EXPECTED VALUE OF BYTE

```

```

                                ;COUNT FOR LINE 5, 0
004766                                HLT      4                                ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR
004766 104004                                EMT      4
004770 016777 010544 010540                MOV      @HRLVL,@DHRVEC ;RESTORE TRAPCATCHER
004776 005077 010536                                CLR      @HRLVL
005002 016777 010536 010532                MOV      @DHTLVL,@DHTVEC
005010 005077 010530                                CLR      @DHTLVL
005014 012706 017270                6#:     MOV      @STACK,SP ;RESTORE STACK
005020 104400                                SCOPE
                                LINE=LINE+1 ;CHECK FOR ITERATIONS, LOOP
                                BITX=BITX+BITX
005022 000006                                NPRTS   \LINE,\BITX
                                000100
                                ;NPR LOGIC TEST
                                ;SET BYTE COUNT TO 1 FOR LINE 6
                                ;SET BAR BIT FOR LINE 6
                                ;DELAY FOR NPR
                                ;VERIFY THAT BAR BIT FOR LINE 6 CLEARS
                                ;VERIFY THAT TRANSMITTER DONE IS SET

005022                                TS \XN,20,6#
005022 012767 000340 172746                T16:    MOV      @340,PS ;DISABLE ALL INTERRUPTS
005030 012767 000020 010524                MOV      @2C,ICOUNT ;SET UP FOR 20 ITERATIONS
005036 012767 005252 010512                MOV      @6#,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB
                                <>
                                MOV      @,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC
                                XN=XN+1
005044 000017                                MOV      @BIT11,@DHSCR ;ISSUE MASTER CLEAR
005052 004767 004000 010442                JSR      PC,CLEAR ;CLEAR ALL BUS ADDRESS
                                ;AND BYTE COUNT MEMORY LOCATIONS
005056 012777 005140 010456                MOV      @2,@DHTVEC ;SET UP TRANSMITTER
005064 012777 000340 010452                MOV      @340,@DHTLVL ;INTERRUPT VECTOR
005072 012777 000006 010414                MOV      @6,@DHSCR ;SELECT LINE 6
005100 012777 177777 010416                MOV      @-1,@DHBC ;SET BYTE COUNT TO 1
005106 012700 001000                                MOV      @1000,R0
005112 012777 000100 010406                MOV      @100,@DHBAR ;SET BAR BIT FOR
                                ;LINE 6
005120 052777 020000 010366                BIS      @BIT13,@DHSCR ;SET TRANSMITTER INTERRUPT ENABLE
005126 005067 172644                                CLR      PS ;ALLOW INTERRUPTS
005132 005300                1#:    DEC      R0 ;DELAY FOR NPR
005134 001376                                BNE     1#
005136                                HLT      1 ;NO INTERRUPT OCCURED, ERROR
005136 104001                                EMT      1
005140 005777 010350                2#:    TST      @DHSCR ;VERIFY THAT TRANSMITTER
                                ;DONE IS SET
005144 100401                                BMI     3#
005146                                HLT      2 ;TRANSMITTER DONE NOT SET, ERPOP
005146 104002                                EMT      2
005150 005777 010352                3#:    TST      @DHBAR ;WAS BAR BIT CLEARED FOR LINE 6
005154 001404                                BEQ     4#
005156 005005                                CLR      R5 ;(R5)=EXPECTED DATA IN
                                ;BUFFER ACTIVE REGISTER, 0
005160 017704 010312                MOV      @DHBAR,R4 ;(R4)=ACTUAL CONTENTS OF
                                ;BUFFER ACTIVE REGISTER
005164                                HLT      0 ;BUS ACTIVE BIT NOT CLEARED, ERROR
005164 104000                                EMT      0

```

```

005166 022777 000001 010326 4:  CMP    #1,@DHBA      ;WAS BUS ADDRESS INCREMENTED
005174 001405          BEQ    5:
005176 017704 010320          MOV    @DHBA,R4      ;(R4)=ACTUAL CONTENTS
                                ;OF BUS ADDRESS MEMORY FOR
                                ;LINE 6
005202 012705 000001          MOV    #1,R5        ;(R5)=EXPECTED VALUE OF
                                ;BUS ADDRESS MEMORY FOR
                                ;LINE 6, 1
005206          HLT    3          ;BUS ADDRESS NOT UPDATED
005206 104003          EMT    3
005210 005777 010310 5:  TST    @DHBC        ;CORRECTLY, ERROR
005214 001416          BEQ    6:          ;DID BYTE COUNT DECREMENT TO 0
005216 017704 010302          MOV    @DHBC,R4    ;(R4)=ACTUAL VALUE OF BYTE
                                ;COUNT FOR LINE 6
005222 005005          CLR    R5          ;(R5)=EXPECTED VALUE OF BYTE
                                ;COUNT FOR LINE 6, 0
005224          HLT    4          ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR
005224 104004          EMT    4
005226 016777 010306 010302  MOV    @DHLVL,@DHRVEC ;RESTORE TRAPCATCHER
005234 005077 010300          CLR    @DHLVL
005240 016777 010300 010274  MOV    @DHTLVL,@DHTVEC
005246 005077 010272          CLR    @DHTLVL
005252 012706 017270 6:  MOV    @STACK,SP   ;RESTORE STACK
005256 104400          SCOPE              ;CHECK FOR ITERATIONS, LOOP
                                LINE=LINE+1
005260 000200          BITX=BITX+BITX
                                NPRTS \LINE,\BITX

                                ;NPR LOGIC TEST
                                ;SET BYTE COUNT TO 1 FOR LINE 7
                                ;SET BAR BIT FOR LINE 7
                                ;DELAY FOR NPR
                                ;VERIFY THAT BAR BIT FOR LINE 7 CLEARS
                                ;VERIFY THAT TRANSMITTER DONE IS SET

005260          TS \XN,20,6:
005260 012767 000340 172510 T17: MOV    #340,PS    ;DISABLE ALL INTERRUPTS
005266 012767 000020 010266  MOV    #20,ICOUNT  ;SET UP FOR 20 ITERATIONS
005274 012767 005510 010254  MOV    #6,@ESCAPE  ;SET UP TO ESCAPE TO NEXT TEST

                                .IF NB <>
                                MOV    @,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
005302 000020          MOV    @BIT11,@DHSCR ;ISSUE MASTER CLEAR
005310 004767 010106 010204  JSR    PC,CLEAR    ;CLEAR ALL BUS ADDRESS
                                ;AND BYTE COUNT MEMORY LOCATIONS
005314 012777 005376 010220  MOV    #2,@DHTVEC  ;SET UP TRANSMITTER
005322 012777 000340 010214  MOV    #340,@DHTLVL ;INTERRUPT VECTOR
005330 012777 000007 010156  MOV    #7,@DHSCR   ;SELECT LINE 7
005336 012777 177777 010160  MOV    #-1,@DHBC   ;SET BYTE COUNT TO 1
005344 012700 0C1000          MOV    #1000,R0
005350 012777 000200 010150  MOV    #200,@DHBAR ;SET BAR BIT FOR
                                ;LINE 7
005356 052777 020000 010130  BIS    @BIT13,@DHSCR ;SET TRANSMITTER INTERRUPT ENABLE
005364 005067 172406          CLR    PS          ;ALLOW INTERRUPTS
005370 005300 1:  DEC    R0          ;DELAY FOR NPR

```

```

005372 001376          SNE      1$
005374          HLT      1          ;NO INTERRUPT OCCURED, ERROR
005374 104001          EMT      1
005376 005777 010112 2$:  TST      @DHSCR          ;VERIFY THAT TRANSMITTER
                                ;DONE IS SET

005402 100401          BMI      3$
005404          HLT      2          ;TRANSMITTER DONE NOT SET, ERROR
005404 104002          EMT      2
005406 005777 010114 3$:  TST      @DHBAR          ;WAS BAR BIT CLEARED FOR LINE 7
005412 001404          BEQ      4$
005414 005005          CLR      R5          ;(R5)=EXPECTED DATA IN
                                ;BUFFER ACTIVE REGISTER, 0
                                ;(R4)=ACTUAL CONTENTS OF
                                ;BUFFER ACTIVE REGISTER
005416 017704 010104          MOV      @DHBAR,R4          ;BUS ACTIVE BIT NOT CLEARED, ERROR

005422          HLT      0
005422 104000          EMT      0
005424 022777 000001 010070 4$:  CMP      @1,@DHBA          ;WAS BUS ADDRESS INCREMENTED
005432 001405          BEQ      5$
005434 017704 010062          MOV      @DHBA,R4          ;(R4)=ACTUAL CONTENTS
                                ;OF BUS ADDRESS MEMORY FOR
                                ;LINE 7
005440 012705 000001          MOV      @1,R5          ;(R5)=EXPECTED VALUE OF
                                ;BUS ADDRESS MEMORY FOR
                                ;LINE 7, 1
005444          HLT      3          ;BUS ADDRESS NOT UPDATED
005444 104003          EMT      3
                                ;CORRECTLY, ERROR
005446 005777 010052 5$:  TST      @DHBC          ;DID BYTE COUNT DECREMENT TO 0
005452 001416          BEQ      6$
005454 017704 010044          MOV      @DHBC,R4          ;(R4)=ACTUAL VALUE OF BYTE
                                ;COUNT FOR LINE 7
005460 005005          CLR      R5          ;(R5)=EXPECTED VALUE OF BYTE
                                ;COUNT FOR LINE 7, 0
                                ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR

005462          HLT      4
005462 104004          EMT      4
005464 016777 010050 010044          MOV      @DHRLVL,@DHRVEC ;RESTORE TRAPCATCHER
005472 005077 010042          CLR      @DHRLVL
005476 016777 010042 010036          MOV      @DHTLVL,@DHTVEC
005504 005077 010034          CLW      @DHTLVL
005510 012706 017270 6$:  MOV      @STACK.SP          ;RESTORE STACK
005514 104400          SCOPE          ;CHECK FOR ITERATIONS, LOOP
                                LINE=LINE+1
                                BITX=BITX+BITX
005516          NPRTS  \LINE,\BITX

                                ;NPR LOGIC TEST
                                ;SET BYTE COUNT TO 1 FOR LINE 10
                                ;SET BAR BIT FOR LINE 10
                                ;DELAY FOR NPR
                                ;VERIFY THAT BAR BIT FOR LINE 10 CLEARS
                                ;VERIFY THAT TRANSMITTER DONE IS SET

005516          TS \XN,20,6$
005516 012767 000340 172252 T20:  MOV      @340,PS          ;DISABLE ALL INTERRUPTS
005524 012767 000020 010030          MOV      @20,ICOUNT          ;SET UP FOR 20 ITERATIONS
005532 012767 005746 010016          MOV      @6$,ESCAPE          ;SET UP TO ESCAPE TO NEXT TEST

                                .IF NB <>

```



```

                                MOV     #.FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
005540 012777 004000 007746      MOV     #BIT11,@DHSCR           ;ISSUE MASTER CLEAR
005546 004767 007650              JSR     PC,CLEAR              ;CLEAR ALL BUS ADDRESS
                                ;AND BYTE COUNT MEMORY LOCATIONS
005552 012777 005634 007762      MOV     #2,@DHTVEC            ;SET UP TRANSMITTER
005560 012777 000340 007756      MOV     #340,@DHTLVL          ;INTERRUPT VECTOR
005566 012777 000010 007720      MOV     #10,@DHSCR            ;SELECT LINE 10
005574 012777 177777 007722      MOV     #-1,@DMBC             ;SET BYTE COUNT TO 1
005602 012700 001000              MOV     #1000,R0
005606 012777 000400 007712      MOV     #400,@DMBAR           ;SET BAR BIT FOR
                                ;LINE 10
005614 052777 020000 007672      BIS     #BIT13,@DHSCR         ;SET TRANSMITTER INTERRUPT ENABLE
005622 005067 172150              CLR     PS                    ;ALLOW INTERRUPTS
005626 005300                    1$:  DEC     R0                  ;DELAY FOR NPR
005630 001376                    BNE     1$
005632                    HLT     1                      ;NO INTERRUPT OCCURED, ERROR
005632 104001                    EMT     1
005634 005777 007654                    2$:  TST     @DHSCR              ;VERIFY THAT TRANSMITTER
                                ;DONE IS SET
005640 100401                    BMI     3$
005642                    HLT     2                      ;TRANSMITTER DONE NOT SET, ERROR
005642 104002                    EMT     2
005644 005777 007656                    3$:  TST     @DMBAR              ;WAS BAR BIT CLEARED FOR LINE 10
005650 001404                    BEQ     4$
005652 005005                    CLR     R5
                                ;(R5)=EXPECTED DATA IN
                                ;BUFFER ACTIVE REGISTER, 0
005654 017704 007646      MOV     @DMBAR,R4             ;(R4)=ACTUAL CONTENTS OF
                                ;BUFFER ACTIVE REGISTER
005660                    HLT     0                      ;BUS ACTIVE BIT NOT CLEARED, ERROR
005660 104000                    EMT     0
005662 022777 000001 007632 4$:  CMP     #1,@DMBA              ;WAS BUS ADDRESS INCREMENTED
005670 001405                    BEQ     5$
005672 017704 007624      MOV     @DMBA,R4
                                ;(R4)=ACTUAL CONTENTS
                                ;OF BUS ADDRESS MEMORY FOR
                                ;LINE 10
005676 012705 000001      MOV     #1,R5                 ;(R5)=EXPECTED VALUE OF
                                ;BUS ADDRESS MEMORY FOR
                                ;LINE 10, 1
005702                    HLT     3                      ;BUS ADDRESS NOT UPDATED
005702 104003                    EMT     3
                                ;CORRECTLY, ERROR
005704 005777 007614                    5$:  TST     @DMBC              ;DID BYTE COUNT DECREMENT TO 0
005710 001416                    BEQ     6$
005712 017704 007606      MOV     @DMBC,R4
                                ;(R4)=ACTUAL VALUE OF BYTE
                                ;COUNT FOR LINE 10
005716 005005                    CLR     R5
                                ;(R5)=EXPECTED VALUE OF BYTE
                                ;COUNT FOR LINE 10, 0
005720                    HLT     4                      ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR
005720 104004                    EMT     4
005722 016777 007612 007606      MOV     @DTRLVL,@DHRVEC       ;RESTORE TRAPCATCHER
005730 005077 007604      CLR     @DTRLVL
005734 016777 007604 007600      MOV     @DTRLVL,@DHTVEC
005742 005077 007576      CLR     @DHTLVL
005746 012706 017270                    6$:  MOV     #STACK,SP           ;RESTORE STACK
005752 104400      SCOPE                    ;CHECK FOR ITERATIONS, LOOP

```

```

000011 LINE=LINE+1
001000 BITX=BITX+BITX
005754 NPRTS \LINE,\BITX

;NPR LOGIC TEST
;SET BYTE COUNT TO 1 FOR LINE 11
;SET BAR BIT FOR LINE 11
;DELAY FOR NPR
;VERIFY THAT BAR BIT FOR LINE 11 CLEARS
;VERIFY THAT TRANSMITTER DONE IS SET

005754 TS \XN,20,6#
005754 012767 000340 172014 T21: MOV #340,PS ;DISABLE ALL INTERRUPTS
005762 012767 000020 007572 MOV #20,ICOUNT ;SET UP FOR 20 ITERATIONS
005770 012767 006204 007560 MOV #6#,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST

;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3

.ENDC
XN=XN+1

005776 012777 004000 007510 MOV #BIT11,@DHSCR ;ISSUE MASTER CLEAR
006004 004767 007412 JSR PC,CLEAR ;CLEAR ALL BUS ADDRESS
;AND BYTE COUNT MEMORY LOCATIONS
006010 012777 006072 007524 MOV #2,@DHTEVC ;SET UP TRANSMITTER
006016 012777 000340 007520 MOV #340,@DHTEVL ;INTERRUPT VECTOR
006024 012777 000011 007462 MOV #11,@DHSCR ;SELECT LINE 11
006032 012777 177777 007464 MOV #-1,@DHBC ;SET BYTE COUNT TO 1
006040 012700 001000 MOV #1000,R0
006044 012777 001000 007454 MOV #1000,@DHBAR ;SET BAR BIT FOR
;LINE 11
006052 052777 020000 007434 BIS #BIT13,@DHSCR ;SET TRANSMITTER INTERRUPT ENABLE
006060 005067 171712 CLR PS ;ALLOW INTERRUPTS
006064 005300 1# : DEC R0 ;DELAY FOR NPR
006066 001376 BNE 1#
006070 HLT 1 ;NO INTERRUPT OCCURED, ERROR
006070 104001 EMT 1
006072 005777 007416 2# : TST @DHSCR ;VERIFY THAT TRANSMITTER
;DONE IS SET

006076 100401 BMI 3#
006100 HLT 2 ;TRANSMITTER DONE NOT SET, ERROR
006100 104002 EMT 2
006102 005777 007420 3# : TST @DHBAR ;WAS BAR BIT CLEARED FOR LINE 11
006106 001404 BEQ 4#
006110 005005 CLR R5 ;(R5)=EXPECTED DATA IN
;BUFFER ACTIVE REGISTER, 0
006112 017704 007410 MOV @DHBAR,R4 ;(R4)=ACTUAL CONTENTS OF
;BUFFER ACTIVE REGISTER
006116 HLT 0 ;BUS ACTIVE BIT NOT CLEARED, ERROR
006116 104000 EMT 0
006120 022777 000001 007374 4# : CMP #1,@DHBA ;WAS BUS ADDRESS INCREMENTED
006126 001405 BEQ 5#
006130 017704 007366 MOV @DHBA,R4 ;(R4)=ACTUAL CONTENTS
;OF BUS ADDRESS MEMORY FOR
;LINE 11
006134 012705 000001 MOV #1,R5 ;(R5)=EXPECTED VALUE OF
;BUS ADDRESS MEMORY FOR
;LINE 11, 1
006140 HLT 3 ;BUS ADDRESS NOT UPDATED

```

```

006140 104003          EMT      3
                                ;CORRECTLY, ERROR
006142 005777 007356  5$:  TST      @DHBC          ;DID BYTE COUNT DECREMENT TO 0
006146 001416          BEQ      6$
006150 017704 007350          MOV      @DHBC,R4          ;(R4)=ACTUAL VALUE OF BYTE
                                ;COUNT FOR LINE 11
006154 005005          CLR      R5              ;(R5)=EXPECTED VALUE OF BYTE
                                ;COUNT FOR LINE 11, 0
006156          HLT      4              ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR
006156 104004          EMT      4
006160 016777 007354 007350  MOV      @HRLVL,@DHRVEC ;RESTORE TRAPCATCHER
006166 005077 007346          CLR      @HRLVL
006172 016777 007346 007342  MOV      @HTLVL,@DHTLVL
006200 005077 007340          CLR      @DHTLVL
006204 012706 012720  6$:  MOV      @STACK,SP          ;RESTORE STACK
006210 104400          SCOPE
                                ;CHECK FOR ITERATIONS, LOOP
                                LINE=LINE+1
                                BITX=BITX+GITX
006212 000012          NPRTS  \LINE,\BITX
                                ;NPR LOGIC TEST
                                ;SET BYTE COUNT TO 1 FOR LINE 12
                                ;SET BAR BIT FOR LINE 12
                                ;DELAY FOR NPR
                                ;VERIFY THAT BAR BIT FOR LINE 12 CLEARS
                                ;VERIFY THAT TRANSMITTER DONE IS SET

006212          TS  \XN,20,6$
006212 012767 000340 171556 T22:  MOV      @340,PS          ;DISABLE ALL INTERRUPTS
006220 012767 000020 007334      MOV      @20,ICOUNT      ;SET UP FOR 20 ITERATIONS
006226 012767 006442 007322      MOV      @6$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      @,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
006234 012777 004000 007252  MOV      @BIT11,@DHSCR    ;ISSUE MASTER CLEAR
006242 004767 007154          JSR      PC,CLEAR        ;CLEAR ALL BUS ADDRESS
                                ;AND BYTE COUNT MEMORY LOCATIONS
006246 012777 006330 007266  MOV      @2$,@DH1VEC      ;SET UP TRANSMITTER
006254 012777 000340 007262  MOV      @340,@DHTLVL     ;INTERPUPT VECTOR
006262 012777 000012 007224  MOV      @12,@DHSCR       ;SELECT LINE 12
006270 012777 177777 007226  MOV      @-1,@DHBC        ;SET BYTE COUNT TO 1
006276 012700 001000          MOV      @1000,R0
006302 012777 002000 007216  MOV      @2000,@DHBAR     ;SET BAR BIT FOR
                                ;LINE 12
006310 052777 020000 007176  BIS      @BIT13,@DHSCR    ;SET TRANSMITTER INTERRUPT ENABLE
006316 005067 171454          CLR      PS              ;ALLOW INTERRUPTS
006322 005300          1$:  DEC      R0              ;DELAY FOR NPR
006324 001376          BNE     1$
006326          HLT     1              ;NO INTERRUPT OCCURED, ERROR
006326 104001          EMT     1
006330 005777 007160  2$:  TST      @DHSCR          ;VERIFY THAT TRANSMITTER
                                ;DONE IS SET
006334 100401          BMI     3$
006336          HLT     2              ;TRANSMITTER DONE NOT SET, ERROR
006336 104002          EMT     2
006340 005777 007162  3$:  TST      @DHBAR          ;WAS BAR BIT CLEARED FOR LINE 12

```

```

006344 001404          BEQ    4$
006346 005005          CLR    R5                ;(R5)=EXPECTED DATA IN
                                ;BUFFER ACTIVE REGISTER, 0
006350 017704 007152    MOV    @DHBAR,R4        ;(R4)=ACTUAL CONTENTS OF
                                ;BUFFER ACTIVE REGISTER
006354          HLT    0                ;BUS ACTIVE BIT NOT CLEARED, ERROR
006354 104000          EMT    0
006356 022777 000001 007136 4$:  CMP    #1,@DHBA        ;WAS BUS ADDRESS INCREMENTED
006364 001405          BEQ    5$
006366 017704 007130    MOV    @DHBA,R4        ;(R4)=ACTUAL CONTENTS
                                ;OF BUS ADDRESS MEMORY FOR
                                ;LINE 12
006372 012705 000001    MOV    #1,R5          ;(R5)=EXPECTED VALUE OF
                                ;BUS ADDRESS MEMORY FOR
                                ;LINE 12, 1
006376          HLT    3                ;BUS ADDRESS NOT UPDATED
006376 104003          EMT    3
006400 005777 007120    5$:  TST    @DHBC
006404 001416          BEQ    6$                ;CORRECTLY, ERROR
006406 017704 007112    MOV    @DHBC,R4        ;DID BYTE COUNT DECREMENT TO 0
                                ;(R4)=ACTUAL VALUE OF BYTE
                                ;COUNT FOR LINE 12
006412 005005          CLR    R5                ;(R5)=EXPECTED VALUE OF BYTE
                                ;COUNT FOR LINE 12, 0
006414          HLT    4                ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR
006414 104004          EMT    4
006416 016777 007116 007112    MOV    DHRLVL,@DHVEC  ;RESTORE TRAPCATCHER
006424 005077 007110          CLR    @DHRLVL
006430 016777 007110 007104    MOV    DHTLVL,@DHTVEC
006436 005077 007102          CLR    @DHTLVL
006442 012706 017270    6$:  MOV    #STACK,SP      ;RESTORE STACK
006446 104400          SCOPE                    ;CHECK FOR ITERATIONS, LOOP
                                LINE=LINE+1
006450 000013          BITX=BITX+BITX
                                NPRTS \LINE,\BITX
                                ;NPR LOGIC TEST
                                ;SET BYTE COUNT TO 1 FOR LINE 13
                                ;SET BAR BIT FOR LINE 13
                                ;DELAY FOR NPR
                                ;VERIFY THAT BAR BIT FOR LINE 13 CLEARS
                                ;VERIFY THAT TRANSMITTER DONE IS SET

006450          TS \XN,20,6$
006450 012767 000340 171320    T23:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
006456 012767 000020 007076    MOV    #20,ICOUNT     ;SET UP FOR 20 ITERATIONS
006464 012767 006700 007064    MOV    #6$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV    #,FREEZ1        ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
006472 000024          MOV    #BIT11,@DHSCR  ;ISSUE MASTER CLEAR
006500 012777 004000 007014    JSR    PC,CLEAR       ;CLEAR ALL BUS ADDRESS
                                ;AND BYTE COUNT MEMORY LOCATIONS
006504 012777 006566 007030    MOV    #2$,@DHTVEC    ;SET UP TRANSMITTER
006512 012777 000340 007024    MOV    #340,@DHTLVL   ;INTERRUPT VECTOR
006520 012777 000013 006766    MOV    #13,@DHSCR     ;SELECT LINE 13

```

```

006526 012777 177777 006770      MOV    #1, @CHBC      ;SET BYTE COUNT TO 1
006534 012700 001000              MOV    #1000, R0
006540 012777 004000 006760      MOV    #4000, @DHBAR ;SET BAR BIT FOR
                                ;LINE 13
006546 052777 020000 006740      BIS    @BIT13, @DHSCR ;SET TRANSMITTER INTERRUPT ENABLE
006554 005067 171216              CLR    PS             ;ALLOW INTERRUPTS
006560 005300                      1#:   DEC    R0             ;DELAY FOR NPR
006562 001376                      BNE    1#
006564                      HLT    1                 ;NO INTERRUPT OCCURED, ERROR
006566 104001                      EMT    1
006566 005777 006722      2#:   TST    @DHSCR      ;VERIFY THAT TRANSMITTER
                                ;DONE IS SET
006572 100401                      BMI    3#
006574                      HLT    2                 ;TRANSMITTER DONE NOT SET, ERROR
006574 104002                      EMT    2
006576 005777 006724      3#:   TST    @DHBAR      ;WAS BAR BIT CLEARED FOR LINE 13
006602 001404                      BEQ    4#
006604 005005                      CLR    R5             ;(R5)=EXPECTED DATA IN
                                ;BUFFER ACTIVE REGISTER, 0
006606 017704 006714      MOV    @DHBAR, R4      ;(R4)=ACTUAL CONTENTS OF
                                ;BUFFER ACTIVE REGISTER
006612                      HLT    0                 ;BUS ACTIVE BIT NOT CLEARED, ERROR
006612 104000                      EMT    0
006614 022777 000001 006700 4#:   CMP    #1, @DHBA      ;WAS BUS ADDRESS INCREMENTED
006622 001405                      BEQ    5#
006624 017704 006672      MOV    @DHBA, R4      ;(R4)=ACTUAL CONTENTS
                                ;OF BUS ADDRESS MEMORY FOR
                                ;LINE 13
006630 012705 000001      MOV    #1, R5         ;(R5)=EXPECTED VALUE OF
                                ;BUS ADDRESS MEMORY FOR
                                ;LINE 13, 1
006634                      HLT    3                 ;BUS ADDRESS NOT UPDATED
006634 104003                      EMT    3
006636 005777 006662      5#:   TST    @DHBC      ;CORRECTLY, ERROR
006642 001416                      BEQ    6#             ;DID BYTE COUNT DECREMENT TO 0
006644 017704 006654      MOV    @DHBC, R4      ;(R4)=ACTUAL VALUE OF BYTE
                                ;COUNT FOR LINE 13
006650 005005                      CLR    R5             ;(R5)=EXPECTED VALUE OF BYTE
                                ;COUNT FOR LINE 13, 0
                                ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR
006652                      HLT    4                 ;
006652 104004                      EMT    4
006654 016777 006660 006654      MOV    DHRLVL, @DHRVEC ;RESTORE TRAPCATCHER
006662 005077 006652                      CLR    @DHRLVL
006666 016777 006652 006646      MOV    DHTLVL, @DHTVEC
006674 005077 006644                      CLR    @DHTLVL
006700 012706 017270      6#:   MOV    #STACK, SP ;RESTORE STACK
006704 104400                      SCOPE ;CHECK FOR ITERATIONS, LOOP
                                LINE=LINE+1
000014 BITX=BITX+BITX
010000 NPRTS \LINE, \BITX

                                ;NPR LOGIC TEST
                                ;SET BYTE COUNT TO 1 FOR LINE 14
                                ;SET BAR BIT FOR LINE 14
                                ;DELAY FOR NPR
                                ;VERIFY THAT BAR BIT FOR LINE 14 CLEARS

```

;VERIFY THAT TRANSMITTER DONE IS SET

```

006706          TS \XN,20.6$
006706 012767 000340 171062 T24:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
006714 012767 000020 006640      MOV    #20,ICOUNT      ;SET UP FOR 20 ITERATIONS
006722 012767 007136 006626      MOV    #6$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
                          .IF NB  <>
                          MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                          .FENDC
                          XN=XN+1
006730 000025          MOV    #BIT11,@DHSCR      ;ISSUE MASTER CLEAR
006736 004767 006400 006556      JSR    PC,CLEAR       ;CLEAR ALL BUS ADDRESS
                          ;AND BYTE COUNT MEMORY LOCATIONS
006742 012777 007024 006572      MOV    #2$,@DHTVEC    ;SET UP TRANSMITTER
006750 012777 000340 006566      MOV    #340,@DHTLVL  ;INTERRUPT VECTOR
006756 012777 000014 006530      MOV    #14,@DHSCR    ;SELECT LINE 14
006764 012777 177777 006532      MOV    #-1,@DHBC     ;SET BYTE COUNT TO 1
006772 012700 001000          MOV    #1000,RO
006776 012777 010000 006522      MOV    #10000,@DHBAR ;SET BAR BIT FOR
                          ;LINE 14
007004 052777 020000 006502      BIS    #BIT13,@DHSCR ;SET TRANSMITTER INTERRUPT ENABLE
007012 005067 170760          CLR    PS            ;ALLOW INTERRUPTS
007016 005300          1$:  DEC    RO            ;DELAY FOR NPR
007020 001376          BNE    1$
007022          HLT    1            ;NO INTERRUPT OCCURED, ERROR
007022 104001          EMT    1
007024 005777 006464          2$:  TST    @DHSCR      ;VERIFY THAT TRANSMITTER
                          ;DONE IS SET
007030 100401          BMI    3$
007032          HLT    2            ;TRANSMITTER DONE NOT SET, ERROR
007032 104002          EMT    2
007034 005777 006466          3$:  TST    @DHBAR      ;WAS BAR BIT CLEARED FOR LINE 14
007040 001404          BEQ    4$
007042 005005          CLR    R5
                          ;(R5)=EXPECTED DATA IN
                          ;BUFFER ACTIVE REGISTER, 0
007044 017704 006456          MOV    @DHBAR,R4      ;(R4)=ACTUAL CONTENTS OF
                          ;BUFFER ACTIVE REGISTER
007050          HLT    0            ;BUS ACTIVE BIT NOT CLEARED, ERROR
007050 104000          EMT    0
007052 022777 000001 006442 4$:  CMP    #1,@DHBA      ;WAS BUS ADDRESS INCREMENTED
007060 001405          BEQ    5$
007062 017704 006434          MOV    @DHBA,R4
                          ;(R4)=ACTUAL CONTENTS
                          ;OF BUS ADDRESS MEMORY FOR
                          ;LINE 14
007066 012705 000001          MOV    #1,R5         ;(R5)=EXPECTED VALUE OF
                          ;BUS ADDRESS MEMORY FOR
                          ;LINE 14, 1
007072          HLT    3            ;BUS ADDRESS NOT UPDATED
007072 104003          EMT    3
007074 005777 006424          5$:  TST    @DHBC      ;CORRECTLY, ERROR
007100 001416          BEQ    6$          ;DID BYTE COUNT DECREMENT TO 0
007102 017704 006416          MOV    @DHBC,R4
                          ;(R4)=ACTUAL VALUE OF BYTE
                          ;COUNT FOR LINE 14
007106 005005          CLR    R5         ;(R5)=EXPECTED VALUE OF BYTE
007110          HLT    4            ;COUNT FOR LINE 14, 0
                          ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR

```

```

007110 104004          EMT      4
007112 016777 006422 006416      MOV     DHRLVL,@DHRVEC ;RESTORE TRAPCATCHER
007120 005077 006414          CLR     @DHRLVL
007124 015777 006414 006410      MOV     DHTLVL,@DHTVEC
007132 005077 006406          CLR     @DHTLVL
007136 012706 012720          6$:    MOV     @STACK,SP ;RESTORE STACK
007142 104400          SCOPE
000015          LINE=LINE+1 ;CHECK FOR ITERATIONS, LOOP
020000          BITX=BITX+BITX
007144          NPRTS  \LINE,\BITX

;NPR LOGIC TEST
;SET BYTE COUNT TO 1 FOR LINE 15
;SET BAR BIT FOR LINE 15
;DELAY FOR NPR
;VERIFY THAT BAR BIT FOR LINE 15 CLEARS
;VERIFY THAT TRANSMITTER DONE IS SET

007144          TS \XN,20,6$
007144 012767 000340 170624      T25:   MOV     @340,PS ;DISABLE ALL INTERRUPTS
007152 012767 000020 006402      MOV     @20,ICOUNT ;SET UP FOR 20 ITERATIONS
007160 012767 007374 006370      MOV     @6$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST

;IF NO <>
MOV     @,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1

007166 000026          MOV     @BIT11,@DHSCR ;ISSUE MASTER CLEAR
007174 004767 006222          JSR     PC,CLEAR ;CLEAR ALL BUS ADDRESS
;AND BYTE COUNT MEMORY LOCATIONS
007200 012777 007262 006334          MOV     @2,@DHTVEC ;SET UP TRANSMITTER
007206 012777 000340 006330          MOV     @340,@DHTLVL ;INTERRUPT VECTOR
007214 012777 000015 006272          MOV     @15,@DHSCR ;SELECT LINE 15
007222 012777 177777 006274          MOV     @-1,@DHBC ;SET BYTE COUNT TO 1
007230 012700 001000          MOV     @1000,R0
007234 012777 020000 006264          MOV     @20000,@DHBAR ;SET BAR BIT FOR
;LINE 15
007242 052777 020000 006244          BIS     @BIT13,@DHSCR ;SET TRANSMITTER INTERRUPT ENABLE
007250 005067 170522          CLR     PS ;ALLOW INTERRUPTS
007254 005300          1$:    DEC     R0 ;DELAY FOR NPR
007256 001376          BNE     1$
007260          HLT     1 ;NO INTERRUPT OCCURED, ERROR
007260 104001          EMT     1
007262 005777 006226          2$:    TST     @DHSCR ;VERIFY THAT TRANSMITTER
;DONE IS SET

007266 100401          BMI     3$
007270          HLT     2 ;TRANSMITTER DONE NOT SET, ERROR
007270 104002          EMT     2
007272 005777 006230          3$:    TST     @DHBAR ;WAS BAR BIT CLEARED FOR LINE 15
007276 001404          BEQ     4$
007300 005005          CLR     R5 ;(R5)=EXPECTED DATA IN
;BUFFER ACTIVE REGISTER, 0
; (R4)=ACTUAL CONTENTS OF
;BUFFER ACTIVE REGISTER
007302 017704 006220          MOV     @DHBAR,R4 ;BUS ACTIVE BIT NOT CLEARED, ERROR

007306          HLT     0
007306 104000          EMT     0
007310 022777 000001 006204          4$:    CMP     @1,@DHBA ;WAS BUS ADDRESS INCREMENTED
007316 001405          BEQ     5$

```



```

007516 104001          EMT      1
007520 005777 005770  2$:   TST      @DHSCR                ;VERIFY THAT TRANSMITTER
                                           ;DONE IS SET
007524 104001          BMI      3$
007526          HLT      2                ;TRANSMITTER DONE NOT SET, ERROR
007526 104002          EMT      2
007530 005777 005772  3$:   TST      @DHBAR                ;WAS BAR BIT CLEARED FOR LINE 16
007534 001404          BEQ      4$
007536 005005          CLR      R5                ;(R5)=EXPECTED DATA IN
                                           ;BUFFER ACTIVE REGISTER, 0
007540 017704 005762          MOV      @DHBAR,R4          ;(R4)=ACTUAL CONTENTS OF
                                           ;BUFFER ACTIVE REGISTER
007544          HLT      0                ;BUS ACTIVE BIT NOT CLEARED, ERROR
007544 104000          EMT      0
007546 022777 000001 005746 4$:  CMP      @1,@DHBA          ;WAS BUS ADDRESS INCREMENTED
007554 001405          BEQ      5$
007556 017704 005740          MOV      @DHBA,R4          ;(R4)=ACTUAL CONTENTS
                                           ;OF BUS ADDRESS MEMORY FOR
                                           ;LINE 16
007562 012705 000001          MOV      @1,R5                ;(R5)=EXPECTED VALUE OF
                                           ;BUS ADDRESS MEMORY FOR
                                           ;LINE 16, 1
007566          HLT      3                ;BUS ADDRESS NOT UPDATED
007566 104003          EMT      3
007570 005777 005730  5$:   TST      @DHBC                ;CORRECTLY, ERROR
007574 001416          BEQ      6$                ;DID BYTE COUNT DECREMENT TO 0
007576 017704 005722          MOV      @DHBC,R4          ;(R4)=ACTUAL VALUE OF BYTE
                                           ;COUNT FOR LINE 16
007602 005005          CLR      R5                ;(R5)=EXPECTED VALUE OF BYTE
                                           ;COUNT FOR LINE 16, 0
007604          HLT      4                ;BYTE COUNT DID NOT DECREMENT TO 0, ERROR
007604 104004          EMT      4
007606 016777 005726 005722          MOV      @DHLVL,@DHRVEC ;RESTORE TRAPCATCHER
007614 005077 005720          CLR      @DHLVL
007620 016777 005720 005714          MOV      @DHTLVL,@DHTVEC
007626 005077 005712          CLR      @DHTLVL
007632 012706 017270  6$:   MOV      @STACK,SP          ;RESTORE STACK
007636 104403          SCOPE
007636 000017          LINE=LINE+1
007636 100000          BITX=BITX+BITX
007640          NPRTS  \LINE,\BITX

                                           ;NPR LOGIC TEST
                                           ;SET BYTE COUNT TO 1 FOR LINE 17
                                           ;SET BAR BIT FOR LINE 17
                                           ;DELAY FOR NPR
                                           ;VERIFY THAT BAR BIT FOR LINE 17 CLEARS
                                           ;VERIFY THAT TRANSMITTER DONE IS SET

007640          TS  \XN,20,6$
007640 012767 000340 170130  T27:  MOV      @340,PS          ;DISABLE ALL INTERRUPTS
007646 012767 000020 005706          MOV      @20,ICOUNT        ;SET UP FOR 20 ITERATIONS
007654 012767 010070 005674          MOV      @6$,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST

                                           .IF NB  <>
                                           MOV      @,FREEZ1          ;SET UP TO LOOP WITH DATA
                                           .ENDC

```



```

22      000000      LINE=0
23      000000      XLINE=LINE
24      000001      BITX=1
25      000001      XBIT=BITX
27      000020      .REPT 20
28      NPRTS2 \LINE,\BITX
29      .MLIST
30      LINE=LINE+1
31      BITX=BITX+BITX
32      .LIST
33      .ENDR
      NPRTS2 \LINE,\BITX
    
```

010076

```

;NPR LOGIC TEST
;SET BYTE COUNT ON ALL LINES TO 1
;SET BAR BIT FOR LINE 0
;VERIFY THAT BYTE COUNT FOR LINE 0 GOES TO 0
;VERIFY THAT BUS ADDRESS FOR LINE 0 IS INCREMENTED
;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
;ARE UNCHANGED
    
```

```

010076      TS \XN,10,8#
010076 012767 000340 167672 T30:  MOV  #340,PS ;DISABLE ALL INTERRUPTS
010104 012767 000010 005450      MOV  #10,ICOUNT ;SET UP FOR 10 ITERATIONS
010112 012767 010244 005436      MOV  #8#,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV  #.FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
      .ENDC
      XN=XN+1
010120 000031 012777 004000 005366      MOV  #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
      ;TO SET ALL TBMT BITS
010126 004767 005270      JSR  PC,CLEAR ;CLEAR ALL BYTE COUNT AND
      ;BUS ADDRESS MEMORY LOCATIONS
010132 004767 005316      JSR  PC,LOAD ;SET ALL BYTE COUNT LOCATIONS TO -1
010136 012777 000001 005362      MOV  #1,@DHBAR ;SET BAR BIT FOR LINE 0
010144 005777 005344 3#:  TST  @DHSCR ;WAIT FOR TRANSMITTER DONE
010150 100375      BPL  3#
010152 012700 000020      MOV  #20,R0 ;SET UP TO CHECK ALL 16 LINES
010156 005077 005332      CLR  @DHSCR ;START AT LINE 0
010162 005001      CLR  R1 ;KEEP TRACK OF LINE NUMBER
010164 012705 177777 4#:  MOV  #-1,R5 ;(R5)=EXPECTED BYTE COUNT,
      ;IF LINE NUMBER NOT = 0
010170 005003      CLR  R3 ;(R3)=EXPECTED BUS ADDRESS,
      ;IF LINE NUMBER NOT = 0
010172 017704 005326      MOV  @DHBC,R4 ;(R4)=ACTUAL BYTE ACOUNT
010176 017702 005320      MOV  @DHBA,R2 ;(R5)=ACTUAL BUS ADDRESS
010202 027727 005306 000000      CMP  @DHSCR,#0 ;IF LINE BEING COMPARED IS LINE 0
010210 001002      BNE  5#
010212 005005      CLR  R5 ;EXPECTED BYTE COUNT=0
010214 005203      INC  R3 ;EXPECTED BUS ADDRESS = 1
010216 020504 5#:  CMP  R5,R4 ;IS BYTE COUNT CORRECT
010220 001401      BEQ  6#
010222      HLT  4 ;BYTE COUNT ERROR
010224      EMT  4
010226 020302 6#:  CMP  R3,R2 ;IS BUS ADDRESS CORRECT
010230 001401      BEQ  7#
      HLT  3 ;BUS ADDRESS ERROR
    
```

```

010230 104003          EMT      3
010232 005277 005256 7#: INC     @DHSCR      ;PREPARE TO CHECK NEXT LINE
010236 005201          INC     R1
010240 005300          DEC     R0      ;CONTINUE IF ALL NOT DONE
010242 001350          BNE     4#
010244 104400          8#: SCOPE      ;CHECK FOR ITERATIONS, LOOP
          000001      LINE=LINE+1
          000002      BITX=BITX+BITX
010246          NPRTS2 \LINE,\BITX

          ;NPR LOGIC TEST
          ;SET BYTE COUNT ON ALL LINES TO 1
          ;SET BAR BIT FOR LINE 1
          ;VERIFY THAT BYTE COUNT FOR LINE 1 GOES TO 0
          ;VERIFY THAT BUS ADDRESS FOR LINE 1 IS INCREMENTED
          ;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
          ;ARE UNCHANGED

010246          TS \XN,10,8#
010246 012767 000340 167522 T31: MOV     @340,PS      ;DISABLE ALL INTERRUPTS
010254 012767 000010 005300      MOV     @10,ICOUNT    ;SET UP FOR 10 ITERATIONS
010262 012767 010414 005266      MOV     @8#,ESCAPE   ;SET UP TO ESCAPE TO NEXT TEST

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

010270 000032          MOV     @BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
          012777 004000 005216      JSR     PC,CLEAR      ;TO SET ALL TBMT BITS
          ;CLEAR ALL BYTE COUNT AND
          ;BUS ADDRESS MEMORY LOCATIONS
010276 004767 005120          JSR     PC,LOAD      ;SET ALL BYTE COUNT LOCATIONS TO -1
          ;SET BAR BIT FOR LINE 1
010302 004767 005146          MOV     @2,@DHBAR    ;WAIT FOR TRANSMITTER DONE
          010306 012777 000002 005212 3#: TST     @DHSCR
          010314 005777 005174          BPL     3#
          010320 100375          MOV     @20,R0      ;SET UP TO CHECK ALL 16 LINES
          010322 012700 000020          CLR     @DHSCR     ;START AT LINE 0
          010326 005077 005162          CLR     R1        ;KEEP TRACK OF LINE NUMBER
          010332 005001          MOV     @-1,R5     ;(R5)=EXPECTED BYTE COUNT,
          010334 012705 177777 4#:      CLR     R3        ;IF LINE NUMBER NOT = 1
          ;(R3)=EXPECTED BUS ADDRESS,
          ;IF LINE NUMBER NOT = 1
          ;(R4)=ACTUAL BYTE ACOUNT
          ;(R5)=ACTUAL BUS ADDRESS
010340 005003          MOV     @DHBC,R4
          010342 017704 005156          MOV     @DHBA,R2
          010344 017702 005150          CMP     @DHSCR,@1  ;IF LINE BEING COMPARED IS LINE 1
          010352 027727 005136 000001 5# BNE     5#
          010360 001002          CLR     R5
          010362 005005          INC     R3        ;EXPECTED BYTE COUNT=0
          010364 005203          INC     R3        ;EXPECTED BUS ADDRESS = 1
          010366 020504          5#: CMP     R5,R4   ;IS BYTE COUNT CORRECT
          010370 001401          BEQ     6#
          010372          HLT     4      ;BYTE COUNT ERROR
          010372 104004          EMT     4
          010374 020302          6#: CMP     R3,R2   ;IS BUS ADDRESS CORRECT
          010376 001401          BEQ     7#
          010400          HLT     3      ;BUS ADDRESS ERROR
          010400 104003          EMT     3
          010402 005277 005106 7#: INC     @DHSCR    ;PREPARE TO CHECK NEXT LINE
          010406 005201          INC     R1

```

```

010410 005300          DEC    R0          ;CONTINUE IF ALL NOT DONE
010412 001350          BNE    4$
010414 104400          8$: SCOPE          ;CHECK FOR ITERATIONS, LOOP
          000002          LINE=LINE+1
          000004          BITX=BITX+BITX
010416          NPRTS2 \LINE,\BITX

          ;NPR LOGIC TEST
          ;SET BYTE COUNT ON ALL LINES TO 1
          ;SET BAR BIT FOR LINE 2
          ;VERIFY THAT BYTE COUNT FOR LINE 2 GOES TO 0
          ;VERIFY THAT BUS ADDRESS FOR LINE 2 IS INCREMENTED
          ;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
          ;ARE UNCHANGED

010416          TS \XN,10,8$
010416 012767 000340 167352 T32:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
010424 012767 000010 005130      MOV    #10,ICOUNT      ;SET UP FOR 10 ITERATIONS
010432 012767 010564 005116      MOV    #8$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST

          .IF NB <>
          MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
          .ENDC
          XN=XN+1
010440 012777 004000 005046      MOV    @BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
          ;TO SET ALL TBMT BITS
010446 004767 004750          JSR    PC,CLEAR        ;CLEAR ALL BYTE COUNT AND
          ;BUS ADDRESS MEMORY LOCATIONS
010452 004767 004776          JSR    PC,LOAD        ;SET ALL BYTE COUNT LOCATIONS TO -1
010456 012777 000004 005042      MOV    #4,@DMBAR      ;SET BAR BIT FOR LINE 2
010464 005777 005024          3$: TST    @DHSCR        ;WAIT FOR TRANSMITTER DONE
010470 100375          BPL    3$
010472 012700 000020          MOV    #20,R0        ;SET UP TO CHECK ALL 16 LINES
010476 005077 005012          CLR    @DHSCR        ;START AT LINE 0
010502 005001          CLR    R1            ;KEEP TRACK OF LINE NUMBER
010504 012705 177777          4$: MOV    #-1,R5      ;(R5)=EXPECTED BYTE COUNT.
          ;IF LINE NUMBER NOT = 2
010510 005003          CLR    R3            ;(R3)=EXPECTED BUS ADDRESS.
          ;IF LINE NUMBER NOT = 2
010512 017704 005006          MOV    @DMBC,R4      ;(R4)=ACTUAL BYTE ACOUNT
010516 017702 005000          MOV    @DMBA,R2      ;(R5)=ACTUAL BUS ADDRESS
010522 027727 004766 000002      CMP    @DHSCR,#2     ;IF LINE BEING COMPARED IS LINE 2
010530 001002          BNE    5$
010532 005005          CLR    R5            ;EXPECTED BYTE COUNT=0
010534 005203          INC    R3            ;EXPECTED BUS ADDRESS = 1
010536 020504          5$: CMP    R5,R4      ;IS BYTE COUNT CORRECT
010540 001401          BEQ    6$
010542          HLT    4          ;BYTE COUNT ERROR
010542 104004          EMT    4
010544 020302          6$: CMP    R3,R2      ;IS BUS ADDRESS CORRECT
010546 001401          BEQ    7$
010550          HLT    3          ;BUS ADDRESS ERROR
010550 104003          EMT    3
010552 005277 004736          7$: INC    @DHSCR      ;PREPARE TO CHECK NEXT LINE
010556 005201          INC    R1
010560 005300          DEC    R0          ;CONTINUE IF ALL NOT DONE
010562 001350          BNE    4$
010564 104400          8$: SCOPE          ;CHECK FOR ITERATIONS, LOOP

```

```

000003          LINE=LINE+1
000010          BITX=BITX+BITX
010566          NPRTS2 \LINE,\BITX

;NPR LOGIC TEST
;SET BYTE COUNT ON ALL LINES TO 1
;SET BAR BIT FOR LINE 3
;VERIFY THAT BYTE COUNT FOR LINE 3 GOES TO 0
;VERIFY THAT BUS ADDRESS FOR LINE 3 IS INCREMENTED
;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
;ARE UNCHANGED

010566          TS \XN,10,8#
010566 012767 000340 167202 T33:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
010574 012767 000010 004760      MOV    #10,ICOUNT      ;SET UP FOR 10 ITERATIONS
010602 012767 010734 004746      MOV    #8#,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
                          .TF NB  <>
                          MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                          .ENDC
                          XN=XN+1
010610 012777 004000 004676      MOV    #BIT11,0DHSCR  ;MASTER CLEAR INTERFACE
                          ;TO SET ALL TBMT BITS
010616 004767 004600              JSR    PC,CLEAR      ;CLEAR ALL BYTE COUNT AND
                          ;BUS ADDRESS MEMORY LOCATIONS
010622 004767 004626              JSR    PC,LOAD       ;SET ALL BYTE COUNT LOCATIONS TO -1
010626 012777 000010 004672      MOV    #10,0DHBAR    ;SET BAR BIT FOR LINE 3
010634 005777 004654      3#:  TST    0DHSCR        ;WAIT FOR TRANSMITTER DONE
010640 100375              BPL    3#
010642 012700 000020              MOV    #20,R0        ;SET UP TO CHECK ALL 16 LINES
010646 005077 004642              CLR    0DHSCR        ;START AT LINE 0
010652 005001              CLR    R1            ;KEEP TRACK OF LINE NUMBER
010654 012705 177777      4#:  MOV    #-1,R5      ;(R5)=EXPECTED BYTE COUNT.
                          ;IF LINE NUMBER NOT = 3
010660 005003              CLR    R3            ;(R3)=EXPECTED BUS ADDRESS.
                          ;IF LINE NUMBER NOT = 3
010662 017704 004636              MOV    0DHBC,R4      ;(R4)=ACTUAL BYTE ACOUNT
010666 017702 004630              MOV    0DHBA,R2      ;(R5)=ACTUAL BUS ADDRESS
010672 027727 004616 000003      CMP    0DHSCR,#3    ;IF LINE BEING COMPARED IS LINE 3
010700 001002              BNE    5#
010702 005005              CLR    R5            ;EXPECTED BYTE COUNT=0
010704 005203              INC    R3            ;EXPECTED BUS ADDRESS = 1
010706 020504      5#:  CMP    R5,R4      ;IS BYTE COUNT CORRECT
010710 001401              BEQ    6#
010712              HLT    4
                          ;BYTE COUNT ERROR
010712 104004              EMT    4
010714 020302      6#:  CMP    R3,R2      ;IS BUS ADDRESS CORRECT
010716 001401              BEQ    7#
010720              HLT    3
                          ;BUS ADDRESS ERROR
010720 104003              EMT    3
010722 005277 004566      7#:  INC    0DHSCR    ;PREPARE TO CHECK NEXT LINE
010726 005201              INC    R1
010730 005300              DEC    R0
010732 001350              BNE    4#
                          ;CONTINUE IF ALL NOT DONE
010734 104400      8#:  SCOPE          ;CHECK FOR ITERATIONS, LOOP
                          000004
                          000020
010736          BITX=BITX+BITX
          NPRTS2 \LINE,\BITX

```

```

;NPR LGGIC TEST
;SET BYTE COUNT ON ALL LINES TO 1
;SET BAR BIT FOR LINE 4
;VERIFY THAT BYTE COUNT FOR LINE 4 GOES TO 0
;VERIFY THAT BUS ADDRESS FOR LINE 4 IS INCREMENTED
;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
;ARE UNCHANGED

010736          TS \XN,10,8#
010736 012767 000340 167032 T34:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
010744 012767 000010 004610      MOV    #10,ICOUNT       ;SET UP FOR 10 ITERATIONS
010752 012767 011104 004576      MOV    #8,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST

                .IF NB <>
                MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                .ENDC
                XN=XN+1
010760 000035 012777 004000 004526  MOV    #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
                                ;TO SET ALL TBMT BITS
010766 004767 004430          JSR    PC,CLEAR        ;CLEAR ALL BYTE COUNT AND
                                ;BUS ADDRESS MEMORY LOCATIONS
010772 004767 004456          JSR    PC,LOAD         ;SLT ALL BYTE COUNT LOCATIONS TO -1
010776 012777 000020 004522      MOV    #20,@DHBAR     ;SET BAR BIT FOR LINE 4
011004 005777 004504          3#:  TST    @DHSCR        ;WAIT FOR TRANSMITTER DONE
011010 100375          BPL    3#
011012 012700 000020          MOV    #20,R0         ;SET UP TO CHECK ALL 16 LINES
011016 005077 004472          CLR    @DHSCR        ;START AT LINE 0
011022 005001          CLR    R1            ;KEEP TRACK OF LINE NUMBER
011024 012705 177777          4#:  MOV    #-1,R5      ;(R5)=EXPECTED BYTE COUNT,
                                ;IF LINE NUMBER NOT = 4
011030 005003          CLR    R3            ;(R3)=EXPECTED BUS ADDRESS,
                                ;IF LINE NUMBER NOT = 4
011032 017704 004466          MOV    @DHBC,R4       ;(R4)=ACTUAL BYTE ACOUNT
011036 017702 004460          MOV    @DHBA,R2       ;(R5)=ACTUAL BUS ADDRESS
011042 027727 004446 0000C4      CMP    @DHSCR,#4     ;IF LINE BEING COMPARED IS LINE 4
011050 001002          BNE    5#
011052 005005          CLR    R5            ;EXPECTED BYTE COUNT=0
011054 005203          INC    R3            ;EXPECTED BUS ADDRESS = 1
011056 020504          5#:  CMP    R5,R4        ;IS BYTE COUNT CORRECT
011060 001401          BEQ    6#
011062          HLT    4            ;BYTE COUNT ERROR
011062 104004          EMT    4
011064 020302          6#:  CMP    R3,R2        ;IS BUS ADDRESS CORRECT
011066 001401          BEQ    7#
011070          HLT    3            ;BUS ADDRESS ERROR
011070 104003          EMT    3
011072 005277 004416          7#:  INC    @DHSCR        ;PREPARE TO CHECK NEXT LINE
011076 005201          INC    R1
011100 005300          DEC    R0
011102 001350          BNE    4#            ;CONTINUE IF ALL NOT DONE
011104 104400          8#:  SCOPE          ;CHECK FOR ITERATIONS, LOOP
                LINE=LINE+1
                BITX=BITX+BITX
011106          NPRTS2 \LINE,\BITX

;NPR LOGIC TEST
;SET BYTE COUNT ON ALL LINES TO 1

```

```

;SET BAR BIT FOR LINE 5
;VERIFY THAT BYTE COUNT FOR LINE 5 GOES TO 0
;VERIFY THAT BUS ADDRESS FOR LINE 5 IS INCREMENTED
;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
;ARE UNCHANGED

011106          TS \XN,10,8#
011106 012767 000340 166662 T35: MOV    #340,PS          ;DISABLE ALL INTERRUPTS
011114 012767 000010 004440      MOV    #10,ICOUNT      ;SET UP FOR 10 ITERATIONS
011122 012767 011254 004426      MOV    #8#,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST

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

011130 012777 004000 004356      MOV    #BIT11,@DHSCR  ;MASTER CLEAR INTERFACE
                                ;TO SET ALL TBMT BITS
011136 004767 004250              JSR    PC,CLEAR      ;CLEAR ALL BYTE COUNT AND
                                ;BUS ADDRESS MEMORY LOCATIONS
011142 004767 004306              JSR    PC,LOAD      ;SET ALL BYTE COUNT LOCATIONS TO -1
011146 012777 000040 004352      MOV    #40,@DHBAR    ;SET BAR BIT FOR LINE 5
011154 005777 004334              3#: TST    @DHSCR      ;WAIT FOR TRANSMITTER DONE
011160 100375                    BPL    3#
011162 012700 000020              MOV    #20,R0        ;SET UP TO CHECK ALL 16 LINES
011166 005077 004322              CLR    @DHSCR        ;START AT LINE 0
011172 005001                    CLR    R1            ;KEEP TRACK OF LINE NUMBER
011174 012705 177777              4#: MOV    #-1,R5     ;(R5)=EXPECTED BYTE COUNT,
                                ;IF LINE NUMBER NOT = 5
                                ;(R3)=EXPECTED BUS ADDRESS,
                                ;IF LINE NUMBER NOT = 5
                                ;(R4)=ACTUAL BYTE ACOUNT
                                ;(R5)=ACTUAL BUS ADDRESS
011200 005003                    CLR    R3
                                ;IF LINE BEING COMPARED IS LINE 5
011202 017704 004316              MOV    @DMBC,R4
011206 017702 004310              MOV    @DMBA,R2
011212 027727 004276 000005      CMP    @DHSCR,#5
011220 001002                    BNE    5#
011222 005005                    CLR    R5            ;EXPECTED BYTE COUNT=0
011224 005203                    INC    R3            ;EXPECTED BUS ADDRESS = 1
011226 020504                    5#: CMP    R5,R4        ;IS BYTE COUNT CORRECT
011230 001401                    BEQ    6#
011232                    HLT    4            ;BYTE COUNT ERROR
011232 104004                    EMT    4
011234 020302                    6#: CMP    R3,R2        ;IS BUS ADDRESS CORRECT
011236 001401                    BEQ    7#
011240                    HLT    3            ;BUS ADDRESS ERROR
011240 104003                    EMT    3
011242 005277 004246              7#: INC    @DHSCR      ;PREPARE TO CHECK NEXT LINE
011246 005201                    INC    R1
011250 005300                    DEC    R0            ;CONTINUE IF ALL NOT DONE
011252 001350                    BNE    4#
011254 104400                    8#: SCOPE          ;CHECK FOR ITERATIONS, LOOP
                000006          LINE=LINE+1
                000100          BITX=BITX+BITX
011256          NPRTS2 \LINE,\BITX

;NPR LOGIC TEST
;SET BYTE COUNT ON ALL LINES TO 1
;SFT BAR BIT FOR LINE 6
;VERIFY THAT BYTE COUNT FOR LINE 6 GOES TO 0
;VERIFY THAT BUS ADDRESS FOR LINE 6 IS INCREMENTED

```



```

;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
;ARE UNCHANGED

011256 TS \XN,10,8#
011256 012767 000340 166512 T36: MOV #340,PS ;DISABLE ALL INTERRUPTS
011264 012767 000010 004270 MOV #10,ICOUNT ;SET UP FOR 10 ITERATIONS
011272 012767 011424 004256 MOV #8,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST

;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3

.ENDC
XN=XN+1

011300 000037 012777 004000 004206 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
;TO SET ALL TBMT BITS
011306 004767 004110 JSR PC,CLEAR ;CLEAR ALL BYTE COUNT AND
;BUS ADDRESS MEMORY LOCATIONS
011312 004767 004136 JSR PC,LOAD ;SET ALL BYTE COUNT LOCATIONS TO -1
011316 012777 000100 004202 MOV #100,@DHBAR ;SET BAR BIT FOR LINE 6
011324 005777 004164 3#: TST @DHSCR ;WAIT FOR TRANSMITTER DONE
011330 100375 BPL 3#
011332 012700 000020 MOV #20,R0 ;SET UP TO CHECK ALL 16 LINES
011336 005077 004152 CLR @DHSCR ;START AT LINE 0
011342 005001 CLR R1 ;KEEP TRACK OF LINE NUMBER
011344 012705 177777 4#: MOV #-1,R5 ;(R5)=EXPECTED BYTE COUNT,
;IF LINE NUMBER NOT = 6
; (R3)=EXPECTED BUS ADDRESS,
;IF LINE NUMBER NOT = 6
; (R4)=ACTUAL BYTE ACOUNT
; (R5)=ACTUAL BUS ADDRESS
011350 005003 CLR R3 ;IF LINE BEING COMPARED IS LINE 6

011352 017704 004146 MCV @DHBC,R4
011356 017702 004140 MOV @DHBA,R2
011362 027727 004126 000006 CMP @DHSCR,#6
011370 001002 BNE 5#
011372 005005 CLR R5 ;EXPECTED BYTE COUNT=0
011374 005203 INC R3 ;EXPECTED BUS ADDRESS = 1
011376 020504 5#: CMP R5,R4 ;IS BYTE COUNT CORRECT
011400 001401 BEQ 6#
011402 HLT 4 ;BYTE COUNT ERROR
011402 104004 EMT 4
011404 020302 6#: CMP R3,R2 ;IS BUS ADDRESS CORRECT
011406 001401 BEQ 7#
011410 HLT 3 ;BUS ADDRESS ERROR
011410 104003 EMT 3
011412 005277 004076 7#: INC @DHSCR ;PREPARE TO CHECK NEXT LINE
011416 005201 INC R1
011420 005300 DEC R0 ;CONTINUE IF ALL NOT DONE
011422 001350 BNE 4#
011424 104400 8#: SCOPE ;CHECK FOR ITERATIONS, LOOP
000007 LINE=LINE+1
000200 BITX=BITX+BITX
011426 NPRTS2 \LINE,\BITX

;NPR LOGIC TEST
;SET BYTE COUNT ON ALL LINES TO 1
;SET BAR BIT FOR LINE 7
;VERIFY THAT BYTE COUNT FOR LINE 7 GOES TO 0
;VERIFY THAT BUS ADDRESS FOR LINE 7 IS INCREMENTED
;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
;ARE UNCHANGED

```

```

011426          TS \XN,10,8#
011426 012767 000340 166342 T37:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
011434 012767 000010 004120      MOV    #10,ICOUNT      ;SET UP FOR 10 ITERATIONS
011442 012767 011574 004106      MOV    #8#,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                        MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                        .ENDC
011450 000040          XN=XN+1
011450 012777 004000 004036      MOV    #BIT11,@DHSCR  ;MASTER CLEAR INTERFACE
                                ;TO SET ALL TBMT BITS
011456 004767 003740          JSR    PC,CLEAR        ;CLEAR ALL BYTE COUNT AND
                                ;BUS ADDRESS MEMORY LOCATIONS
011462 004767 003766          JSR    PC,LOAD         ;SET ALL BYTE COUNT LOCATIONS TO -1
011466 012777 000200 004032      MOV    #200,@DHBAR    ;SET BAR BIT FOR LINE 7
011474 005777 004014 3#:      TST    @DHSCR         ;WAIT FOR TRANSMITTER DONE
011500 100375          BPL    3#
011502 012700 000020          MOV    #20,R0         ;SET UP TO CHECK ALL 16 LINES
011506 005077 004002          CLR    @DHSCR        ;START AT LINE 0
011512 005001          CLR    R1            ;KEEP TRACK OF LINE NUMBER
011514 012705 177777 4#:      MOV    #-1,R5          ;(R5)=EXPECTED BYTE COUNT,
                                ;IF LINE NUMBER NOT = 7
011520 005003          CLR    R3            ;(R3)=EXPECTED BUS ADDRESS,
                                ;IF LINE NUMBER NOT = 7
011522 017704 003776          MOV    @DHBC,R4       ;(R4)=ACTUAL BYTE ACOUNT
011526 017702 003770          MOV    @DHBA,R2       ;(R5)=ACTUAL BUS ADDRESS
011532 027727 003756 000007      CMP    @DHSCR,#7     ;IF LINE BEING COMPARED IS LINE 7
011540 001002          BNE    5#
011542 005005          CLR    R5            ;EXPECTED BYTE COUNT=0
011544 005203          INC    R3           ;EXPECTED BUS ADDRESS = 1
011546 020504 5#:      CMP    R5,R4          ;IS BYTE COUNT CORRECT
011550 001401          BEQ    6#
011552          HLT    4            ;BYTE COUNT ERROR
011552 104004          EMT    4
011554 020302 6#:      CMP    R3,R2          ;IS BUS ADDRESS CORRECT
011556 001401          BEQ    7#
011560          HLT    3            ;BUS ADDRESS ERROR
011560 104003          EMT    3
011562 005277 003726 7#:      INC    @DHSCR        ;PREPARE TO CHECK NEXT LINE
011566 005201          INC    R1
011570 005300          DEC    R0
011572 001350          BNE    4#            ;CONTINUE IF ALL NOT DONE
011574 104400 8#:      SCOPE          ;CHECK FOR ITERATIONS, LOOP
                        LINE=LINE+1
011576          BITX=BITX+BITX
                        NPRTS2 \LINE,\BITX

                        ;NPR LOGIC TEST
                        ;SET BYTE COUNT ON ALL LINES TO 1
                        ;SET BAR BIT FOR LINE 10
                        ;VERIFY THAT BYTE COUNT FOR LINE 10 GOES TO 0
                        ;VERIFY THAT BUS ADDRESS FOR LINE 10 IS INCREMENTED
                        ;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
                        ;ARE UNCHANGED

011576          TS \XN,10,8#
011576 012767 000340 166172 T40:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
011604 012767 000010 003750      MOV    #10,ICOUNT      ;SET UP FOR 10 ITERATIONS

```

```

011612 012767 011744 003736      MOV    #8$,ESCAPE          ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
      .ENDC
      XN=XN+1
011620 000041 012777 004000 003666      MOV    #BIT11,#DHSCR      ;MASTER CLEAR INTERFACE
      ;TO SET ALL TBMT BITS
011626 004767 003570                JSR    PC,CLEAR          ;CLEAR ALL BYTE COUNT AND
      ;BUS ADDRESS MEMORY LOCATIONS
011632 004767 003616                JSR    PC,LOAD          ;SET ALL BYTE COUNT LOCATIONS TO -1
011636 012777 000400 003662      MOV    #400,#DHMBAR      ;SET BAR BIT FOR LINE 10
011644 005777 003644      3$:   TST    #DHSCR          ;WAIT FOR TRANSMITTER DONE
011650 100375                BPL    3$
011652 012700 000020                MOV    #20,R0          ;SET UP TO CHECK ALL 16 LINES
011656 005077 003632      CLR    #DHSCR          ;START AT LINE 0
011662 005001                CLR    R1          ;KEEP TRACK OF LINE NUMBER
011664 012705 177777      4$:   MOV    #-1,R5          ;(R5)=EXPECTED BYTE COUNT,
      ;IF LINE NUMBER NOT = 10
      ;(R3)=EXPECTED BUS ADDRESS,
      ;IF LINE NUMBER NOT = 10
011670 005003                CLR    R3          ;(R4)=ACTUAL BYTE ACOUNT
      ;(R5)=ACTUAL BUS ADDRESS
011672 017704 003626      MOV    #DHBC,R4
011676 017702 003620      MOV    #DHBA,R2
011702 027727 003606 000010      CMP    #DHSCR,#10      ;IF LINE BEING COMPARED IS LINE 10
011710 001002                BNE    5$
011712 005005                CLR    R5          ;EXPECTED BYTE COUNT=0
011714 005203                INC    R3          ;EXPECTED BUS ADDRESS = 1
011716 020504      5$:   CMP    R5,R4          ;IS BYTE COUNT CORRECT
011720 001401                BEQ    6$
011722                HLT    4          ;BYTE COUNT ERROR
011722 104004                EMT    4
011724 020302      6$:   CMP    R3,R2          ;IS BUS ADDRESS CORRECT
011726 001401                BEQ    7$
011730                HLT    3          ;BUS ADDRESS ERROR
011730 104003                EMT    3
011732 005277 003556      7$:   INC    #DHSCR          ;PREPARE TO CHECK NEXT LINE
011736 005201                INC    R1
011740 005300                DEC    R0          ;CONTINUE IF ALL NOT DONE
011742 001350                BNE    4$
011744 104400      8$:   SCOPE          ;CHECK FOR ITERATION'S. LOOP
      LINE=LINE+1
011746 000011      BITX=BITX+BITX
      001000      NPRTS2 \LINE,\BITX
      ;NPR LOGIC TEST
      ;SFT BYTE COUNT ON ALL LINES TO 1
      ;SET BAR BIT FOR LINE 11
      ;VERIFY THAT BYTE COUNT FOR LINE 11 GOES TO 0
      ;VERIFY THAT BUS ADDRESS FOR LINE 11 IS INCREMENTED
      ;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
      ;ARE UNCHANGED

011746      TS \XN,10,8$
011746 012767 000340 166022      T41:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
011754 012767 000010 003600      MOV    #10,ICOUNT      ;SET UP FOR 10 ITERATIONS
011762 012767 012114 003566      MOV    #8$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3

```

```

      .ENDC
      XN=XN+1
011770 000042 004000 003516      MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
011776 004767 003420      JSR      PC,CLEAR          ;TO SET ALL TBMT BITS
012002 004767 003446      JSR      PC,LOAD          ;CLEAR ALL BYTE COUNT AND
012006 012777 001000 003512      MOV      #1000,@DHBAR     ;BUS ADDRESS MEMORY LOCATIONS
012014 005777 003474      3$:      FST      @DHSCR      ;SET ALL BYTE COUNT LOCATIONS TO -1
012020 100375      BPL      3$              ;SET BAR BIT FOR LINE 11
012022 012700 000020      MOV      #20,R0          ;WAIT FOR TRANSMITTER DONE
012026 005077 003462      CLR      @DHSCR          ;SET UP TO CHECK ALL 16 LINES
012032 005001      CLR      R1              ;START AT LINE 0
012034 012705 177777      4$:      MOV      #-1,R5      ;KEEP TRACK OF LINE NUMBER
012040 005003      CLR      R3              ;(R5)=EXPECTED BYTE COUNT.
012042 017704 003456      MOV      @DHBC,R4        ;IF LINE NUMBER NOT = 11
012046 017702 003450      MOV      @DHBA,R2        ;(R3)=EXPECTED BUS ADDRESS.
012052 027727 003436 000C11      CMP      @DHSCR,#11      ;IF LINE NUMBER NOT = 11
012060 001002      BNE      5$              ;(R4)=ACTUAL BYTE ACOUNT
012062 005005      CLR      R5              ;(R5)=ACTUAL BUS ADDRESS
012064 005203      INC      R3              ;IF LINE BEING COMPARED IS LINE 11
012066 020504      5$:      CMP      R5,R4          ;EXPECTED BYTE COUNT=0
012070 001401      BEQ      6$              ;EXPECTED BUS ADDRESS = 1
012072      HLT      4              ;IS BYTE COUNT CORRECT
012074 020302      6$:      CMP      R3,R2          ;BYTE COUNT ERROR
012076 001401      BEQ      7$              ;IS BUS ADDRESS CORRECT
012100      HLT      3              ;BUS ADDRESS ERROR
012100 104003      FMT      3
012102 005277 003406      7$:      INC      @DHSCR      ;PREPARE TO CHECK NEXT LINE
012106 005201      INC      R1
012110 005300      DEC      R0              ;CONTINUE IF ALL NOT DONE
012112 001350      BNE      4$
012114 104400      8$:      SCOPE          ;CHECK FOR ITERATIONS, LOOP
      000012      LINE=LINE+1
      002000      BITX=BITX+BITX
012116      NPRTS2 \LINE,\BITX

      ;NPR LOGIC TEST
      ;SET BYTE COUNT ON ALL LINES TO 1
      ;SET BAR BIT FOR LINE 12
      ;VERIFY THAT BYTE COUNT FOR LINE 12 GOES TO 0
      ;VERIFY THAT BUS ADDRESS FOR LINE 12 IS INCREMENTED
      ;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
      ;ARE UNCHANGED

012116      TS \XN,10,8$
012116 012767 000340 165652      T42:     MOV      #340,PS      ;DISABLE ALL INTERRUPTS
012124 012767 000010 003430      MOV      #10,ICOUNT     ;SET UP FOR 10 ITERATIONS
012132 012767 012264 003416      MOV      #8$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB      <>
      MOV      #,FREEZ1    ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
012140 000043 004000 003346      MOV      #BIT11,@DHSCR  ;MASTER CLEAR INTERFACE

```



```

012322 004767 003126          JSR    PC,LOAD          ;SET ALL BYTE COUNT LOCATIONS TO 1
012326 012777 004000 003172  MOV    #4000, @DHBAR      ;SET BAR BIT FOR LINE 13
012334 005777 003154          3$:   TST    @DHSCR          ;WAIT FOR TRANSMITTER DONE
012340 100375                   BPL    3$
012342 012700 000020          MCV    #20, R0           ;SET UP TO CHECK ALL 16 LINES
012346 005077 003142          CLR    @DHSCR           ;START AT LINE 0
012352 005001                   CLR    R1                ;KEEP TRACK OF LINE NUMBER
012354 012705 177777          4$:   MOV    #-1, R5       ;(R5)=EXPECTED BYTE COUNT,
                                ;IF LINE NUMBER NOT = 13
                                ;(R3)=EXPECTED BUS ADDRESS,
                                ;IF LINE NUMBER NOT = 13
                                ;(R4)=ACTUAL BYTE ACOUNT
                                ;(R5)=ACTUAL BUS ADDRESS
012360 005003                   CLR    R3                ;IF LINE BEING COMPARED IS LINE 13
012362 017704 003136          MOV    @DHBC, R4
012366 017702 003130          MOV    @DHBA, R2
012372 027727 003116 000013  CMP    @DHSCR, #13
012400 001002                   BNE    5$
012402 005005                   CLR    R5                ;EXPECTED BYTE COUNT=0
012404 005203                   INC    R3                ;EXPECTED BUS ADDRESS = 1
012406 020504          5$:   CMP    R5, R4           ;IS BYTE COUNT CORRECT
012410 001401                   BEQ    6$
012412                   HLT    4
012412 104004                   EMT    4                ;BYTE COUNT ERROR
012414 020302          6$:   CMP    R3, R2           ;IS BUS ADDRESS CORRECT
012416 001401                   BEQ    7$
012420                   HLT    3
012420 104003                   EMT    3                ;BUS ADDRESS ERROR
012422 005277 003066          7$:   INC    @DHSCR         ;PREPARE TO CHECK NEXT LINE
012426 005201                   INC    R1
012430 005300                   DEC    R0                ;CONTINUE IF ALL NOT DONE
012432 001350                   BNE    4$
012434 104400          8$:   SCOPE                   ;CHECK FOR ITERATIONS, LOOP
                                LINE=LINE+1
012436                                BITX=BITX+BITX
                                NPRTS2 \LINE, \BITX
                                ;NPR LOGIC TEST
                                ;SET BYTE COUNT ON ALL LINES TO 1
                                ;SET BAR BIT FOR LINE 14
                                ;VERIFY THAT BYTE COUNT FOR LINE 14 GOES TO 0
                                ;VERIFY THAT BUS ADDRESS FOR LINE 14 IS INCREMENTED
                                ;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
                                ;ARE UNCHANGED

012436 TS \XN, 10, 8$
012436 012767 000340 165332 T44:  MOV    #340, #S          ;DISABLE ALL INTERRUPTS
012444 012767 000010 003110          MOV    #10, ICOUNT      ;SET UP FOR 10 ITERATIONS
012452 012767 012604 003076          MOV    #8$, ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB
                                <>
                                MOV    #, FREEZ1          ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC
                                XN=XN+1
012460 000045                   MOV    #BIT11, @DHSCR     ;MASTER CLEAR INTERFACE
012466 004767 002730          JSR    PC, CLEAR         ;TO SET ALL TBMT BITS
                                ;CLEAR ALL BYTE COUNT AND
                                ;BUS ADDRESS MEMORY LOCATIONS
012472 004767 002756          JSR    PC, LOAD          ;SET ALL BYTE COUNT LOCATIONS TO -1
012476 012777 010000 003022  MOV    #10000, @DHBAR    ;SET BAR BIT FOR LINE 14
012504 005777 003004          3$:   TST    @DHSCR          ;WAIT FOR TRANSMITTER DONE

```

```

012510 100375          BPL      3#
012512 012700 000020  MOV      #20,R0          ;SET UP TO CHECK ALL 16 LINES
012515 005077 002772  CLR      @DHSCR         ;START AT LINE 0
012522 005001          CLR      R1              ;KEEP TRACK OF LINE NUMBER
012524 012705 177777  4# : MOV      #-1,R5         ;(R5)=EXPECTED BYTE COUNT,
                                ;IF LINE NUMBER NOT = 14
012530 005003          CLR      R3              ;(R3)=EXPECTED BUS ADDRESS,
                                ;IF LINE NUMBER NOT = 14
012532 017704 002766  MOV      @DH3C,R4         ;(R4)=ACTUAL BYTE ACOUNT
012536 017702 002760  MOV      @DH8A,R2         ;(R5)=ACTUAL BUS ADDRESS
012542 027727 002746 000014  CMP      @DHSCR,#14      ;IF LINE BEING COMPARED IS LINE 14
012550 001002          BNE      5#
012552 005005          CLR      R5              ;EXPECTED BYTE COUNT=0
012554 005203          INC      R3              ;EXPECTED BUS ADDRESS = 1
012556 020504 5# : CMP      R5,R4          ;IS BYTE COUNT CORRECT
012560 001401          BEQ      6#
012562          HLT      4              ;BYTE COUNT ERROR
012562 104004          EMT      4
012564 020302 6# : CMP      R3,R2          ;IS BUS ADDRESS CORRECT
012566 001401          BEQ      7#
012570          HLT      3              ;BUS ADDRESS ERROR
012570 104003          EMT      3
012572 005277 002716 7# : INC      @DHSCR         ;PREPARE TO CHECK NEXT LINE
012576 005201          INC      R1
012600 005300          DEC      R0              ;CONTINUE IF ALL NOT DONE
012602 001350          BNE      4#
012604 104400 8# : SCOPE          ;CHECK FOR ITERATIONS, LOOP
          000015          LINE=LINE+1
          020000          BITX=BITX+BITX
012606          NPRTS2 \LINE,\BITX

          ;NPR LOGIC TEST
          ;SET BYTE COUNT ON ALL LINES TO 1
          ;SET BAR BIT FOR LINE 15
          ;VERIFY THAT BYTE COUNT FOR LINE 15 GOES TO 0
          ;VERIFY THAT BUS ADDRESS FOR LINE 15 IS INCREMENTED
          ;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
          ;ARE UNCHANGED

012606          TS \XN,10,8#
012606 012767 000340 165162 T45: MOV      #340,PS          ;DISABLE ALL INTERRUPTS
012614 012767 000010 002740  MOV      #10,ICOUNT      ;SET UP FOR 10 ITERATIONS
012622 012767 012754 002726  MOV      #8#,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
          .IF NB <>
          MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA      , 3
          .ENDC
          XN=XN+1
012630 000046          MOV      @BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
012636 004767 002560  JSR      PC,CLEAR        ;TO SET ALL TBMT BITS
                                ;CLEAR ALL BYTE COUNT AND
                                ;BUS ADDRESS MEMORY LOCATIONS
012642 004767 002606          JSR      PC,LOAD          ;SET ALL BYTE COUNT LOCATIONS TO -1
012646 012777 020000 002652  MOV      #20000,@DHBAR    ;SET BAR BIT FOR LINE 15
012654 005777 002634 3# : TST      @DHSCR         ;WAIT FOR TRANSMITTER DONE
012660 100375          BPL      3#
012662 012700 000020  MOV      #20,R0          ;SET UP TO CHECK ALL 16 LINES
012666 005077 002622  CLR      @DHSCR         ;START AT LINE 0

```



```

013050 005003          CLR      R3          ;(R3)=EXPECTED BUS ADDRESS,
                                ;IF LINE NUMBER NOT = 16
013052 017704 002446    MOV      @DMBC,R4      ;(R4)=ACTUAL BYTE ACOUNT
013056 017702 002440    MOV      @DMBA,R2      ;(R5)=ACTUAL BUS ADDRESS
013062 027727 002426 000016  CMP      @DMSCR,#16    ;IF LINE BEING COMPARED IS LINE 16
013070 001002          BNE      5#
013072 005005          CLR      R5          ;EXPECTED BYTE COUNT=0
013074 005203          INC      R3          ;EXPECTED BUS ADDRESS = 1
013076 020504          5#:    CMP      R5,R4      ;IS BYTE COUNT CORRECT
013100 001401          BEQ      6#
013102          HLT      4          ;BYTE COUNT ERROR
013102 104004          EMT      4
013104 020302          6#:    CMP      R3,R2      ;IS BUS ADDRESS CORRECT
013106 001401          BEQ      7#
013110          HLT      3          ;BUS ADDRESS ERROR
013110 104003          EMT      3
013112 005277 002376    7#:    INC      @DMSCR      ;PREPARE TO CHECK NEXT LINE
013116 005201          INC      R1
013120 005300          DEC      R0
013122 001350          BNE      8#          ;CONTINUE IF ALL NOT DONE
013124 104400          8#:    SCOPE          ;CHECK FOR ITERATIONS, LOOP
                                LINE=LINE+1
013126          BITX=BITX+BITX
                                NPRTS2 \LINE,\BITX

                                ;NPR LOGIC TEST
                                ;SET BYTE COUNT ON ALL LINES TO 1
                                ;SET BAR BIT FOR LINE 17
                                ;VERIFY THAT BYTE COUNT FOR LINE 17 GOES TO 0
                                ;VERIFY THAT BUS ADDRESS FOR LINE 17 IS INCREMENTED
                                ;VERIFY THAT BYTE COUNTS AND BUS ADDRESSES FOR ALL OTHER LINES
                                ;ARE UNCHANGED

013126          TS \XN,10,8#
013126 012767 000340 164642 T47:    MOV      #340,PS      ;DISABLE ALL INTERRUPTS
013134 012767 000010 002420    MOV      #10,ICOUNT   ;SET UP FOR 10 ITERATIONS
013142 012767 013274 002406    MOV      #6#,ESCAPE   ;SET UP TO ESCAPE TO NEXT TEST

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

013150 012777 004000 002336    MOV      @BIT11,@DMSCR ;MASTER CLEAR INTERFACE
                                ;TO SET ALL TBMT BITS
013156 004767 002240          JSR      PC,CLEAR     ;CLEAR ALL BYTE COUNT AND
                                ;BUS ADDRESS MEMORY LOCATIONS
013162 004767 002266          JSR      PC,LOAD      ;SET ALL BYTE COUNT LOCATIONS TO -1
013166 012777 100000 002332    MOV      #100000,@DMBAR ;SET BAR BIT FOR LINE 17
013174 005777 002314          3#:    TST      @DMSCR      ;WAIT FOR TRANSMITTER DONE
013200 100375          BPL      3#
013202 012700 000020          MOV      #20,R0      ;SET UP TO CHECK ALL 16 LINES
013206 005077 002302          CLR      @DMSCR      ;START AT LINE 0
013212 005001          CLR      R1          ;KEEP TRACK OF LINE NUMBER
013214 012705 177777          4#:    MOV      #-1,R5     ;(R5)=EXPECTED BYTE COUNT,
                                ;IF LINE NUMBER NOT = 17
013220 005003          CLR      R3          ;(R3)=EXPECTED BUS ADDRESS,
                                ;IF LINE NUMBER NOT = 17
013222 017704 002276    MOV      @DMBC,R4      ;(R4)=ACTUAL BYTE ACOUNT

```

```

013226 017702 002270          MOV      @DMBA,R2          ;(R5)=ACTUAL BUS ADDRESS
013232 027727 002256 000017  CMP      @DMSCR,#17      ;IF LINE BEING COMPARED IS LINE 17
013240 001002          BNE      5$              ;
013242 005005          CLR      R5              ;EXPECTED BYTE COUNT=0
013244 005203          INC      R3              ;EXPECTED BUS ADDRESS = 1
013246 020504          5$:    CMP      R5,R4      ;IS BYTE COUNT CORRECT
013250 001401          BEQ      6$              ;
013252          HLT      4              ;BYTE COUNT ERROR
013252 104004          EMT      4              ;
013254 020302          6$:    CMP      R3,R2      ;IS BUS ADDRESS CORRECT
013256 001401          BEQ      7$              ;
013260          HLT      3              ;BUS ADDRESS ERROR
013260 104003          EMT      3              ;
013262 005277 002226          7$:    INC      @DMSCR      ;PREPARE TO CHECK NEXT LINE
013266 005201          INC      R1              ;
013270 005300          DEC      R0              ;CONTINUE IF ALL NOT DONE
013272 001350          BNE      4$              ;
013274 104400          8$:    SCOPE              ;CHECK FOR ITERATIONS, LOOP
          000020          LINE=LINE+1
          000000          BITX=BITX+BITX
35          000000          LINE=0
36          000000          XLINE=LINE
37          000001          BITX=1
38          000001          XBIT=BITX

```

```

1
2
3      ;SILO LOGIC TEST (MAINTENANCE MODE)
4      ;FORCE 1 CHARACTER INTO SILO USING SILO
5      ;MAINTENANCE BIT
6      ;VERIFY THAT CHARACTER AVAILABLE INTERRUPT OCCURS
7      ;VERIFY THAT CHARACTER AVAILABLE BIT IS
8      ;SET IN SYSTEM CONTROL REGISTER
9      ;VERIFY THAT SILO FILL REGISTER INCREMENTS TO 1
10     ;VERIFY THAT NEXT RECEIVED CHARACTER REGISTER
11     ;VALID DATA BIT IS SET
12     ;VERIFY THAT CORRECT DATA PATTERN WAS RECEIVED
13 013276 TS \XN,10,5#
    013276 012767 000340 164472 T50:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
    013304 012767 000010 002250      MOV    #10,ICOUNT  ;SET UP FOR 10 ITERATIONS
    013312 012767 013454 002236      MOV    #5#,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV    #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
14 013320 012777 004000 002166      MOV    #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
15 013326 012777 013374 002202      MOV    #2#,@DHRVEC  ;SET UP RECEIVER INTERRUPT
16 013334 012777 000340 002176      MOV    #340,@DHRLVL ;VECTOR AND STATUS
17 013342 052777 100000 002162      BIS    #BIT15,@DHSSR ;SET SILO MAINTENANCE
18 013350 012777 000100 002136      MOV    #BIT06,@DHSCR ;SET RECEIVER INTERRUPT ENABLE
19 013356 005067 164414      CLR    PS           ;ALLOW INTERRUPTS
20 013362 012700 001000      MOV    #1000,R0     ;SET UP DELAY FOR SILO LOAD
21 013366 005300      1# :  DEC    R0          ;DELAY FOR SILO LOAD
22 013370 001376      BNE    1#
23 013372      HLT    1          ;NO CHARACTER AVAILABLE INTERRUPT OCCURED
    013372 104001      EMT    1
24 013374 042777 100000 002130 2# :  BIC    #BIT15,@DHSSR ;CLEAR SILO MAINTENANCE BIT
25 013402 105777 002106      TSTB  @DHSCR        ;IS CHARACTER AVAILABLE BIT SET
26 013406 100401      BMI    3#
27 013410      HLT    5          ;CHARACTER AVAILABLE NOT, ERROR
    013410 104005      EMT    5
28 013412 017703 002114      3# :  MOV    @DHSSR,R3    ;READ SILO STATUS REGISTER
29 013416 017704 002074      MOV    @DHARC,R4    ;(R4)=ACTUAL DATA IN NEXT
30      ;RECEIVED CHARACTER REGISTER
31 013422 012705 125252      MOV    #125252,R5   ;(R5)=EXPECTED DATA IN
32      ;NEXT RECEIVED CHARACTER REGISTER
33 013426 020504      CMP    R5,R4        ;IS DATA IN SILO CORRECT
34 013430 001401      BEQ    4#
35 013432      HLT    6          ;SILO DATA ERROR
    013432 104006      EMT    6
36 013434 010304      4# :  MOV    R3,R4        ;GET SILO STATUS REGISTER
37 013436 042704 000300      BIC    #300,R4      ;CLEAR UNWANTED BITS
38 013442 012705 000400      MOV    #400,R5      ;(R5)=EXPECTED DATA
39      ;IN SILO STATUS REGISTER
40 013446 020504      CMP    R5,R4        ;CHECK RESULTS
41 013450 001401      BEQ    5#
42 013452      HLT    7          ;SILO STATUS ERROR
    013452 104007      EMT    7
43 013454 012706 017270      5# :  MOV    #STACK,SP    ;RESTORE STACK
44 013460 104400      SCOPE ;CHECK FOR ITERATIONS, LOOP
45
46      ;SILO LOGIC TEST (MAINTENANCE MODE)

```

```

47                                     ;LOAD 63(DECIMAL) CHARACTERS INTO SILO
48                                     ;VERIFY THAT SILO STATUS REGISTER COUNTS UP CORRECTLY
49
50 013462                               TS \XN,1,5,1#
   013462 012767 000340 164306 T51:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
   013470 012767 000001 002064      MOV    #1,ICOUNT        ;SET UP FOR 1 ITERATIONS
   013476 012767 013614 002052      MOV    #5,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST
                                     .IF NB <1#>
   013504 012767 013516 002046      MOV    #1,FREEZ1        ;SET UP TO LOOP WITH DATA           ; 3
                                     .ENDC
                                     XN=XN+1
51 013512                               MOV    #1,R1           ;SET UP TO LOAD SILO WITH 1 CHARACTER
52 013516 012777 004000 001770 1#:  MOV    #BIT11,DMHSCR    ;MASTER CLEAR INTERFACE
53 013524 010100                               MOV    R1,R0           ;SAVE COUNT
54 013526 005005                               CLR    R5              ;WILL BE COUNT OF CHARACTERS LOADED
55 013530 052777 100000 001774 2#:  BIS    #BIT15,DMHSSR    ;LOAD A CHARACTER
56 013536 012702 001000                               MOV    #1000,R2        ;STALL FOR SILO
57 013542 005302                               3#:  DEC    R2
58 013544 001376                               BNE    3#
59 013546 042777 100000 001756      BIC    #BIT15,DMHSSR    ;CLEAR LOAD BIT
60 013554 005205                               INC    R5              ;UPDATE COUNT OF CHARACTERS LOADED
61 013556 005300                               DEC    R0              ;IF ALL CHARACTERS NOT LOADED
62 013560 0C1363                               BNE    2#              ;LOAD ANOTHER
63 013562 017704 001744      MOV    DMHSSR,R4        ;READ SILO STATUS REGISTER
64 013566 042704 000300      BIC    #300,R4          ;CLEAR UNWANTED BITS
65 013572 000304      SWAB   R4              ;GET DATA INTO LOW BYTE
66 013574 020504      CMP    R5,R4           ;COMPARE
67 013576 001401      BEQ   4#
68 013600      HLT    7              ;SILO STATUS ERROR
   013600 104007      EMT    7
69 013602 10410      4#:  SCOPE1          ;CHECK FOR LOOP WITH CURRENT
70                                     ;LOAD COUNT (R5)
71 013604 005201      INC    R1              ;ADD 1 TO NUMBER OF CHARACTERS
72                                     ;TO BE LOADED INTO SILO
73 013606 022701 000100      CMP    #100,R1        ;CONTINUE UNTIL SILO IS FULL
74 013612 001341      BNE   1#
75 013614 104400      5#:  SCOPE
76
77                                     ;SILO LOGIC TEST (MAINTENANCE MODE)
78                                     ;LOAD 64 (DECIMAL) CHARACTERS INTO SILO
79                                     ;READ CHARACTERS OUT OF SILO
80                                     ;VERIFY THAT SILO STATUS REGISTER COUNTS DOWN CORRECTLY
81
82 013616                               TS \XN,1,7,1#
   013616 012767 000340 164152 T52:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
   013624 012767 000001 001730      MOV    #1,ICOUNT        ;SET UP FOR 1 ITERATIONS
   013632 012767 013774 001716      MOV    #7,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST
                                     .IF NB <1#>
   013640 012767 013652 001712      MOV    #1,FREEZ1        ;SET UP TO LOOP WITH DATA           ; 3
                                     .ENDC
                                     XN=XN+1
83 013646                               MOV    #1,R1           ;SET UP TO READ 1 CHARACTER
84 013652 012777 004000 001634 1#:  MOV    #BIT11,DMHSCR    ;MASTER CLEAR INTERFACE
85 013660 012705 000100      MOV    #100,R5         ;SILO STATUS COUNT SHOULD BE
86 013664 160105      SUB   R1,R5            ;100(OCTAL)-NUMBER
87                                     ;OF CHARACTERS READ
88 013666 012700 000100      MOV    #100,R0        ;SET UP TO LOAD SILO

```

```

89 013672 012702 001000      2$:  MOV    #1000,R2          ;SET UP DELAY
90 013676 052777 100000 001626  BIS    #BIT15,@DHSSR      ;LOAD 1 CHARACTER
91 013704 005302                3$:  DEC    R2                ;DELAY
92 013706 001376                BNE    3$
93 013710 042777 100000 001614  BIC    #BIT15,@DHSSR      ;CLEAR SILO LOAD BIT
94 013716 005300                DEC    R0                ;CONTINUE IF SILO NOT FULL
95 013720 001364                BNE    2$
96 013722 010100                MOV    R1,R0             ;SET UP TO READ SILO
97 013724 012702 001000      4$:  MOV    #1000,R2          ;SET UP DELAY FOR SILO
98 013730 005777 001562      TST    @DHNR0            ;READ SILO
99 013734 005302                5$:  DEC    R2                ;DELAY FOR SILO
100 013736 001376                BNE    5$
101 013740 005300                DEC    R0                ;UPDATE NUMBER OF CHARACTER S TO BE READ
102 013742 001370                BNE    4$                ;CONTINUE READING
103 013744 117704 001564      MOVB   @DHSLR,R4         ;READ SILO STATUS REGISER
104 013750 042704 000300      BIC    #300,R4          ;CLEAR UNWANTED BITS
105                                ;(R5)=EXPECTED SILO STATUS
106                                ;(IN UPPER BYTE)
107                                ;(R4)=ACTUAL SILO STATUS
108                                ;(IN UPPER BYTE)
109 013754 020504                CMP    R5,R4             ;COMPARE EXPECTED AND RECEIVED DATA
110 013756 001401                BEQ    6$
111 013760                HLT    7
112 013762 104007                EMT    7
113 013764 005201                6$:  SCOPE1                ;CHECK FOR LOOP WITH SAME READ COUNT
114 013766 020127 000101      INC    R1                ;UPDATE COUNT OF CHARACTERS TO BE READ
115 013772 001327                CMP    R1,#101          ;CONTINUE IF NOT DONE
116 013774 104400                BNE    1$
116 013774 104400                7$:  SCOPE

```

```

1
2
3           ;SILO LOGIC TEST (MAINTENANCE MODE)
4           ;SET SILO ALARM LEVEL TO ALL VALUES 0-77
5           ;VERIFY THAT CHARACTER AVAILABLE FLAG
6           ;DOES NOT SET UNTIL SILO FILL LEVEL IS GREATER THAN
7           ;SILO ALARM LEVEL
8 013776    TS \XN,40,5$,1$
013776    012767 000340 163772 T53:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
014004    012767 000040 001550      MOV    #40,ICOUNT       ;SET UP FOR 40 ITERATIONS
014012    012767 014152 001536      MOV    #5$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
           .IF NB <1$>
014020    012767 014030 001532      MOV    #1$,FREEZ1     ;SET UP TO LOOP WITH DATA           ; 3
           .ENDC
           XN=XN+1
9 014026    005005
10 014030    012777 004000 001456 1$:  CLR    R5             ;START AT ALARM LEVEL 0
11 014036    010577 001470      MOV    #BIT11,@DHSCR  ;MASTERCLEAR INTERFACE
12 014042    010501      MOV    R5,@DHSSR     ;SET ALARM LEVEL
13 014044    005201      MOV    R5,R1         ;SET FILL COUNT
14 014046    052777 100000 001456 2$:  INC    R1            ;ONE MORE THAN ALARM LEVEL
15 014054    012700 001000      BIS    #BIT15,@DHSSR ;LOAD A CHARACTER
16 014060    005300      MOV    #1000,R0      ;WAIT FOR SILO TO SETTLE
17 014062    001376      DEC    R0
18 014064    042777 100000 001440      BNE    --2
19 014072    005301      BIC    #BIT15,@DHSSR ;CLEAR MAINTENANCE BIT
20 014074    105777 001414      DEC    R1            ;UPDATE FILL COUNT
21 014100    100406      TSTB   @DHSCR        ;IS CHARACTER AVAILABLE FLAG SET
22 014102    005701      BMI    3$           ;YES
23 014104    001360      TST    R1            ;CHARACTER AVAILABLE FLAG NOT SET
24 014106    117703 001422      BNE    2$           ;SHOULD IT BE
25 014112    014112 104010      MOVB   @DHSLR,R3     ;READ SILO FILL LEVEL
           HLT    10    ;SILO ALARM ERROR
           EMT    10
26
           ;NO CHARACTER AVAILABLE WHEN EXPECTED
27 014114    000406
28 014116    020127 000001 3$:  BR     4$
29 014122    003403      CMP    R1,#1
30 014124    117703 001404      BLE    4$
           MOVB   @DHSLR,R3     ;READ SILO FILL LEVEL
31 014130    014130 104010      HLT    10           ;SILO ALARM ERROR
           EMT    10
32 014132    104410 4$:  SCOPE1
33 014134    005705      TST    R5
34 014136    001001      BNE    10$
35 014140    000261      SEC
36 014142    006105 10$:  ROL    R5
37 014144    022705 000100      CMP    #100,R5      ;GO TO NEXT ALARM LEVEL
38 014150    001327      BNE    1$           ;CONTINUE IF NOT DONE
39 014152    104400 5$:  SCOPE
           ;CHECK FOR ITERATIONS. LOOP

```

1
2 014154

.EOP r/BEGIN/

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

014154	104401			EOP:	TYPE		;TYPE NAME OF TEST	
014156	016217				MEPASS			
014160	005067	001426			CLR	LAST	;CLEAR LAST ERROR PC	
014164	005067	001356			CLR	ERRFLG	;CLEAR ERROR FLAG	
014170	005267	001354			INC	PASCNT	;UPDATE PASS COUNT	
014174	005767	164602			TST	LIGHTS	; ARE WE USING LIGHTS?	: 4
014200	001005				BNE	2#	; BRANCH IF WE ARE	: 6
014202	104401				TYPE		; TYPE PASCOUNT MESSAGE	: 5
014204	016232				PASTXT			: 5
014206	104402				OCTASC		; PRINT PASSCOUNT	: 4
014210	014246				PASARG			: 4
014212	000403				BR	3#	; CONTINUE	: 4
014214				2#:				: 4
014214	016767	001330	164560		MOV	PASCNT,LIGHTS	;DISPLAY PASS COUNT	: 4
014222				3#:				: 4
014222	013701	000042			MOV	@#42,R1	;CHECK FOR ACT-11 OR DDP	
014226	001405				BEQ	RESTR	;IF NOT, CONTINUE TESTING	
014230	000005				RESET			
014232	004711			LOGICAL:	JSR	PC,(R1)		
014234	000240				NOP			
014236	000240				NOP			
014240	000240				NOP			
014242	000167	165100		RESTR:	JMP	BEGIN		
014246	000001			PASARG:	.WORD	1	; PARAMETERS TO PRINT PASSCOUNT	: 5
014250	006	002			.BYTE	6,2		: 5
014252	015550				.WORD	PASCNT		: 5
3 014254				.SCOPE				
							;CHECK FOR LOOP OR CURRENT JUMP	: 3
							;CHECK FOR ITERATION SUPPRESSION	
014254	032777	002000	164516	SCOPER:	BIT	@SW10,@SWR		: 4
014262	001030				BNE	4#		: 4
014264	032777	040000	164506	1#:	BIT	@SW14,@SWR		: 4
014272	001021				BNE	3#		: 4
014274	032777	004000	164476		BIT	@SW11,@SWR		: 4
014302	001006				BNE	2#		
014304	005267	001254			INC	LPCNT		
014310	026767	001250	001244		CMP	LPCNT,ICOUNT		
014316	001007				BNE	3#		
014320	005067	001240		2#:	CLR	LPCNT		
014324	005067	001216			CLR	ERRFLG		
014330	011667	001220			MOV	(SP),RETRN		
014334	000002				RTI			
014336	016716	001212		3#:	MOV	RETRN,(SP)		
014342	000002				RTI			
014344	005767	001176		4#:	TST	ERRFLG		
014350	001745				BEQ	1#		

014352 000762
4 014354

.SCOP1 BR 2+

;CHECK FOR FREEZE ON CURRENT DATA

014354 03277. 001000 164416 SCOP1R: BIT #SW09,@SWR ; 4
014362 001402 BEQ 1+
014364 016716 001170 MOV FREEZ1,(SP)
014370 000002 1+: RTI
5 014372 .ERROR

;ERROR HANDLER

014372 032777 020000 164400 ERRORS: BIT #SW13,@SWR ; 4
014400 001055 BNE HALTS
014402 021667 001204 CMP (SP),LAST
014406 001404 BEQ 1+
014410 011667 001176 MOV (SP),LAST
014414 005067 001126 CLR ERRFLG
014420 104406 1+: SAV05P
014422 011605 MOV (SP),R5
014424 162705 000002 SUB #2,R5
014430 011504 MOV (R5),R4
014432 006304 ASL R4
014434 006304 ASL R4
014436 042704 177001 BIC #177001,R4
014442 062704 016352 ADD #ERRTAB,R4
014446 012467 000040 MOV (R4)+,ERRMSG
014452 011467 000052 MOV (R4),DATABP
014456 005767 001064 TST ERRFLG
014462 001403 BEQ TYPMSG
014464 005767 000040 TST DATABP
014470 001011 BNE TYPDAT ; 3
014472 104401 TYPMSG: TYPE ; 5
014474 016127 MCRLF ; 5
014476 104402 OCTASC ; 5
014500 014576 ERTAB0
014502 012767 000001 001036 MOV #1,ERRFLG
014510 104401 TYPE
014512 000000 ERRMSG: 0
014514 005767 000010 TYPDAT: TST DATABP
014520 001404 BEQ RESREG
014522 104401 TYPE ; 5
014524 016127 MCRLF ; 5
014526 104402 OCTASC
014530 000000 DATABP: 0
014532 104407 RESREG: RES05
014534 005777 164240 HALTS: TST @SWR ; 4
014540 100005 BPL EXITER
014542 010046 PUSHRO
014544 016600 000002 MOV 2(SP),R0
014550 000000 HALT
014552 012600 POPRO
014554 00526~ 000772 EXITER: INC ERRCNT
014560 032777 002000 164212 BIT #SW10,@SWR ; 4
014566 001402 BEQ 1+
014570 016716 000762 MOV ESCAPE,(SP)

014574 000002
 014576 000001
 014600 006 002
 014602 015604
 6 014604

1\$: RTI
 ERTABO: 1
 .BYTE 6,2
 SAVPC
 .TRPSRV
 ;TRAP DISPATCH SERVICE
 ;ARGUMENT OF TRAP IS EXTRACTED
 ;AND USED AS OFFSET TO OBTAIN POINTER
 ;TO SELECTED SUBROUTINE

; 3

014604 011646
 014606 162716 000002
 014612 017616 000000
 014616 006316
 014620 042716 177001
 014624 062716 016272
 014630 017616 000000
 014634 000136
 7 014636

TRPSRV: MOV (SP),-(SP) ;GET PC OF RETURN
 SUB #2,(SP) ;=PC OF TRAP
 MOV @ (SP),(SP) ;GET TRP
 TRPOK: ASL (SP) ;MULTIPLY TRAP ARG BY 2
 BIC #177001,(SP) ;CLEAR UNWANTED BITS
 ADD #TRPTAB,(SP) ;POINTER TO SUBROUTINE ADDRESS
 MOV @ (SP),(SP) ;SUBROUTINE ADDRESS
 JMP @ (SP)+ ;GO TO SUBROUTINE
 .TYPER

;TELETYPE OUTPUT ROUTINE

014636 017605 000000
 014642 062716 000002
 014646 105777 000636
 014652 100375
 014654 105715
 014656 001001
 014660 000002
 014662 112577 000624
 014666 000767
 8 014670

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

; 3

;ASCII STRING INPUT ROUTINE

014670 017667 000000 000006
 014676 062716 000002
 014702 104401
 014704 000000
 014706 012704 016314
 014712 012703 000007
 014716 105777 000562
 014722 100375
 014724 117714 000556
 014730 142714 000200
 014734 122427 000015
 014740 001413
 014742 117777 000540 000542
 014750 105777 000534
 014754 100375
 014756 005303
 014760 001356
 014762 104401
 014764 016123
 014766 000745
 014770 000002
 9 014772

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

;CONVERT ASCII STRING TO OCTAL

; 3

014772 011605
 014774 012567 000146
 015000 012567 000144
 015004 012567 000142
 015010 112567 000140
 015014 112567 000135
 015020 010516
 015022 005005
 015024 012704 016314
 015030 122714 000015
 015034 001420
 015036 121427 000060
 015042 002415
 015044 121427 000067
 015050 003012
 015052 142714 000060
 015056 152405
 015060 122714 000015
 015064 001406
 015066 006305
 015070 006305
 015072 006305
 015074 000760
 015076 104404
 015100 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

015102 020567 000042
 015106 101373
 015110 020567 000032
 015114 103770
 015116 136705 000032
 015122 001365

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

; 3

;STORE NUMBER AT SPECIFIED ADDRESS

015124 016704 000022
 015130 010524
 015132 062705 000002
 015136 105367 000013
 015142 001372
 015144 000002
 015146 000000
 015150 000000
 015152 000000
 015154 000000
 015155
 10 015156

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

;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER

015156 017601 000000
 015162 062716 000002

OCTASN: MOV #8(SP),R1
 ADD #2,(SP)

; 5

```

015166 012167 000130      MOV      (R1)+,WRDCNT
015172 112167 000126      1#:     MOVB   (R1)+,CHRCNT
015176 112167 000123      MOVB   (R1)+,SPACNT
015202 013167 000120      MOV     @R1)+,BINWRD      ; 3
015206 016704 000114      2#:     MOV     BINWRD,R4
015212 116705 000106      MOVB   CHRCNT,R5
015216 012700 016326      MOV     @TEMP,R0
015222 010403      3#:     MOV     R4,R3
015224 042703 177770      BIC     @177770,R3
015230 062703 000260      ADD     @260,R3
015234 110320      MOVB   R3,(R0)+
015236 006204      ASR    R4
015240 006204      ASR    R4
015242 006204      ASR    R4
015244 005305      DEC    R5
015246 001365      BNE    3#
015250 012703 016340      MOV     @MDATA,R3
015254 114023      4#:     MOVB   -(R0),(R3)+
015256 105367 000042      DECB   CHRCNT
015262 001374      BNE    4#
015264 105767 000035      TSTB   SPACNT
015270 001405      BEQ    6#
015272 112723 000240      5#:     MOVB   @240,(R3)+
015276 105367 000023      DECB   SPACNT
015302 001373      BNE    5#
015304 105013      6#:     CLRB   (R3)
015306 104401      TYPE
015310 016340      MDATA
015312 005367 000004      DEC    WRDCNT
015316 001325      BNE    1#
015320 000002      RTI
015322 000000      WRDCNT: 0
015324 000000      CHRCNT: 0
015325 001325      SPACNT=CHRCNT+1
015326 000000      BINWRD: 0
11 015330      .SAVREG

                                ;SAVE PC OF TEST THAT FAILED AND R0-R5

015330 016667 000004 000246 SV05P: MOV     4(SP),SAVPC

                                ;SAVE R0-R5

015336 010567 000236      SV05:  MOV     R5,SAVR5
015342 010467 000230      MOV     R4,SAVR4
015346 010367 000222      MOV     R3,SAVR3
015352 010267 000214      MOV     R2,SAVR2
015356 010167 000206      MOV     R1,SAVR1
015362 010067 000200      MOV     R0,SAVR0
015366 000002      RTI
12 015370      .RESREG

                                ;RESTORE R0-R5

015370 016700 000172      RS05:  MOV     SAVR0,R0
015374 016701 000170      MOV     SAVR1,R1
015400 016702 000166      MOV     SAVR2,R2
015404 016703 000164      MOV     SAVR3,R3

```

```

015410 016704 000162          MOV    SAVR4,R4
015414 016705 000160          MOV    SAVR5,R5
015420 000002
13 015422          CLRBCA
                                ;CLEAR BYTE COUNT AND BUS ADDRESS MEMORIES

015422 012700 000020          CLEAR: MOV    #20,R0          ;SET UPT TO CLEAR 16 (DECIMAL) LOCATIONS
015426 005077 000062          CLR    @DHSCR          ;START AT LOCATION 0
015432 005077 000066          1$: CLR    @DHBC          ;CLEAR BYTE COUNT
015436 005077 000060          CLR    @DHBA          ;CLEAR BUS ADDRESS
015442 005277 000046          INC    @DHSCR          ;ADVANCE LINE NUMBER
015446 005300          DEC    R0          ;CONTINUE IF NOT DONE
015450 001370          BNE    1$
015452 000207          RTS    PC          ;RETURN
14 015454          LDBC
                                ;LOAD ALL BYTE COUNT MEMORY LOCATIONS WITH -1

015454 012700 000020          LOAD: MOV    #20,R0          ;SET UP TO LOAD 16 (DECIMAL) LOCATIONS
015460 005077 000030          CLR    @DHSCR          ;START WITH LINE 0
015464 012777 177777 000032 2$: MOV    #-1,@DHBC          ;SET BYTE COUNT TO -1
015472 005277 000016          INC    @DHSCR          ;ADVANCE LINE NUMBER
015476 005300          DEC    R0          ;CONTINUE IF NOT DONE
015500 001371          BNE    2$
015502 000207          RTS    PC          ;RETURN TO CALLING ROUTINE
15 015504          .POINT  ↑/DHSCR,DHNRRC,DHLPR,DHBA,DHBC,DHBAR,DHBCR,DHSSR,DHSLR,DHRVEC,DHRLVL,DHTVEC,DHTLVL/
                                ;INDIRECT POINTERS
                                ; 3

015504 177560          TKCSR: 177560
015506 177562          TKOBR: 177562
015510 177564          TPCSR: 177564
015512 177566          TPDBR: 177566
                                .IRP  A <DHSCR,DHNRRC,DHLPR,DHBA,DHBC,DHBAR,DHBCR,DHSSR,DHSLR,DHRVEC,DHRLVL,DHTVEC,DH
TLVL>
                                A: 0
                                .ENDM
015514 000000          DHSCR: 0
015516 000000          DHNRRC: 0
015520 000000          DHLPR: 0
015522 000000          DHBA: 0
015524 0000C0          DHBC: 0
015526 000000          DHBAR: 0
015530 000000          DHBCR: 0
015532 000000          DHSSR: 0
015534 000000          DHSLR: 0
015536 000000          DHRVEC: 0
015540 000000          DHRLVL: 0
015542 000000          DHTVEC: 0
015544 000000          DHTLVL: 0
16 015546          .VARIA ↑/HCORE/
                                ;PROGRAM VARIABLES

015546 000000          ERRFLG: 0          ;ERROR FLAG
015550 000000          PASCNT: 0          ;PASS COUNT
015552 000000          ERRCNT: 0          ;ERROR COUNT
015554 000000          RETRN: 0          ;SCOPE RETURN ADDRESS FOR TEST LOOPING
015556 000000          ESCAPE: 0          ;ADDRESS FOR ERROR ESCAPE

```

```

015560 000000 FREEZ1: 0 ;DATA LOOPING RETURN ADDRESS
015562 000000 ICOUNT: 0 ;ITERATION COUNT FOR TEST IN PROGRESS
015564 000000 LPCNT: 0 ;NUMBER OF ITERATIONS THIS TEST
015566 000000 SAVR0: 0 ;R0 SAVE AREA
015570 000000 SAVR1: 0 ;R1 SAVE AREA
015572 000000 SAVR2: 0 ;R2 SAVE AREA ; 3
015574 000000 SAVR3: 0 ;R3 SAVE ARE
015576 000000 SAVR4: 0 ;R4 SAVE AREA
015600 000000 SAVR5: 0 ;R5 SAVE AREA
015602 000000 SAVSP: 0 ;STACK POINTER SAVE AREA
015604 000000 SAVPC: 0 ;CALLING ROUTINE SAVE AREA
015606 000000 INIFLG: 0 ;PROGRAM INITIALIZATION FLAG
015610 000000 STFLG: 0 ;PROGRAM START FLAG
015612 000000 LAST: 0 ;LAST ERROR PC
.IRP A <HCORE>
A: 0
.ENDM
HCORE: 0
.PFAIL
;ENTER HERE ON POWER FAILURE

015616 010046 PFAIL: MOV R0,-(SP) ;SAVE R0-R5 ON PROCESSOR STACK
015620 010146 MOV R1,-(SP)
015622 010246 MOV R2,-(SP)
015624 010346 MOV R3,-(SP)
015626 010446 MOV R4,-(SP)
015630 010546 MOV R5,-(SP)
015632 016746 162166 MOV 24,-(SP)
015636 010667 177740 MOV SP,SAVSP ;SAVE STACK POINTER
015642 012767 015654 162154 MOV #RESTART,24 ;SET UP FOR POWER UP TRAP ; 3
015650 000000 HALT ;HALT ON POWER DOWN NORMAL
015652 000777 BR .

;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

015654 016706 177722 RESTAR: MOV SAVSP,SP ;RESTORE STACK POINTER
015660 012605 MOV (SP)+,R5 ;RESTORE R0-R5
015662 012604 MOV (SP)+,R4
015664 012603 MOV (SP)+,R3
015666 012602 MOV (SP)+,R2
015670 012601 MOV (SP)+,R1
015672 012600 MOV (SP)+,R0
015674 012767 015616 162122 MOV #PFAIL,24 ;SET UP FOR POWER FAILURE
015702 012767 000340 162066 MOV #340,PS
015710 012706 017270 MOV #STACK,SP
015714 005067 000406 CLR TEMP
015720 005267 000402 INC TEMP
015724 001375 BNE .-4
015726 104401 TYPE ; 5
015730 016127 MCRLF ; 5
015732 104402 OCTASC
015734 015756 PFTAB
015736 104401 TYPE
015740 016132 MPFAIL
015742 005067 177600 CLR ERRFLG
015746 005067 177640 CLR LAST

```

015752	000177	177576			JMP	@RETRN
015756	000001				PFTAB:	1
015760	000006	000002				6.2
015764	015554					RETRN
18 015766					.MSG	↑/DH11 TRANSMITTER AND RECEIVER LOGIC TEST/,↑/CZDMC-CO/
015766	015	012	012		MTITLE:	.ASCIZ <15><12><12>/DH11 TRANSMITTER AND RECEIVER LOGIC TEST /<15><12>
015771	104	110	061			
015774	061	040	124			
015777	122	101	116			
016002	123	115	111			
016005	124	124	105			
016010	122	040	101			
016013	116	104	040			
016016	122	105	103			
016021	105	111	126			
016024	105	122	040			
016027	114	117	107			
016032	111	103	040			
016035	124	105	123			
016040	124	040	015			
016043	012	000				
016045	015	012	126		MVECTOR:	.ASCIZ <15><12>/VECTOR ADDRESS-/
016050	105	103	124			
016053	117	122	040			
016056	101	104	104			
016061	122	105	123			
016064	123	055	000			
016067	015	012	103		MREGAD:	.ASCIZ <15><12>/CONTROL REGISTER ADDRESS-/
016072	117	116	124			
016075	122	117	114			
016100	040	122	105			
016103	107	111	123			
016106	124	105	122			
016111	040	101	104			
016114	104	122	105			
016117	123	123	055			
016122	000					
016123	040	040	077		MQM:	.ASCIZ / ?/
016126	000					
016127	015	012	000		MCRLF:	.ASCIZ <15><12>
016132	040	040	120		MPFAIL:	.ASCIZ / POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
016135	117	127	105			
016140	122	040	106			
016143	101	111	114			
016146	125	122	105			
016151	054	040	120			
016154	122	117	107			
016157	122	101	115			
016162	040	122	105			
016165	123	124	101			
016170	122	124	040			
016173	101	124	040			
016176	124	105	123			
016201	124	040	111			
016204	116	040	120			
016207	122	117	107			

```

016212 122 105 123
016215 123 000
016217 015 012 103 MEPASS: .ASCIZ <15><12>/CZDHC-CO/
016222 132 104 110
016225 103 055 103
016230 060 000
016232 015 012 120 PASTXT: .ASCIZ <15><12>/PASS COUNT = / ; 5
016235 101 123 123
016240 040 103 117
016243 125 116 124
016246 040 075 040
016251 000
016252 015 012 122 MR: .ASCIZ <15><12>/R/
016255 000
016256 015 012 124 MTSTPC: .ASCIZ <15><12>/TEST PC-/
016261 105 123 124
016264 040 120 103
016267 055 000

19 016272 .EVEN
      .TRPTAB

      ;TABLE OF POINTERS FOR TRAP DECODING

016272 014254 TRPTAB: SCOPER
016274 014636 TYPER
016276 015156 OCTASN
016300 014670 INSTRG
016302 014762 INSTRE
016304 014772 PARAMS
016306 015330 SV05P
016310 015370 RS05
016312 014354 SCOP1R
20 016314 .BUFFER

      ;BUFFERS FOR INPUT-OUTPUT

016314 000000 INBUF: 0
      016326 .*.10
016326 000000 TEMP: 0
      016340 .*.10

016340 000000 MDATA: 0
      016352 .*.10
21 016352 .ERRTAB

      ;TABLE OF POINTERS TO ERROR MESSAGES AND DATA

016352 ERRTAB:
22 016352 016416 EM0
23 016354 017044 DT1
24 016356 016476 EM1
25 016360 000000 0
26 016362 016513 EM2
27 016364 000000 0
28 016366 016544 EM3
29 016370 017044 DT1
30 016372 016603 EM4

```

31	016374	017044				DT1
32	016376	016641				EM5
33	016400	000000				0
34	016402	016675				EM6
35	016404	017044				DT1
36	016406	016732				EM7
37	016410	017044				DT1
38	016412	016771				EM10
39	016414	017056				DT2
40	016416	125	116	105	EM0:	.ASCII /UNEXPECTED INTERRUPT/
	016421	130	120	105		
	016424	103	124	105		
	016427	104	040	111		
	016432	116	124	105		
	016435	122	122	125		
	016440	120	124			
41	016442	015	012	103		.ASCIZ <15><12>/CONTROL REGISTER CONTENTS/
	016445	117	116	124		
	016450	122	117	114		
	016453	040	122	105		
	016456	107	111	123		
	016461	124	105	122		
	016464	040	103	117		
	016467	116	124	105		
	016472	116	124	123		
	016475	000				
42	016476	116	117	040	EM1:	.ASCIZ /NO INTERRUPT/
	016501	111	116	124		
	016504	105	122	122		
	016507	125	120	124		
	016512	000				
43	016513	124	122	101	EM2:	.ASCIZ /TRANSMITTER DONE NOT SET/
	016516	116	123	115		
	016521	111	124	124		
	016524	105	122	040		
	016527	104	117	116		
	016532	105	040	116		
	016535	117	124	040		
	016540	123	105	124		
	016543	000				
44	016544	102	125	123	EM3:	.ASCII /BUS ADDRESS ERROR/
	016547	040	101	104		
	016552	104	122	105		
	016555	123	123	040		
	016560	105	122	122		
	016563	117	122			
45	016565	015	012	105		.ASCIZ <15><12>/EXP REC/
	016570	130	120	040		
	016573	040	040	040		
	016576	040	122	105		
	016601	103	000			
46	016603	102	131	124	EM4:	.ASCII /BYTE COUNT ERROR/
	016606	105	040	103		
	016611	117	125	116		
	016614	124	040	105		
	016617	122	122	117		
	016622	122				


```

47 016623 015 012 105 .ASCIZ <15><12>/EXP REC/
    016626 130 120 040
    016631 040 040 040
    016634 040 122 105
    016637 103 000
48 016641 103 110 101 EM5: .ASCIZ /CHARACTER AVAILABLE NOT SET/
    016644 122 101 103
    016647 124 105 122
    016652 040 101 126
    016655 101 111 114
    016660 101 102 114
    016663 105 040 116
    016666 117 124 040
    016671 123 105 124
    016674 000
49 016675 123 111 114 EM6: .ASCII /SILO DATA ERROR/
    016700 117 040 104
    016703 101 124 101
    016706 040 105 122
    016711 122 117 122
50 016714 015 012 105 .ASCIZ <15><12>/EXP REC/
    016717 130 120 040
    016722 040 040 040
    016725 040 122 105
    016730 103 000
51 016732 123 111 114 EM7: .ASCII /SILO STATUS ERROR/
    016735 117 040 123
    016740 124 101 124
    016743 125 123 040
    016746 105 122 122
    016751 117 122
52 016753 015 012 105 .ASCIZ <15><12>/EXP REC/
    016756 130 120 040
    016761 040 040 040
    016764 040 122 105
    016767 103 000
53 016771 123 111 114 EM10: .ASCII /SILO ALARM ERROR/
    016774 117 040 101
    016777 114 101 122
    017002 115 040 105
    017005 122 122 117
    017010 122
54 017011 015 012 101 .ASCIZ <15><12>/ALARM LEVEL FILL LEVEL/
    017014 114 101 122
    017017 115 040 114
    017022 105 126 105
    017025 114 040 040
    017030 106 111 114
    017033 114 040 114
    017036 105 126 105
    017041 114 000
55 .EVEN
56
57 ;DATA TABLES FOR ERROR OUTPUT
58
59 017044 000002 DT1: 2
60 017046 006 002 .BYTE 6.2

```

61	017050	015600				SAVR5
62	017052	006	000		.BYTE	6.0
63	017054	015576				SAVR4
64	017056	000002			DT2:	2
65	017060	002	014		.BYTE	2.14
66	017062	015600				SAVR5
67	017064	002	000		.BYTE	2.0
68	017066	015574				SAVR3
69	017070				.ENDCOD	
	017070	000000			ENDCOD:	0
70		000001			.END	

ADRCNT = 015155	EM3 016544	MREGAD 016067	SIZE 001162	T17 005260
BEGIN 001346	EM4 016603	MSG 014704	SPACNT = 015325	T2 001622
BINWRD 015326	EM5 016641	MTITLE 015766	STACK = 017270	T20 005516
BITX = 000001	EM6 016675	MTSTPC 016256	START 001004	T21 005754
BIT00 = 000001	EM7 016732	MVECTO 016045	STFLG 015610	T22 006212
BIT01 = 000002	ENDCOD 017070	N = 000001	SV05 015336	T23 006450
BIT02 = 000004	EOP 014154	OCTASC = 104402	SV05P 015330	T24 006706
BIT03 = 000010	ERRCNT 015552	OCTASN 015156	SWR 001000	T25 007144
BIT04 = 000020	ERRFLG 015546	PARAM = 104405	SW00 = 000001	T26 007402
BIT05 = 000040	ERRMSG 014512	PARAMS 014772	SW01 = 000002	T27 007640
BIT06 = 000100	ERRORS 014372	PARAM1 015022	SW02 = 000004	T3 002004
BIT07 = 000200	ERRTAB 016352	PARERR 015076	SW03 = 000010	T30 010076
BIT08 = 000400	ERTAB0 014576	PASARG 014246	SW04 = 000020	T31 010246
BIT09 = 001000	ESCAPE 015556	PASCNT 015550	SW05 = 000040	T32 010416
BIT10 = 002000	EXITER 014554	PASTXT 016232	SW06 = 000100	T33 010566
BIT11 = 004000	FREEZ1 015560	PFAIL 015616	SW08 = 000400	T34 010736
BIT12 = 010000	HALTS 014534	PFTAB 015756	SW09 = 001000	T35 011106
BIT13 = 020000	HCORE 015614	POPPO = 012600	SW10 = 002000	T36 011256
BIT14 = 040000	HILIM 015150	POP1SP = 005726	SW11 = 004000	T37 011426
BIT15 = 100000	ICOUNT 015562	POP2SP = 022626	SW12 = 010000	T4 002166
CHRCNT 015324	INBUF 016314	PS = 177776	SW13 = 020000	T40 011576
CLEAR 015422	INIFLG 015606	PUSHRO = 010046	SW14 = 040000	T41 011746
DATABP 014530	INSTER = 104404	PUSH1S = 005746	SW15 = 100000	T42 012116
DEVADR 015152	INSTR = 104403	PUSH2S = 024640	TEMP 016326	T43 012266
DHBA 015522	INSTRE 014762	RESREG 014532	TKCSR 015504	T44 012436
DHBAR 015526	INSTRG 014670	RESTAR 015654	TKOBR 015506	T45 012606
DHBC 015524	INSTR1 014702	RESTRT 014242	TPCSR 015510	T46 012756
DHBCR 015530	INSTR2 014770	RESOS = 104407	TPOBR 015512	T47 C13126
DHLPR 015520	LAST 015612	RETRN 015554	TRPOK 014616	T5 002360
DHNRC 015516	LIGHTS 001002	PS05 015370	TRPSRV 014604	T50 013276
DHRLVL 015540	LIMITS 015102	SAVPC 015604	TRPTAB 016272	T51 013462
DHRVEC 015536	LINE = 000000	SAVRO 015566	TYPDAT 014514	T52 013616
DHSCR 015514	LOAD 015454	SAVR1 015570	TYPE = 104401	T53 013776
DHSLR 015534	LOBITS 015154	SAVR2 015572	TYPER 014636	T6 002552
DHSSR 015532	LOGICA 014232	SAVR3 015574	TYPMSG 014472	T7 002744
DHTLVL 015544	LOLIM 015146	SAVR4 015576	T1 001440	VEC1 001224
DHTVEC 015542	LPCNT 015564	SAVRS 015600	T10 003136	VEC2 001234
DT1 017044	MCRLF 016127	SAVSP 015602	T11 003374	WRDCNT 015322
DT2 017056	MDATA 016340	SAVOSP = 104406	T12 003632	X = 000000
EMO 016416	MEPASS 016217	SCOPE = 104400	T13 004070	XBIT = 000001
EM1 016476	MPFAIL 016132	SCOPEP 014254	T14 004326	XLIN = 000000
EM10 016771	MQH 016123	SCOPE1 = 104410	T15 004564	XN = 000054
EM2 016513	MR 016252	SCOPIR 014354	T16 005022	Y = 000011

. ABS. 017072 000
000000 001
ERRORS DETECTED: 0

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