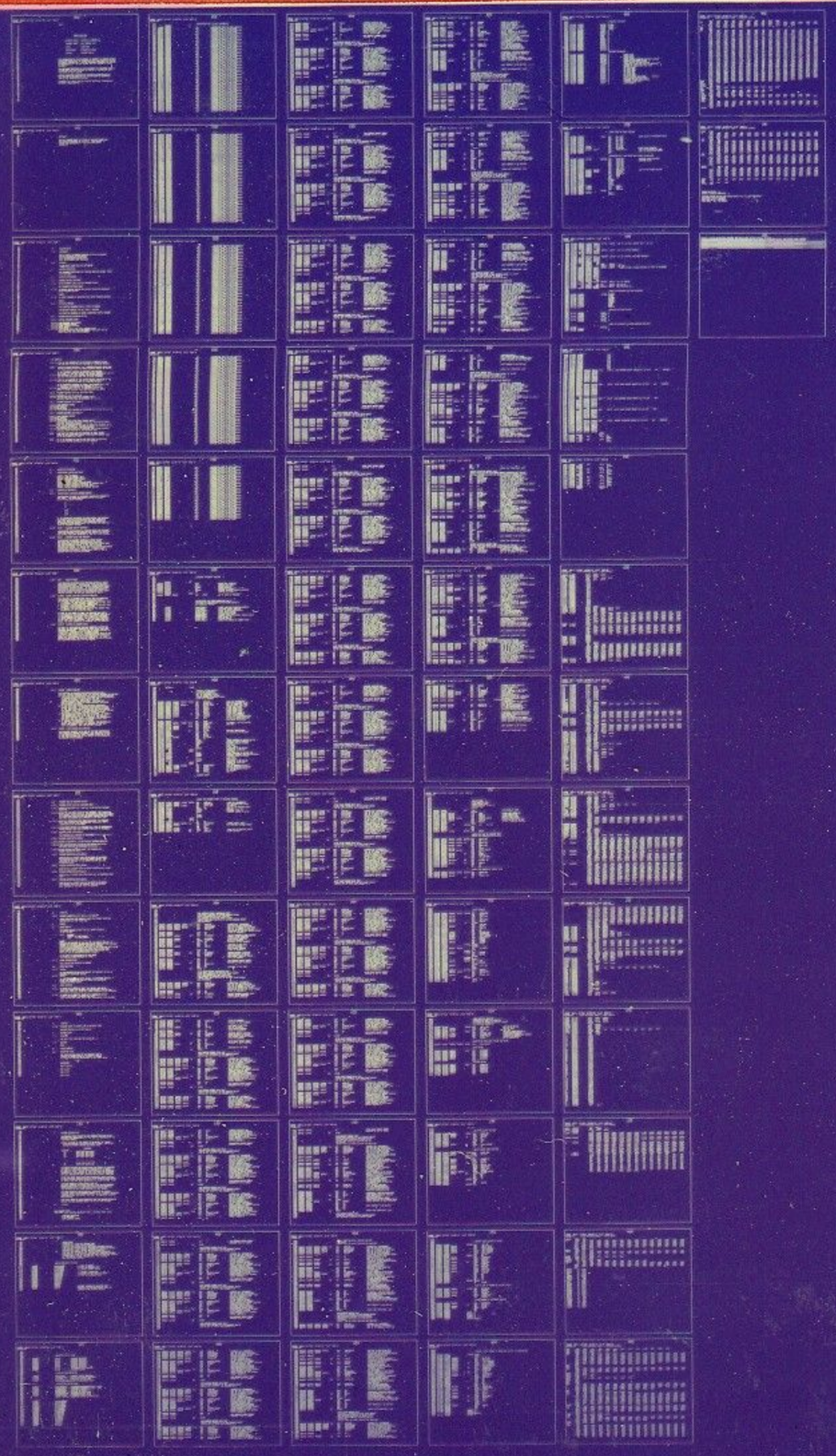
DH11

MEMORY TEST
MD-11-DZDHB-B

EP-DZDHB-B-DL-A

FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A.



```
                                .REM !
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDHB-B-D

PRODUCT NAME: DH11 MEMORY TEST

DATE: JUNE 1976

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: MICHAEL DAVIS

1. ABSTRACT

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

DO1

2. REQUIREMENTS

2.1 EQUIPMENT

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

2.2 STORAGE

THE PROGRAM LOADS INTO 4KW OF MEMORY

3. LOADING PROCEDURE

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

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

4.1.1 AFTER PROGRAM LOAD (INITIAL PROGRAM START)

ALL CONSOLE SWITCHES DOWN

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

SW00=1

4.1.3 TO START PROGRAM AT SELECTED TEST AFTER PROGRAM RESTART

SW01=1

4.2 STARTING ADDRESS

THE STARTING ADDRESS FOR ALL TESTS IS 000200

THE RESTART ADDRESS FOR ALL TESTS I 0002000

THE STARTING ADDRESS TO ENTER A SELECTED TEST IS 000200

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 INITIAL PROGRAM START

4.3.1.1 LOAD PROGRAM INTO MEMORY
4.3.1.2 LOAD ADDRESS 000200
4.3.1.3 CLEAR CONSOLE SWITCHES
4.3.1.4 PRESS START
4.3.1.5 THE PROGRAM WILL TYPE "DH11 MEMORY TEST"
AND WILL THEN TYPE "VECTOR ADDRESS-" AND WAIT FOR AN
INPUT FROM THE TELETYPE KEYBOARD.

```
                                4.3 (CONT'D)

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

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

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

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

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

                                4.3.2   PROGRAM RESTART WITH ALL SWITCHES DOWN

                                4.3.2.1 PERFORM 4.3.1.2 TO 4.3.1.5
                                4.3.2.2 THE PROGRAM WILL TYPE "DH11 MEMORY TEST"
                                        AND WILL THEN CONTINUE AS DESCRIBED IN 4.3.1.9

                                4.3.3   PROGRAM RESTART WITH SW00=1

                                4.3.3.1 LOAD ADDRESS 000200
                                4.3.3.2 SET SW01=1
                                4.3.3.3 PRESS START
                                4.3.3.4 THE PROGRAM WILL PERFORM AS DESCRIBED IN 4.3.1.5 TO 4.3.1.9

                                4.3.4   PROGRAM RESTART WITH SW01=1

                                4.3.4.1 LOAD ADDRESS 000200
                                4.3.4.2 SET SW00=1
                                4.3.4.3 PRESS START
                                4.3.4.4 THE PROGRAM WILL TYPE "DH11 MEMORY TEST"
                                        AND WILL THEN TYPE "TEST PC-" AND WILL WAIT FOR AN INPUT
                                        FROM THE TELETYPE KEYBOARD
                                4.3.4.5 TYPE IN THE ADDRESS OF THE TEST AT WHICH THE PROGRAM IS TO
                                        BE STARTED FOLLOWED BY <CARRIAGE RETURN>
                                4.3.4.6 THE PROGRAM WILL TYPE R TO INDICATE THAT IT HAS STARTED
                                        AND WILL START TESTING AT THE SELECTED TEST.

                                NOTE:   CARE MUST BE TAKEN WHEN THS 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
```

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

                                                .WORD   2
                                                .WORD   0
                                                .WORD   4
                                                .WORD   0
                                                .
                                                .
                                                .
                                                .WORD   772
                                                .WORD   0
                                                .WORD   776
                                                .WORD   0

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

                                5.2.2   START   (PROGRAM INITIALIZATION)

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

                                5.2.3   BEGIN   (PROGRAM START AND RESTART)

                                        THIS ROUTINE IS ENTERED IMMEDIATLY AFTER "START"
                                        AND EACH TIME A PROGRAM PASS HAS BEEN COMPLETED.
                                        THE ROUTINE SETS UP THE PROCESSOR STACK AND STATUS WORD
                                        AND THEN TRANSFERS CONTROL TO THE TEST AT WHICH
                                        TESTING WILL BEGIN.  IF SW01=0 WHEN THIS ROUTINE IS ENTERD
                                        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.
```

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.

5.2.7 ERRORS (ERROR HANDLER)

THIS ROUTINE IS ENTERED UPON ERROR DETECTION ONLY.
WITH ALL CONSOLE SWITCHES DOWN, THE ROUTINE PROCEDES 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 A HALT ON ERROR. IF SW15=1 THE PROGRAM WILL HALT WITH THE PC OF THE CALL TO THE ERROR ROUTINE IN R0. IF SW15=0, THE PROGRAM WILL NOT HALT, BUT WILL CHECK FOR ESCAPE TO NEXT TEST.

D) IF SW10=0, THE ROUTINE WILL RETURN TO THE TEST IN PROGRESS. IF SW10=1, THE ROUTINE WILL ABORT THE CURRENT TEST, AND TRANSFER TO THE NEXT TEST IN SEQUENCE, THRU THE ROUTINE "SCOPER".

5.2.8 TRPSRV (TRAP DECODE AND DISPATCH)

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

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 "DZDH8" 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 "DZDH8" 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 R0.

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.

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 OPERATOT ACTION IS
REQUIRED TO CONTINUE TESTING

6.2.2 SW15=1

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

6.3 SCOPE LOOPING

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

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

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 DZDHB TO END OF TYPEOUT OF DZDHB) IS GIVEN FOR VARIOUS PROCESSORS IN THE TABLE BELOW

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

```
                                9.      PROGRAM DESCRIPTION

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

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

                                                                CONTENTS
                                        LOCATION
                                           00           0000 0000 0000 0000
                                           01           0001 0001 0001 0001
                                           02           0010 0010 0010 0010
                                           03           0011 0011 0011 0011

                                            ...         ...  ...  ...  ...

                                           16           1110 1110 1110 1110
                                           17           1111 1111 1111 1111

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

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

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

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

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

                                10.     LISTING

                                        ↑

                                ;DH11 MEMORY TEST
                                ;COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                        ;STARTING PROCEDURE
                                        ;LOAD PROGRAM
                                        ;LOAD ADDRESS 000200
```

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


                                ;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
        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                  ;RESTART PROGRAM AT SELECTED TEST
        000001                  SW00=1                  ;RESELECT VECTOR AND CONTROL REGISTER
                                                        ;ADDRESS AFTER PROGRAM RESTART
```

```
                                ;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

177570                          SWR=177570      ;CONSOLE SWITCH REGISTER
177570                          LIGHTS=177570   ;PDP-11/45 DISPLAY REGISTER
177776                          PS=177776       ;PROCESSOR STATUS WORD
012606                          STACK=ENDCOD+200;START OF PROCESSOR STACK

                                ;INSTRUCTION DEFINITIONS

005746                          PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
005726                          POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
010046                          PUSHR0=10046    ;SAVE R0 ON STACK
012600                          POPR0=12600     ;RESTORE R0 FROM STACK
024646                          PUSH2SP=24646   ;DECREMENT STACK TWICE
022626                          POP2SP=22626    ;INCREMENT STACK TWICE
                                .EQUIV  EMT,HLT ;BASIC DEFINITION OF ERROR CALL


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

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

C02


```
629  000154  000156                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
630  000156  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
631  000160  000162                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
632  000162  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
633  000164  000166                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
634  000166  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
635  000170  000172                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
636  000172  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
637  000174  000176                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
638  000176  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
639  000200  000202                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
640  000202  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
641  000204  000206                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
642  000206  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
643  000210  000212                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
644  000212  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
645  000214  000216                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
646  000216  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
647  000220  000222                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
648  000222  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
649  000224  000226                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
650  000226  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
651  000230  000232                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
652  000232  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
653  000234  000236                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
654  000236  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
655  000240  000242                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
656  000242  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
657  000244  000246                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
658  000246  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
659  000250  000252                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
660  000252  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
661  000254  000256                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
662  000256  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
663  000260  000262                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
664  000262  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
665  000264  000266                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
666  000266  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
667  000270  000272                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
668  000272  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
669  000274  000276                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
670  000276  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
671  000300  000302                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
672  000302  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
673  000304  000306                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
674  000306  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
675  000310  000312                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
676  000312  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
677  000314  000316                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
678  000316  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
679  000320  000322                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
680  000322  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
681  000324  000326                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
682  000326  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
683  000330  000332                  .+2     ;UNEXPECTED TRAP TO THIS LOCATION
684  000332  000000                  HALT    ;EXAMINE STACK TO FIND CAUSE
```

```
  685  000334  000336                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  686  000336  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  687  000340  000342                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  688  000342  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  689  000344  000346                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  690  000346  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  691  000350  000352                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  692  000352  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  693  000354  000356                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  694  000356  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  695  000360  000362                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  696  000362  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  697  000364  000366                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  698  000366  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  699  000370  000372                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  700  000372  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  701  000374  000376                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  702  000376  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  703  000400  000402                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  704  000402  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  705  000404  000406                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  706  000406  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  707  000410  000412                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  708  000412  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  709  000414  007416                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  710  000416  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  711  000420  000422                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  712  000422  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  713  000424  000426                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  714  000426  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  715  000430  000432                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  716  000432  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  717  000434  000436                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  718  000436  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  719  000440  000442                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  720  000442  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  721  000444  000446                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  722  000446  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  723  000450  000452                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  724  000452  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  725  000454  000456                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  726  000456  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  727  000460  000462                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  728  000462  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  729  000464  000466                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  730  000466  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  731  000470  000472                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  732  000472  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  733  000474  000476                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  734  000476  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  735  000500  000502                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  736  000502  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  737  000504  000506                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  738  000506  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  739  000510  000512                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  740  000512  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
```

```
741  000514  000516                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
742  000516  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
743  000520  000522                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
744  000522  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
745  000524  000526                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
746  000526  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
747  000530  000532                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
750  000532  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
751  000534  000536                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
752  000536  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
753  000540  000542                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
754  000542  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
755  000544  000546                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
756  000546  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
757  000550  000552                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
760  000552  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
761  000554  000556                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
762  000556  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
763  000560  000562                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
764  000562  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
765  000564  000566                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
766  000566  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
767  000570  000572                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
770  000572  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
771  000574  000576                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
772  000576  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
773  000600  000602                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
774  000602  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
775  000604  000606                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
776  000606  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
777  000610  000612                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1000 000612  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
1001 000614  000616                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1002 000616  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
1003 000620  000622                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1004 000622  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
1005 000624  000626                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1006 000626  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
1007 000630  000632                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1010 000632  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
1011 000634  000636                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1012 000636  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
1013 000640  000642                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1014 000642  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
1015 000644  000646                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1016 000646  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
1017 000650  000652                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1020 000652  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
1021 000654  000656                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1022 000656  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
1023 000660  000662                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1024 000662  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
1025 000664  000666                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1026 000666  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
1027 000670  000672                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
1030 000672  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
```

```
  797  000674  000676                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  798  000676  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  799  000700  000702                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  800  000702  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  801  000704  000706                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  802  000706  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  803  000710  000712                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  804  000712  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  805  000714  000716                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  806  000716  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  807  000720  000722                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  808  000722  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  809  000724  000726                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  810  000726  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  811  000730  000732                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  812  000732  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  813  000734  000736                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  814  000736  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  815  000740  000742                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  816  000742  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  817  000744  000746                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  818  000746  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  819  000750  000752                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  820  000752  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  821  000754  000756                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  822  000756  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  823  000760  000762                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  824  000762  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  825  000764  000766                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  826  000766  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  827  000770  000772                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  828  000772  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
  829  000774  000776                    .+2     ;UNEXPECTED TRAP TO THIS LOCATION
  830  000776  000000                    HALT    ;EXAMINE STACK TO FIND CAUSE
```

```
                                                ;STANDARD INTERRUPT VECTORS


        000024                          .=24
000024  011276                          PFAIL                   ;POWER FAIL HANDLER
000026  000340                          340                     ;SERVICE AT LEVEL 7
000030  010142                          ERRORS                  ;ERROR HANDLER
000032  000340                          340                     ;SERVICE AT LEVEL 7
000034  010344                          TRPSRV                  ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                          340                     ;SERVICE AT LEVEL 7
        000200                          .=200
000200  000167  000574                  JMP     START           ;GO TO START OF PROGRAM



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

        104400                          SCOPE=TRAP+Y            ;SCOPE LOOP AND ITERATION HANDLER
        104401                          TYPE=TRAP+Y             ;TELETYPE OUTPUT ROUTINE
        104402                          OCTASC=TRAP+Y           ;OCTAL TO ASCII CONVERSION
        104403                          INSTR=TRAP+Y            ;INPUT ASCII STRING
        104404                          INSTER=TRAP+Y           ;STRING INPUT ERROR
        104405                          PARAM=TRAP+Y            ;CONVERT STRING TO OCTAL, CHECK LIMITS
        104406                          SAVOSP=TRAP+Y           ;SAVE R0-R5, PC
        104407                          RESOS=TRAP+Y            ;RESTORE R0-R5
        104410                          SCOPE1=TRAP+Y           ;CHECK FOR FREEZE ON CURRENT DATA
```

```
  859            001000                  .=1000
  860
  861                                    ;PROGRAM INITIALIZATION
  862                                    ;LOCK OUT INTERRUPTS
  863                                    ;SET UP PROCESSOR STACK
  864                                    ;SET UP POWER FAIL VECTOR
  865                                    ;CLEAR PROGRAM FLAGS AND COUNTS
  866                                    ;TYPE TITLE MESSAGE
  867
  868  001000  012767  000340  176770  START:  MOV     #340,PS         ;LOCK OUT INTERRUPTS
  869  001006  012706  012606          MOV     #STACK,SP       ;SET UP PROCESSOR STACK
  870  001012  012737  011276  000024  MOV     #PFAIL,@#24     ;SET UP POWER FAIL TRAP
  871  001020  005067  010246          CLR     STFLG           ;CLEAR TEST START FLAG
  872  001024  005067  010202          CLR     PASCNT          ;CLEAR PASS COUNT
  873  001030  005067  010200          CLR     ERRCNT          ;CLEAR ERROR COUNT
  874  001034  005067  010170          CLR     ERRFLG          ;CLEAR ERROR FLAG
  875  001040  005067  010164          CLR     ERRPC           ;CLEAR LAST ERROR PC
  876  001044  104401  011442          TYPE    ,MTITLE         ;TYPE TITLE MESSAGE
  877  001050  005767  010214          TST     INIFLG          ;CHECK INITIALIZATION FLAG
  878  001054  001001                  BNE     VEC1            ;IF NOT 0, CHECK SWITCHES
  879                                                           ;FOR REINITIALIZATION
  880  001056  000404                  BR      VEC2
  881  001060  032767  000001  176502  VEC1:   BIT     #SW00,SWR       ;IF SW00=1, GET NEW VECTOR
  882  001066  001445                  BEQ     BEGIN           ;AND CSR
  883  001070  012701  000300          VEC2:   MOV     #300,R1
  884  001074  012702  000302                  MOV     #302,R2
  885  001100  012703  000004                  MOV     #4,R3
  886  001104  010211                  1$:     MOV     R2,(R1)         ;RESTORE TRAPCATCHER
  887  001106  005012                          CLR     (R2)            ;IN FLOATING VECTOR AREA
  888  001110  060301                          ADD     R3,R1
  889  001112  060302                          ADD     R3,R2
  890  001114  020127  001000                  CMP     R1,#1000
  891  001120  001371                          BNE     1$
  892  001122  104403                          INSTR                   ;INPUT ADDRESS OF DEVICE VECTOR
  893  001124  011471                          MVECTOR                 ;MESSAGE "VECTOR ADDRESS-"
  894  001126  104405                          PARAM                   ;CONVERT STRING TO OCTAL
  895  001130  000300                          300                     ;LOW LIMIT
  896  001132  000770                          770                     ;HIGH LIMIT
  897  001134  011220                          DHRVEC                  ;LOCATIONS TO BE FILLED
  898  001136     003                  .BYTE   3                       ;NUMBER OF LOCATIONS
  899  001137     004                  .BYTE   4                       ;LSB MASK
  900  001140  104403                          INSTR                   ;INPUT ADDRESS OF DEVICE CSR
  901  001142  011513                          MREGAD                  ;MESSAGE "CONTROL REGISTER ADDRESS-"
  902  001144  104405                          PARAM                   ;CONVERT STRING TO OCTAL
  903  001146  000000                          0                       ;LOW LIMIT
  904  001150  177776                          177776                  ;HIGH LIMIT
  905  001152  011176                          DHSCR                   ;LOCATIONS TO BE FILLED
  906  001154     007                  .BYTE   7                       ;NUMBER OF LOCATIONS
  907  001155     010                  .BYTE   10                      ;LSB MASK
  908  001156  016767  010032  010032          MOV     DHSSR,DHSLR     ;SET UP ADDRESS OF SILO
  909  001164  005267  010026                  INC     DHSLR           ;STATUS REGISTER HIGH BYTE
  910  001170  005767  010074                  TST     INIFLG          ;IF INITIALIZATION FLAG
  911  001174  001002                          BNE     BEGIN           ;IS CLEARED
  912  001176  005167  010066                  COM     INIFLG          ;SET IT
  913
  914                                    ;PROGRAM START
```

```
                                                ;CHECK FOR PROGRAM START AT SELECTED ADDRESS
                                                
001202  012767  000340  1'%6566  BEGIN:  MOV     #340,PS                 ;LOCK OUT INTERRUPTS
001210  012706  012606                  MOV     #STACK,SP               ;SET UP PROCESSOR STACK
001214  032767  000002  1'%6346          BIT     #SW01,SWR               ;IF SW01=1
001222  001410                          BEQ     1$                      ;GET PC FOR PROGRAM START
001224  104403                          INSTR                           ;GET PC
001226  011657                          MTSTPC                          ;MESSAGE "TEST PC"
0012'   104405                          PARAM                           ;CONVERT STRING TO OCTAL
0012 2  000000                          0
001234  017500                          17500
001236  000207                          RETURN
001240      001                 .BYTE   1
001241      001                 .BYTE   1
001242  000410                          BR      2$
001244  012767  001274  00'764  1$:     MOV     #T1,RETURN              ;NORMAL START, TEST 1
001252  005767  010014                  TST     STFLG                   ;IF LOOPING, BYPASS TYPEOUT
001256  001004                          BNE     3$
001260  005167  010006                  COM     STFLG
001264  104401  011653          2$:     TYPE    ,MR                     ;TYPE "R" TO INDICATE START
001270  000177  007742          3$:     JMP     @RETURN                 ;START TESTING
```

```
                                        ;BUS ADDRESS MEMORY ADDRESSING TEST
                                        ;LOAD EACH LOCATION IN THE BUS ADDRESS MEMORY
                                        ;WITH THE ADDRESS OF THAT LOCATION.
                                        ;THE ADDRESS IS REPEATED EVERY 4 BITS
                                        ;VERIFY THAT EACH LOCATION IN THE BUS ADDRESS MEMORY
                                        ;WAS ADDRESSED.
001274  012767  000340  176474  T1:    MOV     #340,PS         ;DISABLE ALL INTERRUPTS
001302  012767  000100  007734         MOV     #100,ICOUNT     ;SET UP FOR 100 ITERATIONS
001310  012767  001410  007722         MOV     #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
001316  012700  000020                 MOV     #20,R0          ;SET UP TO ADDRESS 20 (OCTAL)
                                                                ;LOCATIONS IN THE BUS ADDRESS MEMORY
001322  005004                         CLR     R4              ;START AT ADDRESS 0
001324  005002                         CLR     R2
001326  010477  007644         1$:    MOV     R4,@DHSCR       ;SELECT ADDRESS IN BUS ADDRESS
                                                                ;MEMORY TO BE ADDRESSED
001332  010277  007646                 MOV     R2,@DHBA        ;LOAD MEMORY LOCATION
001336  062702  010421                 ADD     #10421,R2       ;WITH ITS ADDRESS
001342  005204                         INC     R4              ;ADVANCE TO NEXT ADDRESS
001344  005300                         DEC     R0
001346  001367                         BNE     1$              ;CONTINUE IF NOT DONE
001350  012700  000020                 MOV     #20,R0          ;SET UP TO CHECK
                                                                ;EACH MEMORY ADDRESS
001354  005004                         CLR     R4              ;START AT ADDRESS 0
001356  005002                         CLR     R2
001360  010477  007612         2$:    MOV     R4,@DHSCR       ;ADDRESS MEMORY LOCATION
001364  017703  007614                 MOV     @DHBA,R3        ;READ CONTENTS OF MEMORY
001370  020203                         CMP     R2,R3           ;WAS MEMORY LOCATION LOADED
                                                                ;WITH ITS ADDRESS
001372  001401                         BEQ     3$
001374  104001                         HLT     1               ;BUS ADDRESS MEMORY ERROR
001376  005204                 3$:    INC     R4              ;ADVANCE TO NEXT LOCATION
001400  062702  010421                 ADD     #10421,R2
001404  005300                         DEC     R0
001406  001364                         BNE     2$              ;CONTINUE IF NOT DONE
001410  104400                 4$:    SCOPE

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

001412  012767  000340  176356  T2:    MOV     #340,PS         ;DISABLE ALL INTERRUPTS
001420  012767  000100  007616         MOV     #100,ICOUNT     ;SET UP FOR 100 ITERATIONS
001426  012767  001526  007604         MOV     #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
001434  012700  000020                 MOV     #20,R0          ;SET UP TO ADDRESS 20 (OCTAL)
                                                                ;LOCATIONS IN THE BYTE COUNT MEMORY
001440  005004                         CLR     R4              ;START AT ADDRESS 0
001442  005002                         CLR     R2
001444  010477  007526         1$:    MOV     R4,@DHSCR       ;SELECT ADDRESS IN BYTE COUNT
                                                                ;MEMORY TO BE ADDRESSED
001450  010277  007532                 MOV     R2,@DHBC        ;LOAD MEMORY LOCATION
```

```
   992  001454  062702  010421                 ADD     #10421,R2               ;WITH ITS ADDRESS
   993  001460  005204                         INC     R4                      ;ADVANCE TO NEXT ADDRESS
   994  001462  005300                         DEC     R0
   995  001464  001367                         BNE     1$                      ;CONTINUE IF NOT DONE
   996  001466  012700  000020                 MOV     #20,R0                  ;SET UP TO CHECK
   997                                                                          ;EACH MEMORY ADDRESS
   998  001472  005004                         CLR     R4                      ;START AT ADDRESS 0
   999  001474  005002                         CLR     R2
  1000  001476  010477  007474          2$:    MOV     R4,@DHSCR               ;ADDRESS MEMORY LOCATION
  1001  001502  017703  007500                 MOV     @DHBC,R3                ;READ CONTENTS OF MEMORY
  1002  001506  020203                         CMP     R2,R3                   ;WAS MEMORY LOCATION LOADED
  1003                                                                          ;WITH ITS ADDRESS
  1004  001510  001401                         BEQ     3$
  1005  001512  104002                         HLT     2                       ;BYTE COUNT MEMORY ERROR
  1006  001514  005204                  3$:    INC     R4                      ;ADVANCE TO NEXT LOCATION
  1007  001516  062702  010421                 ADD     #10421,R2
  1008  001522  005300                         DEC     R0
  1009  001524  001364                         BNE     2$                      ;CONTINUE IF NOT DONE
  1010  001526  104400                  4$:    SCOPE
  1011
  1012                                         ;BUS ADDRESS MEMORY DATA TEST
  1013                                         ;VERIFY THAT ADDRESS 0 OF BUS ADDRESS MEMORY
  1014                                         ;CAN BE SET TO 177777 AND CLEARED TO 0
  1015
  1016  001530  012767  000340  176240  T3:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
  1017  001536  012767  000100  007500         MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
  1018  001544  012767  001624  007466         MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
  1019  001552  012705  177777                 MOV     #177777,R5              ;EXPECTED RESULT=177777
  1020  001556  012777  000000  007412         MOV     #0,@DHSCR               ;SELECT LOCATION 0
  1021                                                                          ;OF BUS ADDRESS MEMORY
  1022  001564  012777  177777  007412         MOV     #177777,@DHBA           ;WRITE 177777 INTO MEMORY
  1023  001572  017704  007406                 MOV     @DHBA,R4                ;READ CONTENTS OF MEMORY LOCATION
  1024  001576  020504                         CMP     R5,R4                   ;COMPARE EXPECTED AND
  1025  001600  001401                         BEQ     1$                      ;RECEIVED MEMORY CONTENTS
  1026  001602  104003                         HLT     3                       ;BUS ADDRESS MEMORY ERROR
  1027  001604  005005                  1$:    CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
  1028  001606  042777  177777  007370         BIC     #177777,@DHBA           ;CLEAR MEMORY LOCATION
  1029  001614  017704  007364                 MOV     @DHBA,R4                ;READ CONTENTS OF BUS ADDRESS
  1030                                                                          ;MEMORY ADDRESS 0
  1031  001620  001401                         BEQ     2$
  1032  001622  104003                         HLT     3                       ;BUS ADDRESS MEMORY ERROR
  1033                                                                          ;ADDRESS 0 NOT 0, ERROR
  1034  001624  104400                  2$:    SCOPE
  1035
  1036                                         ;BUS ADDRESS MEMORY DATA TEST
  1037                                         ;VERIFY THAT ADDRESS 1 OF BUS ADDRESS MEMORY
  1038                                         ;CAN BE SET TO 177777 AND CLEARED TO 0
  1039
  1040  001626  012767  000340  176142  T4:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
  1041  001634  012767  000100  007402         MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
  1042  001642  012767  001722  007370         MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
  1043  001650  012705  177777                 MOV     #177777,R5              ;EXPECTED RESULT=177777
  1044  001654  012777  000001  007314         MOV     #1,@DHSCR               ;SELECT LOCATION 1
  1045                                                                          ;OF BUS ADDRESS MEMORY
  1046  001662  012777  177777  007314         MOV     #177777,@DHBA           ;WRITE 177777 INTO MEMORY
  1047  001670  017704  007310                 MOV     @DHBA,R4                ;READ CONTENTS OF MEMORY LOCATION
```

```
1048  001674  020504                        CMP     R5,R4                   ;COMPARE EXPECTED AND
1049  001676  001401                        BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1050  001700  104003                        HLT     3                       ;BUS ADDRESS MEMORY ERROR
1051  001702  005005                1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1052  001704  042777  177777  007272        BIC     #177777,@DHBA           ;CLEAR MEMORY LOCATION
1053  001712  017704  007266                MOV     @DHBA,R4                ;READ CONTENTS OF BUS ADDRESS
1054                                                                        ;MEMORY ADDRESS 1
1055  001716  001401                        BEQ     2$
1056  001720  104003                        HLT     3                       ;BUS ADDRESS MEMORY ERROR
1057                                                                        ;ADDRESS 1 NOT 0, ERROR
1058  001722  104400                2$:     SCOPE
1059
1060                                        ;BUS ADDRESS MEMORY DATA TEST
1061                                        ;VERIFY THAT ADDRESS 2 OF BUS ADDRESS MEMORY
1062                                        ;CAN BE SET TO 177777 AND CLEARED TO 0
1063
1064  001724  012767  000340  176044  T5:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1065  001732  012767  000100  007304        MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1066  001740  012767  002020  007272        MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1067  001746  012705  177777                MOV     #177777,R5              ;EXPECTED RESULT=177777
1068  001752  012777  000002  007216        MOV     #2,@DHSCR               ;SELECT LOCATION 2
1069                                                                        ;OF BUS ADDRESS MEMORY
1070  001760  012777  177777  007216        MOV     #177777,@DHBA           ;WRITE 177777 INTO MEMORY
1071  001766  017704  007212                MOV     @DHBA,R4                ;READ CONTENTS OF MEMORY LOCATION
1072  001772  020504                        CMP     R5,R4                   ;COMPARE EXPECTED AND
1073  001774  001401                        BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1074  001776  104003                        HLT     3                       ;BUS ADDRESS MEMORY ERROR
1075  002000  005005                1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1076  002002  042777  177777  007174        BIC     #177777,@DHBA           ;CLEAR MEMORY LOCATION
1077  002010  017704  007170                MOV     @DHBA,R4                ;READ CONTENTS OF BUS ADDRESS
1078                                                                        ;MEMORY ADDRESS 2
1079  002014  001401                        BEQ     2$
1080  002016  104003                        HLT     3                       ;BUS ADDRESS MEMORY ERROR
1081                                                                        ;ADDRESS 2 NOT 0, ERROR
1082  002020  104400                2$:     SCOPE
1083
1084                                        ;BUS ADDRESS MEMORY DATA TEST
1085                                        ;VERIFY THAT ADDRESS 3 OF BUS ADDRESS MEMORY
1086                                        ;CAN BE SET TO 177777 AND CLEARED TO 0
1087
1088  002022  012767  000340  175746  T6:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1089  002030  012767  000100  007206        MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1090  002036  012767  002116  007174        MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1091  002044  012705  177777                MOV     #177777,R5              ;EXPECTED RESULT=177777
1092  002050  012777  000003  007120        MOV     #3,@DHSCR               ;SELECT LOCATION 3
1093                                                                        ;OF BUS ADDRESS MEMORY
1094  002056  012777  177777  007120        MOV     #177777,@DHBA           ;WRITE 177777 INTO MEMORY
1095  002064  017704  007114                MOV     @DHBA,R4                ;READ CONTENTS OF MEMORY LOCATION
1096  002070  020504                        CMP     R5,R4                   ;COMPARE EXPECTED AND
1097  002072  001401                        BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1098  002074  104003                        HLT     3                       ;BUS ADDRESS MEMORY ERROR
1099  002076  005005                1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1100  002100  042777  177777  007076        BIC     #177777,@DHBA           ;CLEAR MEMORY LOCATION
1101  002106  017704  007072                MOV     @DHBA,R4                ;READ CONTENTS OF BUS ADDRESS
1102                                                                        ;MEMORY ADDRESS 3
1103  002112  001401                        BEQ     2$
```

```
1104  002114  104003                          HLT     3                       ;BUS ADDRESS MEMORY ERROR
1105                                                                          ;ADDRESS 3 NOT 0, ERROR
1106  002116  104400                  2$:     SCOPE
1107
1108                                          ;BUS ADDRESS MEMORY DATA TEST
1109                                          ;VERIFY THAT ADDRESS 4 OF BUS ADDRESS MEMORY
1110                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1111
1112  002120  012767  000340  175650  T7:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1113  002126  012767  000100  007110          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1114  002134  012767  002214  007076          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1115  002142  012705  177777                  MOV     #177777,R5              ;EXPECTED RESULT=177777
1116  002146  012777  000004  007022          MOV     #4,@DHSCR               ;SELECT LOCATION 4
1117                                                                          ;OF BUS ADDRESS MEMORY
1118  002154  012777  177777  007022          MOV     #177777,@DHBA           ;WRITE 177777 INTO MEMORY
1119  002162  017704  007016                  MOV     @DHBA,R4                ;READ CONTENTS OF MEMORY LOCATION
1120  002166  020504                          CMP     R5,R4                   ;COMPARE EXPECTED AND
1121  002170  001401                          BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1122  002172  104003                          HLT     3                       ;BUS ADDRESS MEMORY ERROR
1123  002174  005005                  1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1124  002176  042777  177777  007000          BIC     #177777,@DHBA           ;CLEAR MEMORY LOCATION
1125  002204  017704  006774                  MOV     @DHBA,R4                ;READ CONTENTS OF BUS ADDRESS
1126                                                                          ;MEMORY ADDRESS 4
1127  002210  001401                          BEQ     2$
1128  002212  104003                          HLT     3                       ;BUS ADDRESS MEMORY ERROR
1129                                                                          ;ADDRESS 4 NOT 0, ERROR
1130  002214  104400                  2$:     SCOPE
1131
1132                                          ;BUS ADDRESS MEMORY DATA TEST
1133                                          ;VERIFY THAT ADDRESS 5 OF BUS ADDRESS MEMORY
1134                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1135
1136  002216  012767  000340  175552  T10:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1137  002224  012767  000100  007012          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1138  002232  012767  002312  007000          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1139  002240  012705  177777                  MOV     #177777,R5              ;EXPECTED RESULT=177777
1140  002244  012777  000005  006724          MOV     #5,@DHSCR               ;SELECT LOCATION 5
1141                                                                          ;OF BUS ADDRESS MEMORY
1142  002252  012777  177777  006724          MOV     #177777,@DHBA           ;WRITE 177777 INTO MEMORY
1143  002260  017704  006720                  MOV     @DHBA,R4                ;READ CONTENTS OF MEMORY LOCATION
1144  002264  020504                          CMP     R5,R4                   ;COMPARE EXPECTED AND
1145  002266  001401                          BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1146  002270  104003                          HLT     3                       ;BUS ADDRESS MEMORY ERROR
1147  002272  005005                  1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1148  002274  042777  177777  006702          BIC     #177777,@DHBA           ;CLEAR MEMORY LOCATION
1149  002302  017704  006676                  MOV     @DHBA,R4                ;READ CONTENTS OF BUS ADDRESS
1150                                                                          ;MEMORY ADDRESS 5
1151  002306  001401                          BEQ     2$
1152  002310  104003                          HLT     3                       ;BUS ADDRESS MEMORY ERROR
1153                                                                          ;ADDRESS 5 NOT 0, ERROR
1154  002312  104400                  2$:     SCOPE
1155
1156                                          ;BUS ADDRESS MEMORY DATA TEST
1157                                          ;VERIFY THAT ADDRESS 6 OF BUS ADDRESS MEMORY
1158                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1159
```

```
1160  002314  012767  000340  175454  T11:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1161  002322  012767  000100  006714          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1162  002330  012767  002410  006702          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1163  002336  012705  177777                  MOV     #177777,R5              ;EXPECTED RESULT=177777
1164  002342  012777  000006  006626          MOV     #6,@DHSCR               ;SELECT LOCATION 6
1165                                                                          ;OF BUS ADDRESS MEMORY
1166  002350  012777  177777  006626          MOV     #177777,@DHBA           ;WRITE 177777 INTO MEMORY
1167  002356  017704  006622                  MOV     @DHBA,R4                ;READ CONTENTS OF MEMORY LOCATION
1168  002362  020504                          CMP     R5,R4                   ;COMPARE EXPECTED AND
1169  002364  001401                          BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1170  002366  104003                          HLT     3                       ;BUS ADDRESS MEMORY ERROR
1171  002370  005005                  1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1172  002372  042777  177777  006604          BIC     #177777,@DHBA           ;CLEAR MEMORY LOCATION
1173  002400  017704  006600                  MOV     @DHBA,R4                ;READ CONTENTS OF BUS ADDRESS
1174                                                                          ;MEMORY ADDRESS 6
1175  002404  001401                          BEQ     2$
1176  002406  104003                          HLT     3                       ;BUS ADDRESS MEMORY ERROR
1177                                                                          ;ADDRESS 6 NOT 0, ERROR
1178  002410  104400                  2$:     SCOPE
1179
1180                                          ;BUS ADDRESS MEMORY DATA TEST
1181                                          ;VERIFY THAT ADDRESS 7 OF BUS ADDRESS MEMORY
1182                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1183
1184  002412  012767  000340  175356  T12:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1185  002420  012767  000100  006616          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1186  002426  012767  002506  006604          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1187  002434  012705  177777                  MOV     #177777,R5              ;EXPECTED RESULT=177777
1188  002440  012777  000007  006530          MOV     #7,@DHSCR               ;SELECT LOCATION 7
1189                                                                          ;OF BUS ADDRESS MEMORY
1190  002446  012777  177777  006530          MOV     #177777,@DHBA           ;WRITE 177777 INTO MEMORY
1191  002454  017704  006524                  MOV     @DHBA,R4                ;READ CONTENTS OF MEMORY LOCATION
1192  002460  020504                          CMP     R5,R4                   ;COMPARE EXPECTED AND
1193  002462  001401                          BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1194  002464  104003                          HLT     3                       ;BUS ADDRESS MEMORY ERROR
1195  002466  005005                  1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1196  002470  042777  177777  006506          BIC     #177777,@DHBA           ;CLEAR MEMORY LOCATION
1197  002476  017704  006502                  MOV     @DHBA,R4                ;READ CONTENTS OF BUS ADDRESS
1198                                                                          ;MEMORY ADDRESS 7
1199  002502  001401                          BEQ     2$
1200  002504  104003                          HLT     3                       ;BUS ADDRESS MEMORY ERROR
1201                                                                          ;ADDRESS 7 NOT 0, ERROR
1202  002506  104400                  2$:     SCOPE
1203
1204                                          ;BUS ADDRESS MEMORY DATA TEST
1205                                          ;VERIFY THAT ADDRESS 10 OF BUS ADDRESS MEMORY
1206                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1207
1208  002510  012767  000340  175260  T13:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1209  002516  012767  000100  006520          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1210  002524  012767  002604  006506          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1211  002532  012705  177777                  MOV     #177777,R5              ;EXPECTED RESULT=177777
1212  002536  012777  000010  006432          MOV     #10,@DHSCR              ;SELECT LOCATION 10
1213                                                                          ;OF BUS ADDRESS MEMORY
1214  002544  012777  177777  006432          MOV     #177777,@DHBA           ;WRITE 177777 INTO MEMORY
1215  002552  017704  006426                  MOV     @DHBA,R4                ,READ CONTENTS OF MEMORY LOCATION
```

```
1216  002556  020504                          CMP     R5,R4           ;COMPARE EXPECTED AND
1217  002560  001401                          BEQ     1$              ;RECEIVED MEMORY CONTENTS
1218  002562  104003                          HLT     3               ;BUS ADDRESS MEMORY ERROR
1219  002564  005005                  1$:     CLR     R5              ;EXPECTED RESULT AFTER CLEAR=0
1220  002566  042777  177777  006410          BIC     #177777,@DMBA   ;CLEAR MEMORY LOCATION
1221  002574  017704  006404                  MOV     @DMBA,R4        ;READ CONTENTS OF BUS ADDRESS
1222                                                                  ;MEMORY ADDRESS 10
1223  002600  001401                          BEQ     2$
1224  002602  104003                          HLT     3               ;BUS ADDRESS MEMORY ERROR
1225                                                                  ;ADDRESS 10 NOT 0, ERROR
1226  002604  104400                  2$:     SCOPE
1227
1228                                          ;BUS ADDRESS MEMORY DATA TEST
1229                                          ;VERIFY THAT ADDRESS 11 OF BUS ADDRESS MEMORY
1230                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1231
1232  002606  012767  000340  175162  T14:     MOV     #340,PS         ;DISABLE ALL INTERRUPTS
1233  002614  012767  000100  006422          MOV     #100,ICOUNT     ;SET UP FOR 100 ITERATIONS
1234  002622  012767  002702  006410          MOV     #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
1235  002630  012705  177777                  MOV     #177777,R5      ;EXPECTED RESULT=177777
1236  002634  012777  000011  006334          MOV     #11,@DMSCR      ;SELECT LOCATION 11
1237                                                                  ;OF BUS ADDRESS MEMORY
1238  002642  012777  177777  006334          MOV     #177777,@DMBA   ;WRITE 177777 INTO MEMORY
1239  002650  017704  006330                  MOV     @DMBA,R4        ;READ CONTENTS OF MEMORY LOCATION
1240  002654  020504                          CMP     R5,R4           ;COMPARE EXPECTED AND
1241  002656  001401                          BEQ     1$              ;RECEIVED MEMORY CONTENTS
1242  002660  104003                          HLT     3               ;BUS ADDRESS MEMORY ERROR
1243  002662  005005                  1$:     CLR     R5              ;EXPECTED RESULT AFTER CLEAR=0
1244  002664  042777  177777  006312          BIC     #177777,@DMBA   ;CLEAR MEMORY LOCATION
1245  002672  017704  006306                  MOV     @DMBA,R4        ;READ CONTENTS OF BUS ADDRESS
1246                                                                  ;MEMORY ADDRESS 11
1247  002676  001401                          BEQ     2$
1248  002700  104003                          HLT     3               ;BUS ADDRESS MEMORY ERROR
1249                                                                  ;ADDRESS 11 NOT 0, ERROR
1250  002702  104400                  2$:     SCOPE
1251
1252                                          ;BUS ADDRESS MEMORY DATA TEST
1253                                          ;VERIFY THAT ADDRESS 12 OF BUS ADDRESS MEMORY
1254                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1255
1256  002704  012767  000340  175064  T15:     MOV     #340,PS         ;DISABLE ALL INTERRUPTS
1257  002712  012767  000100  006324          MOV     #100,ICOUNT     ;SET UP FOR 100 ITERATIONS
1258  002720  012767  002730  006312          MOV     #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
1259  002726  012705  177777                  MOV     #177777,R5      ;EXPECTED RESULT=177777
1260  002732  012777  000012  006236          MOV     #12,@DMSCR      ;SELECT LOCATION 12
1261                                                                  ;OF BUS ADDRESS MEMORY
1262  002740  012777  177777  006236          MOV     #177777,@DMBA   ;WRITE 177777 INTO MEMORY
1263  002746  017704  006232                  MOV     @DMBA,R4        ;READ CONTENTS OF MEMORY LOCATION
1264  002752  020504                          CMP     R5,R4           ;COMPARE EXPECTED AND
1265  002754  001401                          BEQ     1$              ;RECEIVED MEMORY CONTENTS
1266  002756  104003                          HLT     3               ;BUS ADDRESS MEMORY ERROR
1267  002760  005005                  1$:     CLR     R5              ;EXPECTED RESULT AFTER CLEAR=0
1268  002762  042777  177777  006214          BIC     #177777,@DMBA   ;CLEAR MEMORY LOCATION
1269  002770  017704  006210                  MOV     @DMBA,R4        ;READ CONTENTS OF BUS ADDRESS
1270                                                                  ;MEMORY ADDRESS 12
1271  002774  001401                          BEQ     2$
```

```
1272  002776  104003                        HLT     3                       ;BUS ADDRESS MEMORY ERROR
1273                                                                          ;ADDRESS 12 NOT 0, ERROR
1274  003000  104400                2$:     SCOPE
1275
1276                                        ;BUS ADDRESS MEMORY DATA TEST
1277                                        ;VERIFY THAT ADDRESS 13 OF BUS ADDRESS MEMORY
1278                                        ;CAN BE SET TO 177777 AND CLEARED TO 0
1279
1280  003002  012767  000340  174766 T16:   MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1281  003010  012767  000100  006226       MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1282  003016  012767  003076  006214       MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1283  003024  012705  177777               MOV     #177777,R5              ;EXPECTED RESULT=177777
1284  003030  012777  000013  006140       MOV     #13,@DHSCR              ;SELECT LOCATION 13
1285                                                                        ;OF BUS ADDRESS MEMORY
1286  003036  012777  177777  006140       MOV     #177777,@DHBA           ;WRITE 177777 INTO MEMORY
1287  003044  017704  006134               MOV     @DHBA,R4                ;READ CONTENTS OF MEMORY LOCATION
1288  003050  020504                       CMP     R5,R4                   ;COMPARE EXPECTED AND
1289  003052  001401                       BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1290  003054  104003                       HLT     3                       ;BUS ADDRESS MEMORY ERROR
1291  003056  005005               1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1292  003060  042777  177777  006116       BIC     #177777,@DHBA           ;CLEAR MEMORY LOCATION
1293  003066  017704  006112               MOV     @DHBA,R4                ;READ CONTENTS OF BUS ADDRESS
1294                                                                        ;MEMORY ADDRESS 13
1295  003072  001401                       BEQ     2$
1296  003074  104003                       HLT     3                       ;BUS ADDRESS MEMORY ERROR
1297                                                                        ;ADDRESS 13 NOT 0, ERROR
1298  003076  104400                2$:     SCOPE
1299
1300                                        ;BUS ADDRESS MEMORY DATA TEST
1301                                        ;VERIFY THAT ADDRESS 14 OF BUS ADDRESS MEMORY
1302                                        ;CAN BE SET TO 177777 AND CLEARED TO 0
1303
1304  003100  012767  000340  174670 T17:   MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1305  003106  012767  000100  006130       MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1306  003114  012767  003174  006116       MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1307  003122  012705  177777               MOV     #177777,R5              ;EXPECTED RESULT=177777
1308  003126  012777  000014  006042       MOV     #14,@DHSCR              ;SELECT LOCATION 14
1309                                                                        ;OF BUS ADDRESS MEMORY
1310  003134  012777  177777  006042       MOV     #177777,@DHBA           ;WRITE 177777 INTO MEMORY
1311  003142  017704  006036               MOV     @DHBA,R4                ;READ CONTENTS OF MEMORY LOCATION
1312  003146  020504                       CMP     R5,R4                   ;COMPARE EXPECTED AND
1313  003150  001401                       BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1314  003152  104003                       HLT     3                       ;BUS ADDRESS MEMORY ERROR
1315  003154  005005               1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1316  003156  042777  177777  006020       BIC     #177777,@DHBA           ;CLEAR MEMORY LOCATION
1317  003164  017704  006014               MOV     @DHBA,R4                ;READ CONTENTS OF BUS ADDRESS
1318                                                                        ;MEMORY ADDRESS 14
1319  003170  001401                       BEQ     2$
1320  003172  104003                       HLT     3                       ;BUS ADDRESS MEMORY ERROR
1321                                                                        ;ADDRESS 14 NOT 0, ERROR
1322  003174  104400                2$:     SCOPE
1323
1324                                        ;BUS ADDRESS MEMORY DATA TEST
1325                                        ;VERIFY THAT ADDRESS 15 OF BUS ADDRESS MEMORY
1326                                        ;CAN BE SET TO 177777 AND CLEARED TO 0
1327
```

D03


```
1328  003176  012767  000340  174572  T20:    MOV     #340,PS          ;DISABLE ALL INTERRUPTS
1329  003204  012767  000100  005037          MOV     #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
1330  003212  012767  003272  006020          MOV     #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
1331  003220  012705  177777                  MOV     #177777,R5       ;EXPECTED RESULT=177777
1332  003224  012777  000015  005744          MOV     #15,@DHSCR       ;SELECT LOCATION 15
1333                                                                   ;OF BUS ADDRESS MEMORY
1334  003232  012777  177777  005744          MOV     #177777,@DHBA    ;WRITE 177777 INTO MEMORY
1335  003240  017704  005740                  MOV     @DHBA,R4         ;READ CONTENTS OF MEMORY LOCATION
1336  003244  020504                          CMP     R5,R4            ;COMPARE EXPECTED AND
1337  003246  001401                          BEQ     1$               ;RECEIVED MEMORY CONTENTS
1338  003250  104003                          HLT     3                ;BUS ADDRESS MEMORY ERROR
1339  003252  005005                  1$:     CLR     R5               ;EXPECTED RESULT AFTER CLEAR=0
1340  003254  042777  177777  005722          BIC     #177777,@DHBA    ;CLEAR MEMORY LOCATION
1341  003262  017704  005716                  MOV     @DHBA,R4         ;READ CONTENTS OF BUS ADDRESS
1342                                                                   ;MEMORY ADDRESS 15
1343  003266  001401                          BEQ     2$
1344  003270  104003                          HLT     3                ;BUS ADDRESS MEMORY ERROR
1345                                                                   ;ADDRESS IS NOT 0, ERROR
1346  003272  104400          2$:     SCOPE
1347
1348                          ;BUS ADDRESS MEMORY DATA TEST
1349                          ;VERIFY THAT ADDRESS 16 OF BUS ADDRESS MEMORY
1350                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1351
1352  003274  012767  000340  174474  T21:    MOV     #340,PS          ;DISABLE ALL INTERRUPTS
1353  003302  012767  000100  005734          MOV     #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
1354  003310  012767  003370  005722          MOV     #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
1355  003316  012705  177777                  MOV     #177777,R5       ;EXPECTED RESULT=177777
1356  003322  012777  000016  005646          MOV     #16,@DHSCR       ;SELECT LOCATION 16
1357                                                                   ;OF BUS ADDRESS MEMORY
1358  003330  012777  177777  005646          MOV     #177777,@DHBA    ;WRITE 177777 INTO MEMORY
1359  003336  017704  005642                  MOV     @DHBA,R4         ;READ CONTENTS OF MEMORY LOCATION
1360  003342  020504                          CMP     R5,R4            ;COMPARE EXPECTED AND
1361  003344  001401                          BEQ     1$               ;RECEIVED MEMORY CONTENTS
1362  003346  104003                          HLT     3                ;BUS ADDRESS MEMORY ERROR
1363  003350  005005                  1$:     CLR     R5               ;EXPECTED RESULT AFTER CLEAR=0
1364  003352  042777  177777  005624          BIC     #177777,@DHBA    ;CLEAR MEMORY LOCATION
1365  003360  017704  005620                  MOV     @DHBA,R4         ;READ CONTENTS OF BUS ADDRESS
1366                                                                   ;MEMORY ADDRESS 16
1367  003364  001401                          BEQ     2$
1368  003366  104003                          HLT     3                ;BUS ADDRESS MEMORY ERROR
1369                                                                   ;ADDRESS 16 NOT 0, ERROR
1370  003370  104400          2$:     SCOPE
1371
1372                          ;BUS ADDRESS MEMORY DATA TEST
1373                          ;VERIFY THAT ADDRESS 17 OF BUS ADDRESS MEMORY
1374                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1375
1376  003372  012767  000340  174376  T22:    MOV     #340,PS          ;DISABLE ALL INTERRUPTS
1377  003400  012767  000100  005636          MOV     #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
1378  003406  012767  003466  005624          MOV     #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
1379  003414  012705  177777                  MOV     #177777,R5       ;EXPECTED RESULT=177777
1380  003420  012777  000017  005550          MOV     #17,@DHSCR       ;SELECT LOCATION 17
1381                                                                   ;OF BUS ADDRESS MEMORY
1382  003426  012777  177777  005550          MOV     #177777,@DHBA    ;WRITE 177777 INTO MEMORY
1383  003434  017704  005544                  MOV     @DHBA,R4         ;READ CONTENTS OF MEMORY LOCATION
```

```
1384  003440  020504                        CMP     R5,R4           ;COMPARE EXPECTED AND
1385  003442  001401                        BEQ     1$              ;RECEIVED MEMORY CONTENTS
1386  003444  104003                        HLT     3               ;BUS ADDRESS MEMORY ERROR
1387  003446  005005                1$:     CLR     R5              ;EXPECTED RESULT AFTER CLEAR=0
1388  003450  042777  177777  005526        BIC     #177777,@DHBA   ;CLEAR MEMORY LOCATION
1389  003456  017704  005522                MOV     @DHBA,R4        ;READ CONTENTS OF BUS ADDRESS
1390                                                                ;MEMORY ADDRESS 17
1391  003462  001401                        BEQ     2$
1392  003464  104003                        HLT     3               ;BUS ADDRESS MEMORY ERROR
1393                                                                ;ADDRESS 17 NOT 0, ERROR
1394  003466  104400                2$:     SCOPE
1395
1396                                        ;BYTE COUNT MEMORY DATA TEST
1397                                        ;VERIFY THAT ADDRESS 0 OF BYTE COUNT MEMORY
1398                                        ;CAN BE SET TO 177777 AND CLEARED TO 0
1399
1400  003470  012767  000340  174300  T23:   MOV     #340,PS         ;DISABLE ALL INTERRUPTS
1401  003476  012767  000100  005540        MOV     #100,ICOUNT     ;SET UP FOR 100 ITERATIONS
1402  003504  012767  003564  005526        MOV     #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
1403  003512  012705  177777                MOV     #177777,R5      ;EXPECTED RESULT=177777
1404  003516  012777  000000  005452        MOV     #0,@DHSCR       ;SELECT LOCATION 0
1405                                                                ;OF BYTE COUNT MEMORY
1406  003524  012777  177777  005454        MOV     #177777,@DHBC   ;WRITE 177777 INTO MEMORY
1407  003532  017704  005450                MOV     @DHBC,R4        ;READ CONTENTS OF MEMORY LOCATION
1408  003536  020504                        CMP     R5,R4           ;COMPARE EXPECTED AND
1409  003540  001401                        BEQ     1$              ;RECEIVED MEMORY CONTENTS
1410  003542  104004                        HLT     4               ;BYTE COUNT MEMORY ERROR
1411  003544  005005                1$:     CLR     R5              ;EXPECTED RESULT AFTER CLEAR=0
1412  003546  042777  177777  005432        BIC     #177777,@DHBC   ;CLEAR MEMORY LOCATION
1413  003554  017704  005426                MOV     @DHBC,R4        ;READ CONTENTS OF BYTE COUNT
1414                                                                ;MEMORY ADDRESS 0
1415  003560  001401                        BEQ     2$
1416  003562  104004                        HLT     4               ;BYTE COUNT MEMORY ERROR
1417                                                                ;ADDRESS 0 NOT 0, ERROR
1418  003564  104400                2$:     SCOPE
1419
1420                                        ;BYTE COUNT MEMORY DATA TEST
1421                                        ;VERIFY THAT ADDRESS 1 OF BYTE COUNT MEMORY
1422                                        ;CAN BE SET TO 177777 AND CLEARED TO 0
1423
1424  003566  012767  000340  174202  T24:   MOV     #340,PS         ;DISABLE ALL INTERRUPTS
1425  003574  012767  000100  005442        MOV     #100,ICOUNT     ;SET UP FOR 100 ITERATIONS
1426  003602  012767  003642  005430        MOV     #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
1427  003610  012705  177777                MOV     #177777,R5      ;EXPECTED RESULT=177777
1428  003614  012777  000001  005354        MOV     #1,@DHSCR       ;SELECT LOCATION 1
1429                                                                ;OF BYTE COUNT MEMORY
1430  003622  012777  177777  005356        MOV     #177777,@DHBC   ;WRITE 177777 INTO MEMORY
1431  003630  017704  005332                MOV     @DHBC,R4        ;READ CONTENTS OF MEMORY LOCATION
1432  003634  020504                        CMP     R5,R4           ;COMPARE EXPECTED AND
1433  003636  001401                        BEQ     1$              ;RECEIVED MEMORY CONTENTS
1434  003640  104004                        HLT     4               ;BYTE COUNT MEMORY ERROR
1435  003642  005005                1$:     CLR     R5              ;EXPECTED RESULT AFTER CLEAR=0
1436  003644  042777  177777  005334        BIC     #177777,@DHBC   ;CLEAR MEMORY LOCATION
1437  003652  017704  005330                MOV     @DHBC,R4        ;READ CONTENTS OF BYTE COUNT
1438                                                                ;MEMORY ADDRESS 1
1439  003656  001401                        BEQ     2$
```

F03


```
1440  003660  104304                          HLT     4                       ;BYTE COUNT MEMORY ERROR
1441                                                                          ;ADDRESS 1 NOT 0, ERROR
1442  003662  104400                  2$:     SCOPE
1443
1444                                          ;BYTE COUNT MEMORY DATA TEST
1445                                          ;VERIFY THAT ADDRESS 2 OF BYTE COUNT MEMORY
1446                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1447
1448  003664  012767  000340  174104  T2$:   MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1449  003672  012767  000100  005344        MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1450  003700  012767  003760  005332        MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1451  003706  012705  177777                MOV     #177777,R5              ;EXPECTED RESULT=177777
1452  003712  012777  000002  005256        MOV     #2,@DHSCR               ;SELECT LOCATION 2
1453                                                                          ;OF BYTE COUNT MEMORY
1454  003720  012777  177777  005260        MOV     #177777,@DHBC           ;WRITE 177777 INTO MEMORY
1455  003726  017704  005254                MOV     @DHBC,R4                ;READ CONTENTS OF MEMORY LOCATION
1456  003732  020504                        CMP     R5,R4                   ;COMPARE EXPECTED AND
1457  003734  001401                        BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1458  003736  104004                        HLT     4                       ;BYTE COUNT MEMORY ERROR
1459  003740  005005                1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1460  003742  042777  177777  005236        BIC     #177777,@DHBC           ;CLEAR MEMORY LOCATION
1461  003750  017704  005232                MOV     @DHBC,R4                ;READ CONTENTS OF BYTE COUNT
1462                                                                          ;MEMORY ADDRESS 2
1463  003754  001401                        BEQ     2$
1464  003756  104004                        HLT     4                       ;BYTE COUNT MEMORY ERROR
1465                                                                          ;ADDRESS 2 NOT 0, ERROR
1466  003760  104400                  2$:     SCOPE
1467
1468                                          ;BYTE COUNT MEMORY DATA TEST
1469                                          ;VERIFY THAT ADDRESS 3 OF BYTE COUNT MEMORY
1470                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1471
1472  003762  012767  000340  174006  T2$:   MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1473  003770  012767  000100  005246        MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1474  003776  012767  004056  005234        MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1475  004004  012705  177777                MOV     #177777,R5              ;EXPECTED RESULT=177777
1476  004010  012777  000003  005160        MOV     #3,@DHSCR               ;SELECT LOCATION 3
1477                                                                          ;OF BYTE COUNT MEMORY
1478  004016  012777  177777  005162        MOV     #177777,@DHBC           ;WRITE 177777 INTO MEMORY
1479  004024  017704  005156                MOV     @DHBC,R4                ;READ CONTENTS OF MEMORY LOCATION
1480  004030  020504                        CMP     R5,R4                   ;COMPARE EXPECTED AND
1481  004032  001401                        BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1482  004034  104004                        HLT     4                       ;BYTE COUNT MEMORY ERROR
1483  004036  005005                1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1484  004040  042777  177777  005140        BIC     #177777,@DHBC           ;CLEAR MEMORY LOCATION
1485  004046  017704  005134                MOV     @DHBC,R4                ;READ CONTENTS OF BYTE COUNT
1486                                                                          ;MEMORY ADDRESS 3
1487  004052  001401                        BEQ     2$
1488  004054  104004                        HLT     4                       ;BYTE COUNT MEMORY ERROR
1489                                                                          ;ADDRESS 3 NOT 0, ERROR
1490  004056  104400                  2$:     SCOPE
1491
1492                                          ;BYTE COUNT MEMORY DATA TEST
1493                                          ;VERIFY THAT ADDRESS 4 OF BYTE COUNT MEMORY
1494                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1495
```

```
1496  004060  012767  000340  173710  T27:  MOV   #340,PS          ;DISABLE ALL INTERRUPTS
1497  004066  012767  000100  005150        MOV   #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
1498  004074  012767  004154  005136        MOV   #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
1499  004102  012705  177777                MOV   #177777,R5       ;EXPECTED RESULT=177777
1500  004106  012777  000004  005062        MOV   #4,@DHSCR        ;SELECT LOCATION 4
1501                                                               ;OF BYTE COUNT MEMORY
1502  004114  012777  177777  005064        MOV   #177777,@DHBC    ;WRITE 177777 INTO MEMORY
1503  004122  017704  005060                MOV   @DHBC,R4         ;READ CONTENTS OF MEMORY LOCATION
1504  004126  020504                        CMP   R5,R4            ;COMPARE EXPECTED AND
1505  004130  001401                        BEQ   1$               ;RECEIVED MEMORY CONTENTS
1506  004132  104004                        HLT   4                ;BYTE COUNT MEMORY ERROR
1507  004134  005005                1$:     CLR   R5               ;EXPECTED RESULT AFTER CLEAR=0
1508  004136  042777  177777  005042        BIC   #177777,@DHBC    ;CLEAR MEMORY LOCATION
1509  004144  017704  005036                MOV   @DHBC,R4         ;READ CONTENTS OF BYTE COUNT
1510                                                               ;MEMORY ADDRESS 4
1511  004150  001401                        BEQ   2$
1512  004152  104004                        HLT   4                ;BYTE COUNT MEMORY ERROR
1513                                                               ;ADDRESS 4 NOT 0, ERROR
1514  004154  104400                2$:     SCOPE
1515
1516                                        ;BYTE COUNT MEMORY DATA TEST
1517                                        ;VERIFY THAT ADDRESS 5 OF BYTE COUNT MEMORY
1518                                        ;CAN BE SET TO 177777 AND CLEARED TO 0
1519
1520  004156  012767  000340  173612  T30:  MOV   #340,PS          ;DISABLE ALL INTERRUPTS
1521  004164  012767  000100  005052        MOV   #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
1522  004172  012767  004252  005040        MOV   #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
1523  004200  012705  177777                MOV   #177777,R5       ;EXPECTED RESULT=177777
1524  004204  012777  000005  004764        MOV   #5,@DHSCR        ;SELECT LOCATION 5
1525                                                               ;OF BYTE COUNT MEMORY
1526  004212  012777  177777  004766        MOV   #177777,@DHBC    ;WRITE 177777 INTO MEMORY
1527  004220  017704  004762                MOV   @DHBC,R4         ;READ CONTENTS OF MEMORY LOCATION
1528  004224  020504                        CMP   R5,R4            ;COMPARE EXPECTED AND
1529  004226  001401                        BEQ   1$               ;RECEIVED MEMORY CONTENTS
1530  004230  104004                        HLT   4                ;BYTE COUNT MEMORY ERROR
1531  004232  005005                1$:     CLR   R5               ;EXPECTED RESULT AFTER CLEAR=0
1532  004234  042777  177777  004744        BIC   #177777,@DHBC    ;CLEAR MEMORY LOCATION
1533  004242  017704  004740                MOV   @DHBC,R4         ;READ CONTENTS OF BYTE COUNT
1534                                                               ;MEMORY ADDRESS 5
1535  004246  001401                        BEQ   2$
1536  004250  104004                        HLT   4                ;BYTE COUNT MEMORY ERROR
1537                                                               ;ADDRESS 5 NOT 0, ERROR
1538  004252  104400                2$:     SCOPE
1539
1540                                        ;BYTE COUNT MEMORY DATA TEST
1541                                        ;VERIFY THAT ADDRESS 6 OF BYTE COUNT MEMORY
1542                                        ;CAN BE SET TO 177777 AND CLEARED TO 0
1543
1544  004254  012767  000340  173514  T31:  MOV   #340,PS          ;DISABLE ALL INTERRUPTS
1545  004262  012767  000100  004754        MOV   #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
1546  004270  012767  004300  004742        MOV   #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
1547  004276  012705  177777                MOV   #177777,R5       ;EXPECTED RESULT=177777
1548  004302  012777  000006  004666        MOV   #6,@DHSCR        ;SELECT LOCATION 6
1549                                                               ;OF BYTE COUNT MEMORY
1550  004310  012777  177777  004670        MOV   #177777,@DHBC    ;WRITE 177777 INTO MEMORY
1551  004316  017704  004664                MOV   @DHBC,R4         ;READ CONTENTS OF MEMORY LOCATION
```

```
1552  004322  020504                          CMP     R5,R4           ;COMPARE EXPECTED AND
1553  004324  001401                          BEQ     1$              ;RECEIVED MEMORY CONTENTS
1554  004326  104004                          HLT     4               ;BYTE COUNT MEMORY ERROR
1555  004330  005005                  1$:     CLR     R5              ;EXPECTED RESULT AFTER CLEAR=0
1556  004332  042777  177777  004646          BIC     #177777,@DHBC   ;CLEAR MEMORY LOCATION
1557  004340  017704  004642                  MOV     @DHBC,R4        ;READ CONTENTS OF BYTE COUNT
1558                                                                  ;MEMORY ADDRESS 6
1559  004344  001401                          BEQ     2$
1560  004346  104004                          HLT     4               ;BYTE COUNT MEMORY ERROR
1561                                                                  ;ADDRESS 6 NOT 0, ERROR
1562  004350  104400                  2$:     SCOPE
1563
1564                                          ;BYTE COUNT MEMORY DATA TEST
1565                                          ;VERIFY THAT ADDRESS 7 OF BYTE COUNT MEMORY
1566                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1567
1568  004352  012767  000340  173416  T32:    MOV     #340,PS         ;DISABLE ALL INTERRUPTS
1569  004360  012767  000100  004656          MOV     #100,ICOUNT     ;SET UP FOR 100 ITERATIONS
1570  004366  012767  004446  004644          MOV     #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
1571  004374  012705  177777                  MOV     #177777,R5      ;EXPECTED RESULT=177777
1572  004400  012777  000007  004570          MOV     #7,@DHSCR       ;SELECT LOCATION 7
1573                                                                  ;OF BYTE COUNT MEMORY
1574  004406  012777  177777  004572          MOV     #177777,@DHBC   ;WRITE 177777 INTO MEMORY
1575  004414  017704  004566                  MOV     @DHBC,R4        ;READ CONTENTS OF MEMORY LOCATION
1576  004420  020504                          CMP     R5,R4           ;COMPARE EXPECTED AND
1577  004422  001401                          BEQ     1$              ;RECEIVED MEMORY CONTENTS
1578  004424  104004                          HLT     4               ;BYTE COUNT MEMORY ERROR
1579  004426  005005                  1$:     CLR     R5              ;EXPECTED RESULT AFTER CLEAR=0
1580  004430  042777  177777  004550          BIC     #177777,@DHBC   ;CLEAR MEMORY LOCATION
1581  004436  017704  004544                  MOV     @DHBC,R4        ;READ CONTENTS OF BYTE COUNT
1582                                                                  ;MEMORY ADDRESS 7
1583  004442  001401                          BEQ     2$
1584  004444  104004                          HLT     4               ;BYTE COUNT MEMORY ERROR
1585                                                                  ;ADDRESS 7 NOT 0, ERROR
1586  004446  104400                  2$:     SCOPE
1587
1588                                          ;BYTE COUNT MEMORY DATA TEST
1589                                          ;VERIFY THAT ADDRESS 10 OF BYTE COUNT MEMORY
1590                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1591
1592  004450  012767  000340  173320  T33:    MOV     #340,PS         ;DISABLE ALL INTERRUPTS
1593  004456  012767  000100  004560          MOV     #100,ICOUNT     ;SET UP FOR 100 ITERATIONS
1594  004464  012767  004544  004546          MOV     #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
1595  004472  012705  177777                  MOV     #177777,R5      ;EXPECTED RESULT=177777
1596  004476  012777  000010  004472          MOV     #10,@DHSCR      ;SELECT LOCATION 10
1597                                                                  ;OF BYTE COUNT MEMORY
1598  004504  012777  177777  004474          MOV     #177777,@DHBC   ;WRITE 177777 INTO MEMORY
1599  004512  017704  004470                  MOV     @DHBC,R4        ;READ CONTENTS OF MEMORY LOCATION
1600  004516  020504                          CMP     R5,R4           ;COMPARE EXPECTED AND
1601  004520  001401                          BEQ     1$              ;RECEIVED MEMORY CONTENTS
1602  004522  104004                          HLT     4               ;BYTE COUNT MEMORY ERROR
1603  004524  005005                  1$:     CLR     R5              ;EXPECTED RESULT AFTER CLEAR=0
1604  004526  042777  177777  004452          BIC     #177777,@DHBC   ;CLEAR MEMORY LOCATION
1605  004534  017704  004446                  MOV     @DHBC,R4        ;READ CONTENTS OF BYTE COUNT
1606                                                                  ;MEMORY ADDRESS 10
1607  004540  001401                          BEQ     2$
```

```
1608  004542  104004                        HLT     4                       ;BYTE COUNT MEMORY ERROR
1609                                                                          ;ADDRESS 10 NOT 0, ERROR
1610  004544  104400                2$:     SCOPE
1611
1612                                        ;BYTE COUNT MEMORY DATA TEST
1613                                        ;VERIFY THAT ADDRESS 11 OF BYTE COUNT MEMORY
1614                                        ;CAN BE SET TO 177777 AND CLEARED TO 0
1615
1616  004546  012767  000340  173222  T34:  MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1617  004554  012767  000100  004462        MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1618  004562  012767  004642  004450        MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1619  004570  012705  177777                MOV     #177777,R5              ;EXPECTED RESULT=177777
1620  004574  012777  000011  004374        MOV     #11,@DHSCR              ;SELECT LOCATION 11
1621                                                                          ;OF BYTE COUNT MEMORY
1622  004602  012777  177777  004376        MOV     #177777,@DHBC           ;WRITE 177777 INTO MEMORY
1623  004610  017704  004372                MOV     @DHBC,R4                ;READ CONTENTS OF MEMORY LOCATION
1624  004614  020504                        CMP     R5,R4                   ;COMPARE EXPECTED AND
1625  004616  001401                        BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1626  004620  104004                        HLT     4                       ;BYTE COUNT MEMORY ERROR
1627  004622  005005                1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1628  004624  042777  177777  004354        BIC     #177777,@DHBC           ;CLEAR MEMORY LOCATION
1629  004632  017704  004350                MOV     @DHBC,R4                ;READ CONTENTS OF BYTE COUNT
1630                                                                          ;MEMORY ADDRESS 11
1631  004636  001401                        BEQ     2$
1632  004640  104004                        HLT     4                       ;BYTE COUNT MEMORY ERROR
1633                                                                          ;ADDRESS 11 NOT 0, ERROR
1634  004642  104400                2$:     SCOPE
1635
1636                                        ;BYTE COUNT MEMORY DATA TEST
1637                                        ;VERIFY THAT ADDRESS 12 OF BYTE COUNT MEMORY
1638                                        ;CAN BE SET TO 177777 AND CLEARED TO 0
1639
1640  004644  012767  000340  173124  T35:  MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1641  004652  012767  000100  004364        MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1642  004660  012767  004740  004352        MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1643  004666  012705  177777                MOV     #177777,R5              ;EXPECTED RESULT=177777
1644  004672  012777  000012  004276        MOV     #12,@DHSCR              ;SELECT LOCATION 12
1645                                                                          ;OF BYTE COUNT MEMORY
1646  004700  012777  177777  004300        MOV     #177777,@DHBC           ;WRITE 177777 INTO MEMORY
1647  004706  017704  004274                MOV     @DHBC,R4                ;READ CONTENTS OF MEMORY LOCATION
1648  004712  020504                        CMP     R5,R4                   ;COMPARE EXPECTED AND
1649  004714  001401                        BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1650  004716  104004                        HLT     4                       ;BYTE COUNT MEMORY ERROR
1651  004720  005005                1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1652  004722  042777  177777  004256        BIC     #177777,@DHBC           ;CLEAR MEMORY LOCATION
1653  004730  017704  004252                MOV     @DHBC,R4                ;READ CONTENTS OF BYTE COUNT
1654                                                                          ;MEMORY ADDRESS 12
1655  004734  001401                        BEQ     2$
1656  004736  104004                        HLT     4                       ;BYTE COUNT MEMORY ERROR
1657                                                                          ;ADDRESS 12 NOT 0, ERROR
1658  004740  104400                2$:     SCOPE
1659
1660                                        ;BYTE COUNT MEMORY DATA TEST
1661                                        ;VERIFY THAT ADDRESS 13 OF BYTE COUNT MEMORY
1662                                        ;CAN BE SET TO 177777 AND CLEARED TO 0
1663
```

```
1664  004742  012767  000340  173026  T36:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1665  004750  012767  000100  004266          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1666  004756  012767  005036  004254          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1667  004764  012705  177777                  MOV     #177777,R5              ;EXPECTED RESULT=177777
1668  004770  012777  000013  004200          MOV     #13,@DHSCR              ;SELECT LOCATION 13
1669                                                                          ;OF BYTE COUNT MEMORY
1670  004776  012777  177777  004202          MOV     #177777,@DHBC           ;WRITE 177777 INTO MEMORY
1671  005004  017704  004176                  MOV     @DHBC,R4                ;READ CONTENTS OF MEMORY LOCATION
1672  005010  020504                          CMP     R5,R4                   ;COMPARE EXPECTED AND
1673  005012  001401                          BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1674  005014  104004                          HLT     4                       ;BYTE COUNT MEMORY ERROR
1675  005016  005005                  1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1676  005020  042777  177777  004160          BIC     #177777,@DHBC           ;CLEAR MEMORY LOCATION
1677  005026  017704  004154                  MOV     @DHBC,R4                ;READ CONTENTS OF BYTE COUNT
1678                                                                          ;MEMORY ADDRESS 13
1679  005032  001401                          BEQ     2$
1680  005034  104004                          HLT     4                       ;BYTE COUNT MEMORY ERROR
1681                                                                          ;ADDRESS 13 NOT 0, ERROR
1682  005036  104400                  2$:     SCOPE
1683
1684                                          ;BYTE COUNT MEMORY DATA TEST
1685                                          ;VERIFY THAT ADDRESS 14 OF BYTE COUNT MEMORY
1686                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1687
1688  005040  012767  000340  172730  T37:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1689  005046  012767  000100  004170          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1690  005054  012767  005134  004156          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1691  005062  012705  177777                  MOV     #177777,R5              ;EXPECTED RESULT=177777
1692  005066  012777  000014  004102          MOV     #14,@DHSCR              ;SELECT LOCATION 14
1693                                                                          ;OF BYTE COUNT MEMORY
1694  005074  012777  177777  004104          MOV     #177777,@DHBC           ;WRITE 177777 INTO MEMORY
1695  005102  017704  004100                  MOV     @DHBC,R4                ;READ CONTENTS OF MEMORY LOCATION
1696  005106  020504                          CMP     R5,R4                   ;COMPARE EXPECTED AND
1697  005110  001401                          BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1698  005112  104004                          HLT     4                       ;BYTE COUNT MEMORY ERROR
1699  005114  005005                  1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1700  005116  042777  177777  004062          BIC     #177777,@DHBC           ;CLEAR MEMORY LOCATION
1701  005124  017704  004056                  MOV     @DHBC,R4                ;READ CONTENTS OF BYTE COUNT
1702                                                                          ;MEMORY ADDRESS 14
1703  005130  001401                          BEQ     2$
1704  005132  104004                          HLT     4                       ;BYTE COUNT MEMORY ERROR
1705                                                                          ;ADDRESS 14 NOT 0, ERROR
1706  005134  104400                  2$:     SCOPE
1707
1708                                          ;BYTE COUNT MEMORY DATA TEST
1709                                          ;VERIFY THAT ADDRESS 15 OF BYTE COUNT MEMORY
1710                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1711
1712  005136  012767  000340  172632  T40:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1713  005144  012767  000100  004072          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1714  005152  012767  005232  004060          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1715  005160  012705  177777                  MOV     #177777,R5              ;EXPECTED RESULT=177777
1716  005164  012777  000015  004004          MOV     #15,@DHSCR              ;SELECT LOCATION 15
1717                                                                          ;OF BYTE COUNT MEMORY
1718  005172  012777  177777  004006          MOV     #177777,@DHBC           ;WRITE 177777 INTO MEMORY
1719  005200  017704  004002                  MOV     @DHBC,R4                ;READ CONTENTS OF MEMORY LOCATION
```

```
1720  005204  020504                          CMP     R5,R4                   ;COMPARE EXPECTED AND
1721  005206  001401                          BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1722  005210  104004                          HLT     4                       ;BYTE COUNT MEMORY ERROR
1723  005212  005005                  1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1724  005214  042777  177777  003764          BIC     #177777,@DHBC           ;CLEAR MEMORY LOCATION
1725  005222  017704  003760                  MOV     @DHBC,R4                ;READ CONTENTS OF BYTE COUNT
1726                                                                          ;MEMORY ADDRESS 15
1727  005226  001401                          BEQ     2$
1728  005230  104004                          HLT     4                       ;BYTE COUNT MEMORY ERROR
1729                                                                          ;ADDRESS 15 NOT 0, ERROR
1730  005232  104400                  2$:     SCOPE
1731
1732                                          ;BYTE COUNT MEMORY DATA TEST
1733                                          ;VERIFY THAT ADDRESS 16 OF BYTE COUNT MEMORY
1734                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1735
1736  005234  012767  000340  172534  T41:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1737  005242  012767  000100  003774          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1738  005250  012767  005330  003762          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1739  005256  012705  177777                  MOV     #177777,R5              ;EXPECTED RESULT=177777
1740  005262  012777  000016  003706          MOV     #16,@DHSCR              ;SELECT LOCATION 16
1741                                                                          ;OF BYTE COUNT MEMORY
1742  005270  012777  177777  003710          MOV     #177777,@DHBC           ;WRITE 177777 INTO MEMORY
1743  005276  017704  003704                  MOV     @DHBC,R4                ;READ CONTENTS OF MEMORY LOCATION
1744  005302  020504                          CMP     R5,R4                   ;COMPARE EXPECTED AND
1745  005304  001401                          BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1746  005306  104004                          HLT     4                       ;BYTE COUNT MEMORY ERROR
1747  005310  005005                  1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1748  005312  042777  177777  003666          BIC     #177777,@DHBC           ;CLEAR MEMORY LOCATION
1749  005320  017704  003662                  MOV     @DHBC,R4                ;READ CONTENTS OF BYTE COUNT
1750                                                                          ;MEMORY ADDRESS 16
1751  005324  001401                          BEQ     2$
1752  005326  104004                          HLT     4                       ;BYTE COUNT MEMORY ERROR
1753                                                                          ;ADDRESS 16 NOT 0, ERROR
1754  005330  104400                  2$:     SCOPE
1755
1756                                          ;BYTE COUNT MEMORY DATA TEST
1757                                          ;VERIFY THAT ADDRESS 17 OF BYTE COUNT MEMORY
1758                                          ;CAN BE SET TO 177777 AND CLEARED TO 0
1759
1760  005332  012767  000340  172436  T42:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1761  005340  012767  000100  003676          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1762  005346  012767  005426  003664          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1763  005354  012705  177777                  MOV     #177777,R5              ;EXPECTED RESULT=177777
1764  005360  012777  000017  003610          MOV     #17,@DHSCR              ;SELECT LOCATION 17
1765                                                                          ;OF BYTE COUNT MEMORY
1766  005366  012777  177777  003612          MOV     #177777,@DHBC           ;WRITE 177777 INTO MEMORY
1767  005374  017704  003606                  MOV     @DHBC,R4                ;READ CONTENTS OF MEMORY LOCATION
1768  005400  020504                          CMP     R5,R4                   ;COMPARE EXPECTED AND
1769  005402  001401                          BEQ     1$                      ;RECEIVED MEMORY CONTENTS
1770  005404  104004                          HLT     4                       ;BYTE COUNT MEMORY ERROR
1771  005406  005005                  1$:     CLR     R5                      ;EXPECTED RESULT AFTER CLEAR=0
1772  005410  042777  177777  003570          BIC     #177777,@DHBC           ;CLEAR MEMORY LOCATION
1773  005416  017704  003564                  MOV     @DHBC,R4                ;READ CONTENTS OF BYTE COUNT
1774                                                                          ;MEMORY ADDRESS 17
1775  005422  001401                          BEQ     2$
```

```
1776  005424  104004                          HLT     4                       ;BYTE COUNT MEMORY EPROR
1777                                                                          ;ADDRESS 17 NOT 0, ERROR
1778  005426  104400                  2$:     SCOPE
1779
1780                                          ;BUS ADDRESS MEMORY TEST
1781                                          ;CLEAR ALL LOCATIONS IN BUS ADDRESS MEMORY
1782                                          ;SET SELECTED LOCATION TO VALUE 177777
1783                                          ;VERIFY THAT SELECTED LOCATION WAS SET
1784                                          ;TO 177777.
1785                                          ;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.
1786
1787  005430  012767  000340  172[illegible]  3$:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1788  005436  012767  000100  005[illegible]          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
1789  005444  012767  005602  003[illegible]          MOV     #6$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1790  005452  012767  005476  003[illegible]          MOV     #2$,FREEZI              ;SET UP TO LOOP WITH DATA
1791  005460  012700  000020                  MOV     #20,R0                  ;SET UP TO TEST 20(OCTAL)
1792                                                                          ;LOCATIONS IN BUS ADDRESS MEMORY
1793  005464  005003                          CLR     R3                      ;FIRST LOCATION TO BE
1794                                                                          ;WRITTEN INTO IS 0
1795  005466  012701  000020          1$:     MOV     #20,R1                  ;SET UP TO CLEAR 20 (OCTAL)
1796                                                                          ;LOCATIONS IN BUS ADDRESS MEMORY
1797  005472  005077  003500                  CLR     @DHSCR                  ;START AT LOCATION 0
1798  005476  005077  003502          2$:     CLR     @DHBA                   ;CLEAR LOCATION IN
1799                                                                          ;BUS ADDRESS MEMORY
1800  005502  005277  003470                  INC     @DHSCR                  ;ADVANCE TO NEXT LOCATION
1801  005506  005301                          DEC     R1                      ;CONTINUE CLEARING
1802  005510  001372                          BNE     2$                      ;IF NOT DONE
1803  005512  010377  003460                  MOV     R3,@DHSCR               ;SELECT ADDRESS TO BE TESTED
1804  005516  012777  177777  003460          MOV     #177777,@DHBA           ;WRITE 177777 INTO LOCATION
1805  005524  005077  003446                  CLR     @DHSCR                  ;ADDRESS LOCATION 0
1806  005530  012701  000020                  MOV     #20,R1                  ;SET UP TO CHECK ALL ADDRESSES
1807                                                                          ;IN BUS ADDRESS MEMORY
1808  005534  012705  177777          3$:     MOV     #177777,R5              ;177777=EXPECTED RESULT
1809                                                                          ;IF ADDRESS READ IS LOCATION
1810                                                                          ;WRITTEN INTO
1811  005540  017704  003440                  MOV     @DHBA,R4                ;READ MEMORY LOCATION
1812  005544  027703  003426                  CMP     @DHSCR,R3               ;IF LINE NUMBER=ADDRESS
1813                                                                          ;OF LOCATION WRITTEN INTO
1814                                                                          ;EXPECTED CONTENTS=177777
1815  005550  001401                          BEQ     4$
1816  005552  005005                          CLR     R5                      ;OTHERWISE, EXPECTED RESULTS=0
1817  005554  020504          4$:     CMP     R5,R4                   ;DOES MEMORY LOCATION CONTAIN
1818  005556  001401                          BEQ     5$                      ;EXPECTED RESULT
1819  005560  104003                          HLT     3                       ;BUS ADDRESS MEMORY ERROR
1820  005562  104410          5$:     SCOPE1                          ;CHECK FOR LOOP WITH CURRENT DATA
1821  005564  005277  003406                  INC     @DHSCR                  ;CHECK CONTENTS OF NEXT LOCATION
1822  005570  005301                          DEC     R1
1823  005572  001360                          BNE     3$
1824  005574  005203                          INC     R3                      ;NEXT ADDRESS TO BE WRITTEN
1825  005576  005300                          DEC     R0
1826  005600  001332                          BNE     1$
1827  005602  104400          6$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1828
1829                                          ;BUS ADDRESS MEMORY TEST
1830                                          ;CLEAR ALL LOCATIONS IN BUS ADDRESS MEMORY
1831                                          ;SET SELECTED LOCATION TO VALUE 125252
```

```
 1832                                                ;VERIFY THAT SELECTED LOCATION WAS SET
 1833                                                ;TO 125252.
 1834                                                ;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.
 1835
 1836  005604  012767  000340  172164  T44:  MOV     #340,PS          ;DISABLE ALL INTERRUPTS
 1837  005612  012767  000100  003424        MOV     #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
 1838  005620  012767  005756  003412        MOV     #6$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
 1839  005626  012767  005652  003406        MOV     #2$,FREEZ1       ;SET UP TO LOOP WITH DATA
 1840  005634  012700  000020                MOV     #20,R0           ;SET UP TO TEST 20(OCTAL)
 1841                                                                 ;LOCATIONS IN BUS ADDRESS MEMORY
 1842  005640  005003                        CLR     R3               ;FIRST LOCATION TO BE
 1843                                                                 ;WRITTEN INTO IS 0
 1844  005642  012701  000020          1$:   MOV     #20,R1           ;SET UP TO CLEAR 20 (OCTAL)
 1845                                                                 ;LOCATIONS IN BUS ADDRESS MEMORY
 1846  005646  005077  003324                CLR     @DHSCR           ;START AT LOCATION 0
 1847  005652  005077  003326          2$:   CLR     @DHBA            ;CLEAR LOCATION IN
 1848                                                                 ;BUS ADDRESS MEMORY
 1849  005656  005277  003314                INC     @DHSCR           ;ADVANCE TO NEXT LOCATION
 1850  005662  005301                        DEC     R1               ;CONTINUE CLEARING
 1851  005664  001372                        BNE     2$               ;IF NOT DONE
 1852  005666  010377  003304                MOV     R3,@DHSCR        ;SELECT ADDRESS TO BE TESTED
 1853  005672  012777  125252  003304        MOV     #125252,@DHBA    ;WRITE 125252 INTO LOCATION
 1854  005700  005077  003272                CLR     @DHSCR           ;ADDRESS LOCATION 0
 1855  005704  012701  000020                MOV     #20,R1           ;SET UP TO CHECK ALL ADDRESSES
 1856                                                                 ;IN BUS ADDRESS MEMORY
 1857  005710  012705  125252          3$:   MOV     #125252,R5       ;125252=EXPECTED RESULT
 1858                                                                 ;IF ADDRESS READ IS LOCATION
 1859                                                                 ;WRITTEN INTO
 1860  005714  017704  003264                MOV     @DHBA,R4         ;READ MEMORY LOCATION
 1861  005720  027703  003252                CMP     @DHSCR,R3        ;IF LINE NUMBER=ADDRESS
 1862                                                                 ;OF LOCATION WRITTEN INTO
 1863                                                                 ;EXPECTED CONTENTS=125252
 1864  005724  001401                        BEQ     4$
 1865  005726  005005                        CLR     R5               ;OTHERWISE, EXPECTED RESULTS=0
 1866  005730  020504          4$:   CMP     R5,R4            ;DOES MEMORY LOCATION CONTAIN
 1867  005732  001401                        BEQ     5$               ;EXPECTED RESULT
 1868  005734  104003                        HLT     3                ;BUS ADDRESS MEMORY ERROR
 1869  005736  104410          5$:   SCOPE1                   ;CHECK FOR LOOP WITH CURRENT DATA
 1870  005740  005277  003232                INC     @DHSCR           ;CHECK CONTENTS OF NEXT LOCATION
 1871  005744  005301                        DEC     R1
 1872  005746  001360                        BNE     3$
 1873  005750  005203                        INC     R3               ;NEXT ADDRESS TO BE WRITTEN
 1874  005752  005300                        DEC     R0
 1875  005754  001332                        BNE     1$
 1876  005756  104400          6$:   SCOPE                    ;CHECK FOR ITERATIONS, LOOP
 1877
 1878                                                ;BUS ADDRESS MEMORY TEST
 1879                                                ;CLEAR ALL LOCATIONS IN BUS ADDRESS MEMORY
 1880                                                ;SET SELECTED LOCATION TO VALUE 52525
 1881                                                ;VERIFY THAT SELECTED LOCATION WAS SET
 1882                                                ;TO 52525.
 1883                                                ;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.
 1884
 1885  005760  012767  000340  172010  T45:  MOV     #340,PS          ;DISABLE ALL INTERRUPTS
 1886  005766  012767  000100  003250        MOV     #100,ICOUNT      ;SET UP FOR 100 ITERATIONS
 1887  005774  012767  006132  003236        MOV     #6$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
```

```
1888  006002  012767  006026  003232          MOV     #2$,FREEZ1      ;SET UP TO LOOP WITH DATA
1889  006010  012700  000020                  MOV     #20,R0          ;SET UP TO TEST 20(OCTAL)
1890                                                                  ;LOCATIONS IN BUS ADDRESS MEMORY
1891  006014  005003                          CLR     R3              ;FIRST LOCATION TO BE
1892                                                                  ;WRITTEN INTO IS 0
1893  006016  012701  000020          1$:     MOV     #20,R1          ;SET UP TO CLEAR 20 (OCTAL)
1894                                                                  ;LOCATIONS IN BUS ADDRESS MEMORY
1895  006022  005077  003150                  CLR     @DHSCR          ;START AT LOCATION 0
1896  006026  005077  003152          2$:     CLR     @DHBA           ;CLEAR LOCATION IN
1897                                                                  ;BUS ADDRESS MEMORY
1898  006032  005277  003140                  INC     @DHSCR          ;ADVANCE TO NEXT LOCATION
1899  006036  005301                          DEC     R1              ;CONTINUE CLEARING
1900  006040  001372                          BNE     2$              ;IF NOT DONE
1901  006042  010377  003130                  MOV     R3,@DHSCR       ;SELECT ADDRESS TO BE TESTED
1902  006046  012777  052525  003130          MOV     #52525,@DHBA    ;WRITE 52525 INTO LOCATION
1903  006054  005077  003116                  CLR     @DHSCR          ;ADDRESS LOCATION 0
1904  006060  012701  000020                  MOV     #20,R1          ;SET UP TO CHECK ALL ADDRESSES
1905                                                                  ;IN BUS ADDRESS MEMORY
1906  006064  012705  052525          3$:     MOV     #52525,R5       ;52525=EXPECTED RESULT
1907                                                                  ;IF ADDRESS READ IS LOCATION
1908                                                                  ;WRITTEN INTO
1909  006070  017704  003110                  MOV     @DHBA,R4        ;READ MEMORY LOCATION
1910  006074  027703  003076                  CMP     @DHSCR,R3       ;IF LINE NUMBER=ADDRESS
1911                                                                  ;OF LOCATION WRITTEN INTO
1912                                                                  ;EXPECTED CONTENTS=52525
1913  006100  001401                          BEQ     4$
1914  006102  005005                          CLR     R5              ;OTHERWISE, EXPECTED RESULTS=0
1915  006104  020504          4$:     CMP     R5,R4           ;DOES MEMORY LOCATION CONTAIN
1916  006106  001401                          BEQ     5$              ;EXPECTED RESULT
1917  006110  104003                          HLT     3               ;BUS ADDRESS MEMORY ERROR
1918  006112  104410          5$:     SCOPE1                  ;CHECK FOR LOOP WITH CURRENT DATA
1919  006114  005277  003056                  INC     @DHSCR          ;CHECK CONTENTS OF NEXT LOCATION
1920  006120  005301                          DEC     R1
1921  006122  001360                          BNE     3$
1922  006124  005203                          INC     R3              ;NEXT ADDRESS TO BE WRITTEN
1923  006126  005300                          DEC     R0
1924  006130  001332                          BNE     1$
1925  006132  104400          6$:     SCOPE                   ;CHECK FOR ITERATIONS, LOOP
1926
1927                                  ;BYTE COUNT MEMORY TEST
1928                                  ;CLEAR ALL LOCATIONS IN BYTE COUNT MEMORY
1929                                  ;SET SELECTED LOCATION TO VALUE 177777
1930                                  ;VERIFY THAT SELECTED LOCATION WAS SET
1931                                  ;TO 177777.
1932                                  ;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.
1933
1934  006134  012767  000340  171634  T46:    MOV     #340,PS         ;DISABLE ALL INTERRUPTS
1935  006142  012767  000100  003074          MOV     #100,ICOUNT     ;SET UP FOR 100 ITERATIONS
1936  006150  012767  006306  003062          MOV     #6$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
1937  006156  012767  006202  003056          MOV     #2$,FREEZ1      ;SET UP TO LOOP WITH DATA
1938  006164  012700  000020                  MOV     #20,R0          ;SET UP TO TEST 20(OCTAL)
1939                                                                  ;LOCATIONS IN BYTE COUNT MEMORY
1940  006170  005003                          CLR     R3              ;FIRST LOCATION TO BE
1941                                                                  ;WRITTEN INTO IS 0
1942  006172  012701  000020          1$:     MOV     #20,R1          ;SET UP TO CLEAR 20 (OCTAL)
1943                                                                  ;LOCATIONS IN BYTE COUNT MEMORY
```

```
1944  006176  005077  002774                 CLR     @DHSCR          ;START AT LOCATION 0
1945  006202  005077  003000          2$:    CLR     @DHBC           ;CLEAR LOCATION IN
1946                                                                  ;BYTE COUNT MEMORY
1947  006206  005277  002764                 INC     @DHSCR          ;ADVANCE TO NEXT LOCATION
1948  006212  005301                         DEC     R1              ;CONTINUE CLEARING
1949  006214  001372                         BNE     2$              ;IF NOT DONE
1950  006216  010377  002754                 MOV     R3,@DHSCR       ;SELECT ADDRESS TO BE TESTED
1951  006222  012777  177777  002756         MOV     #177777,@DHBC   ;WRITE 177777 INTO LOCATION
1952  006230  005077  002742                 CLR     @DHSCR          ;ADDRESS LOCATION 0
1953  006234  012701  000020                 MOV     #20,R1          ;SET UP TO CHECK ALL ADDRESSES
1954                                                                  ;IN BYTE COUNT MEMORY
1955  006240  012705  177777          3$:    MOV     #177777,R5      ;177777=EXPECTED RESULT
1956                                                                  ;IF ADDRESS READ IS LOCATION
1957                                                                  ;WRITTEN INTO
1958  006244  017704  002736                 MOV     @DHBC,R4        ;READ MEMORY LOCATION
1959  006250  027703  002722                 CMP     @DHSCR,R3       ;IF LINE NUMBER=ADDRESS
1960                                                                  ;OF LOCATION WRITTEN INTO
1961                                                                  ;EXPECTED CONTENTS=177777
1962  006254  001401                         BEQ     4$
1963  006256  005005                         CLR     R5              ;OTHERWISE, EXPECTED RESULTS=0
1964  006260  020504                  4$:    CMP     R5,R4           ;DOES MEMORY LOCATION CONTAIN
1965  006262  001401                         BEQ     5$              ;EXPECTED RESULT
1966  006264  104004                         HLT     4               ;BYTE COUNT MEMORY ERROR
1967  006266  104410                  5$:    SCOPE1                  ;CHECK FOR LOOP WITH CURRENT DATA
1968  006270  005277  002702                 INC     @DHSCR          ;CHECK CONTENTS OF NEXT LOCATION
1969  006274  005301                         DEC     R1
1970  006276  001360                         BNE     3$
1971  006300  005203                         INC     R3              ;NEXT ADDRESS TO BE WRITTEN
1972  006302  005300                         DEC     R0
1973  006304  001332                         BNE     1$
1974  006306  104400                  6$:    SCOPE                   ;CHECK FOR ITERATIONS, LOOP
1975
1976                                        ;BYTE COUNT MEMORY TEST
1977                                        ;CLEAR ALL LOCATIONS IN BYTE COUNT MEMORY
1978                                        ;SET SELECTED LOCATION TO VALUE 125252
1979                                        ;VERIFY THAT SELECTED LOCATION WAS SET
1980                                        ;TO 125252.
1981                                        ;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.
1982
1983  006310  012767  000340  171460  T47:  MOV     #340,PS         ;DISABLE ALL INTERRUPTS
1984  006316  012767  000100  002720         MOV     #100,ICOUNT     ;SET UP FOR 100 ITERATIONS
1985  006324  012757  006462  002706         MOV     #6$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
1986  006332  012767  006356  002702         MOV     #2$,FREEZ1      ;SET UP TO LOOP WITH DATA
1987  006340  012700  000020                 MOV     #20,R0          ;SET UP TO TEST 20(OCTAL)
1988                                                                  ;LOCATIONS IN BYTE COUNT MEMORY
1989  006344  005003                         CLR     R3              ;FIRST LOCATION TO BE
1990                                                                  ;WRITTEN INTO IS 0
1991  006346  012701  000020          1$:    MOV     #20,R1          ;SET UP TO CLEAR 20 (OCTAL)
1992                                                                  ;LOCATIONS IN BYTE COUNT MEMORY
1993  006352  005077  002620                 CLR     @DHSCR          ;START AT LOCATION 0
1994  006356  005077  002624          2$:    CLR     @DHBC           ;CLEAR LOCATION IN
1995                                                                  ;BYTE COUNT MEMORY
1996  006362  005277  002610                 INC     @DHSCR          ;ADVANCE TO NEXT LOCATION
1997  006366  005301                         DEC     R1              ;CONTINUE CLEARING
1998  006370  001372                         BNE     2$              ;IF NOT DONE
1999  006372  010377  002600                 MOV     R3,@DHSCR       ;SELECT ADDRESS TO BE TESTED
```

```
2000  006376  012777  125252  002602          MOV     #125252,@DHBC           ;WRITE 125252 INTO LOCATION
2001  006404  005077  002566                  CLR     @DHSCR                  ;ADDRESS LOCATION 0
2002  006410  012701  000020                  MOV     #20,R1                  ;SET UP TO CHECK ALL ADDRESSES
2003                                                                          ;IN BYTE COUNT MEMORY
2004  006414  012705  125252          3$:     MOV     #125252,R5              ;125252=EXPECTED RESULT
2005                                                                          ;IF ADDRESS READ IS LOCATION
2006                                                                          ;WRITTEN INTO
2007  006420  017704  002562                  MOV     @DHBC,R4                ;READ MEMORY LOCATION
2008  006424  027703  002546                  CMP     @DHSCR,R3               ;IF LINE NUMBER=ADDRESS
2009                                                                          ;OF LOCATION WRITTEN INTO
2010                                                                          ;EXPECTED CONTENTS=125252
2011  006430  001401                          BEQ     4$
2012  006432  005005                          CLR     R5                      ;OTHERWISE, EXPECTED RESULTS=0
2013  006434  020504          4$:     CMP     R5,R4                   ;DOES MEMORY LOCATION CONTAIN
2014  006436  001401                          BEQ     5$                      ;EXPECTED RESULT
2015  006440  104004                          HLT     4                       ;BYTE COUNT MEMORY ERROR
2016  006442  104410          5$:     SCOPE1                          ;CHECK FOR LOOP WITH CURRENT DATA
2017  006444  005277  002526                  INC     @DHSCR                  ;CHECK CONTENTS OF NEXT LOCATION
2018  006450  005301                          DEC     R1
2019  006452  001360                          BNE     3$
2020  006454  005203                          INC     R3                      ;NEXT ADDRESS TO BE WRITTEN
2021  006456  005300                          DEC     R0
2022  006460  001332                          BNE     1$
2023  006462  104400          6$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
2024
2025                                          ;BYTE COUNT MEMORY TEST
                                              ;CLEAR ALL LOCATIONS IN BYTE COUNT MEMORY
                                              ;SET SELECTED LOCATION TO VALUE 52525
                                              ;VERIFY THAT SELECTED LOCATION WAS SET
                                              ;TO 52525.
                                              ;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.
2031
2032  006464  012767  000340  171304  TST0:   MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
2033  006472  012767  000100  002544          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
2034  006500  012767  006636  002532          MOV     #6$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
2035  006506  012767  006532  002526          MOV     #2$,FREEZ1              ;SET UP TO LOOP WITH DATA
2036  006514  012700  000020                  MOV     #20,R0                  ;SET UP TO TEST 20(OCTAL)
2037                                                                          ;LOCATIONS IN BYTE COUNT MEMORY
2038  006520  005003                          CLR     R3                      ;FIRST LOCATION TO BE
2039                                                                          ;WRITTEN INTO IS 0
2040  006522  012701  000020          1$:     MOV     #20,R1                  ;SET UP TO CLEAR 20 (OCTAL)
2041                                                                          ;LOCATIONS IN BYTE COUNT MEMORY
2042  006526  005077  002444                  CLR     @DHSCR                  ;START AT LOCATION 0
2043  006532  005077  002450          2$:     CLR     @DHBC                   ;CLEAR LOCATION IN
2044                                                                          ;BYTE COUNT MEMORY
2045  006536  005277  002434                  INC     @DHSCR                  ;ADVANCE TO NEXT LOCATION
2046  006542  005301                          DEC     R1                      ;CONTINUE CLEARING
2047  006544  001372                          BNE     2$                      ;IF NOT DONE
2048  006546  010377  002424                  MOV     R3,@DHSCR               ;SELECT ADDRESS TO BE TESTED
2049  006552  012777  052525  002426          MOV     #52525,@DHBC            ;WRITE 52525 INTO LOCATION
2050  006560  005077  002412                  CLR     @DHSCR                  ;ADDRESS LOCATION 0
2051  006564  012701  000020                  MOV     #20,R1                  ;SET UP TO CHECK ALL ADDRESSES
2052                                                                          ;IN BYTE COUNT MEMORY
2053  006570  012705  052525          3$:     MOV     #52525,R5               ;52525=EXPECTED RESULT
2054                                                                          ;IF ADDRESS READ IS LOCATION
2055                                                                          ;WRITTEN INTO
```

```
2056  006574  017704  002406                MOV     @DHBC,R4                ;READ MEMORY LOCATION
2057  006600  027703  002372                CMP     @DHSCR,R3               ;IF LINE NUMBER=ADDRESS
2058                                                                          ;OF LOCATION WRITTEN INTO
2059                                                                          ;EXPECTED CONTENTS=52525
2060  006604  001401                        BEQ     4$
2061  006606  005005                        CLR     R5                      ;OTHERWISE, EXPECTED RESULTS=0
2062  006610  020504              4$:       CMP     R5,R4                   ;DOES MEMORY LOCATION CONTAIN
2063  006612  001401                        BEQ     5$                      ;EXPECTED RESULT
2064  006614  104004                        HLT     4                       ;BYTE COUNT MEMORY ERROR
2065  006616  104410              5$:       SCOPE1                          ;CHECK FOR LOOP WITH CURRENT DATA
2066  006620  005277  002352                INC     @DHSCR                  ;CHECK CONTENTS OF NEXT LOCATION
2067  006624  005301                        DEC     R1
2068  006626  001360                        BNE     3$
2069  006630  005203                        INC     R3                      ;NEXT ADDRESS TO BE WRITTEN
2070  006632  005300                        DEC     R0
2071  006634  001332                        BNE     1$
2072  006636  104400              6$:       SCOPE                           ;CHECK FOR ITERATIONS, LOOP
2073
2074                                        ;BUS ADDRESS MEMORY TEST
2075                                        ;SET ALL LOCATIONS IN BUS ADDRESS MEMORY TO 177777
2076                                        ;SET SELECTED LOCATION TO VALUE 0
2077                                        ;VERIFY THAT SELECTED LOCATION WAS SET
2078                                        ;TO 0.
2079                                        ;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.
2080
2081  006640  012767  000340  171130  T51:   MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
2082  006646  012767  000100  002370        MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
2083  006654  012767  007014  002356        MOV     #6$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
2084  006662  012767  006706  002352        MOV     #2$,FREEZ1              ;SET UP TO LOOP WITH DATA
2085  006670  012700  000020                MOV     #20,R0                  ;SET UP TO TEST 20(OCTAL)
2086                                                                          ;LOCATIONS IN BUS ADDRESS MEMORY
2087  006674  005003                        CLR     R3                      ;FIRST LOCATION TO BE
2088                                                                          ;WRITTEN INTO IS 0
2089  006676  012701  000020          1$:   MOV     #20,R1                  ;SET UP TO SET 20 (OCTAL)
2090                                                                          ;LOCATIONS IN BUS ADDRESS MEMORY TO 177777
2091  006702  005077  002270                CLR     @DHSCR                  ;START AT LOCATION 0
2092  006706  012777  177777  002270  2$:   MOV     #177777,@DHBA           ;SET LOCATION IN
2093                                                                          ;BUS ADDRESS MEMORY
2094  006714  005277  002256                INC     @DHSCR                  ;ADVANCE TO NEXT LOCATION
2095  006720  005301                        DEC     R1                      ;CONTINUE SETTING
2096  006722  001371                        BNE     2$                      ;IF NOT DONE
2097  006724  010377  002246                MOV     R3,@DHSCR               ;SELECT ADDRESS TO BE TESTED
2098  006730  012777  000000  002246        MOV     #0,@DHBA                ;WRITE 0 INTO LOCATION
2099  006736  005077  002234                CLR     @DHSCR                  ;ADDRESS LOCATION 0
2100  006742  012701  000020                MOV     #20,R1                  ;SET UP TO CHECK ALL ADDRESSES
2101                                                                          ;IN BUS ADDRESS MEMORY
2102  006746  012705  000000          3$:   MOV     #0,R5           ;0=EXPECTED RESULT
2103                                                                          ;IF ADDRESS READ IS LOCATION
2104                                                                          ;WRITTEN INTO
2105  006752  017704  002226                MOV     @DHBA,R4                ;READ MEMORY LOCATION
2106  006756  027703  002214                CMP     @DHSCR,R3               ;IF LINE NUMBER=ADDRESS
2107                                                                          ;OF LOCATION WRITTEN INTO
2108                                                                          ;EXPECTED CONTENTS=0
2109  006762  001401                        BEQ     4$
2110  006764  005105                        COM     R5                      ;OTHERWISE, EXPECTED RESULTS=177777
2111  006766  020504              4$:       CMP     R5,R4                   ;DOES MEMORY LOCATION CONTAIN
```

```
2112  006770  001401                        BEQ     5$                      ;EXPECTED RESULT
2113  006772  104003                        HLT     3                       ;BUS ADDRESS MEMORY ERROR
2114  006774  104410                5$:     SCOPE1                          ;CHECK FOR LOOP WITH CURRENT DATA
2115  006776  005277  002174                INC     @DHSCR                  ;CHECK CONTENTS OF NEXT LOCATION
2116  007002  005301                        DEC     R1
2117  007004  001360                        BNE     3$
2118  007006  005203                        INC     R3                      ;NEXT ADDRESS TO BE WRITTEN
2119  007010  005300                        DEC     R0
2120  007012  001331                        BNE     1$
2121  007014  104400                6$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
2122
2123                                        ;BYTE COUNT MEMORY TEST
2124                                        ;SET ALL LOCATIONS IN BYTE COUNT MEMORY TO 177777
2125                                        ;SET SELECTED LOCATION TO VALUE 0
2126                                        ;VERIFY THAT SELECTED LOCATION WAS SET
2127                                        ;TO 0.
2128                                        ;VERIFY THAT NO OTHER LOCATION WAS MODIFIED.
2130  007016  012767  000340  170752  T52:   MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
2131  007024  012767  000100  002212        MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
2132  007032  012767  007172  002200        MOV     #6$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
2133  007040  012767  007064  002174        MOV     #2$,FREEZ1              ;SET UP TO LOOP WITH DATA
2134  007046  012700  000020                MOV     #20,R0                  ;SET UP TO TEST 20(OCTAL)
2135                                                                        ;LOCATIONS IN BYTE COUNT MEMORY
2136  007052  005003                        CLR     R3                      ;FIRST LOCATION TO BE
2137                                                                        ;WRITTEN INTO IS 0
2138  007054  012701  000020          1$:    MOV     #20,R1                  ;SET UP TO SET 20 (OCTAL)
2139                                                                        ;LOCATIONS IN BYTE COUNT MEMORY TO 177777
2140  007060  005077  002112                CLR     @DHSCR                  ;START AT LOCATION 0
2141  007064  012777  177777  002114  2$:    MOV     #177777,@DHBC           ;SET LOCATION IN
2142                                                                        ;BYTE COUNT MEMORY
2143  007072  005277  002100                INC     @DHSCR                  ;ADVANCE TO NEXT LOCATION
2144  007076  005301                        DEC     R1                      ;CONTINUE SETTING
2145  007100  001371                        BNE     2$                      ;IF NOT DONE
2146  007102  010377  002070                MOV     R3,@DHSCR               ;SELECT ADDRESS TO BE TESTED
2147  007106  012777  000000  002072        MOV     #0,@DHBC                ;WRITE 0 INTO LOCATION
2148  007114  005077  002056                CLR     @DHSCR                  ;ADDRESS LOCATION 0
2149  007120  012701  000020                MOV     #20,R1                  ;SET UP TO CHECK ALL ADDRESSES
2150                                                                        ;IN BYTE COUNT MEMORY
2151  007124  012705  000000          3$:    MOV     #0,R5           ;0=EXPECTED RESULT
2152                                                                        ;IF ADDRESS READ IS LOCATION
2153                                                                        ;WRITTEN INTO
2154  007130  017704  002052                MOV     @DHBC,R4                ;READ MEMORY LOCATION
2155  007134  027703  002036                CMP     @DHSCR,R3               ;IF LINE NUMBER=ADDRESS
2156                                                                        ;OF LOCATION WRITTEN INTO
2157                                                                        ;EXPECTED CONTENTS=0
2158  007140  001401                        BEQ     4$
2159  007142  005105                        COM     R5
2160  007144  020504                  4$:   CMP     R5,R4                   ;DOES MEMORY LOCATION CONTAIN
2161  007146  001401                        BEQ     5$                      ;EXPECTED RESULT
2162  007150  104004                        HLT     4                       ;BYTE COUNT MEMORY ERROR
2163  007152  104410                5$:     SCOPE1                          ;CHECK FOR LOOP WITH CURRENT DATA
2164  007154  005277  002016                INC     @DHSCR                  ;CHECK CONTENTS OF NEXT LOCATION
2165  007160  005301                        DEC     R1
2166  007162  001360                        BNE     3$
2167  007164  005203                        INC     R3                      ;NEXT ADDRESS TO BE WRITTEN
```

```
2168  007166  005300                        DEC     R0
2169  007170  001331                        BNE     1$
2170  007172  104400              6$:       SCOPE                           ;CHECK FOR ITERATIONS, LOOP
2171
2172                                        ;MEMORY EXTENSION MEMORY TEST
2173                                        ;VERIFY THAT LOW AND HIGH ORDER MEMORY EXTENSION BIT CAN BE
2174                                        ;SET AND CLEARED IN SELECTED MEMORY EXTENSION MEMORY LOCATION
2175  007174  012767  000340  170574  TS3:  MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
2176  007202  012767  000100  001534        MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
2177  007210  012767  007364  001522        MOV     #6$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
2178  007216  012767  007242  002016        MOV     #2$,FREEZI              ;SET UP TO LOOP WITH DATA
2179  007224  012700  000020                MOV     #20,R0                  ;SET UP TO TEST 20(OCTAL)
2180                                                                        ;LOCATIONS IN MEMORY EXTENSION MEMORY
2181  007230  005003                        CLR     R3                      ;FIRST LOCATION TO BE
2182                                                                        ;WRITEEN INTO IS 0
2183  007232  012701  000020          1$:   MOV     #20,R1                  ;SET UP TO CLEAR 20 (OCTAL)
2184                                                                        ;LOCATIONS IN MEMORY EXTENSION MEMORY
2185  007236  005077  001734                CLR     @DHSCR                  ;START AT LOCATION 0
2186  007242  042777  000060  001726  2$:   BIC     #60,@DHSCR              ;CLEAR LOCATION IN
2187  007250  012777  000000  001726        MOV     #0,@DHBA                ;MEMORY EXTENSION MEMORY
2188  007256  005277  001714                INC     @DHSCR                  ;ADVANCE TO NEXT LOCATION
2189  007262  005301                        DEC     R1                      ;CONTINUE CLEARING
2190  007264  001366                        BNE     2$                      ;IF NOT DONE
2191  007266  010377  001704                MOV     R3,@DHSCR               ;SELECT ADDRESS TO BE TESTED
2192  007272  052777  000060  001676        BIS     #60,@DHSCR              ;WRITE LOW AND HIGH INTO LOCATION
2193  007300  012777  000000  001676        MOV     #0,@DHBA                ;LOAD ADDRESS
2194  007306  005077  001664                CLR     @DHSCR                  ;ADDRESS LOCATION 0
2195  007312  012701  000020                MOV     #20,R1                  ;SET UP TO CHECK ALL ADDRESSES
2196                                                                        ;IN MEMORY EXTENSION MEMORY
2197  007316  012705  000300          3$:   MOV     #300,R5         ;LOW AND HIGH=EXPECTED RESULT
2198                                                                        ;IF ADDRESS READ IS LOCATION
2199                                                                        ;WRITTEN INTO
2200  007322  017704  001666                MOV     @DHSSR,R4               ;READ MEMORY LOCATION
2201  007326  027703  001644                CMP     @DHSCR,R3               ;IF LINE NUMBER=ADDRESS
2202                                                                        ;OF LOCATION WRITTEN INTO
2203                                                                        ;EXPECTED CONTENTS=LOW AND HIGH
2204  007332  001401                        BEQ     4$
2205  007334  005005                        CLR     R5                      ;OTHERWISE, EXPECTED RESULTS=0
2206  007336  020504          4$:   CMP     R5,R4                   ;DOES MEMORY LOCATION CONTAIN
2207  007340  001401                        BEQ     5$                      ;EXPECTED RESULT
2208  007342  104005                        HLT     5                       ;MEMORY EXTENSION DATA ERROR
2209  007344  104410          5$:   SCOPE1                          ;CHECK FOR LOOP WITH CURRENT DATA
2210  007346  005277  001624                INC     @DHSCR                  ;CHECK CONTENTS OF NEXT LOCATION
2211  007352  005301                        DEC     R1
2212  007354  001340                        BNE     3$
2213  007356  005203                        INC     R3                      ;NEXT ADDRESS TO BE WRITTEN
2214  007360  005300                        DEC     R0
2215  007362  001323                        BNE     1$
2216  007364  104400              6$:       SCOPE                           ;CHECK FOR ITERATIONS, LOOP
2217
2218                                        ;MEMORY EXTENSION MEMORY TEST
2219                                        ;VERIFY THAT LOW ORDER MEMORY EXTENSION BIT CAN BE
2220                                        ;SET AND CLEARED IN SELECTED MEMORY EXTENSION MEMORY LOCATION
2221  007366  012767  000340  170402  TS4:  MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
2222  007374  012767  000100  001642        MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
2223  007402  012767  007556  001630        MOV     #6$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
```

```
2224  007410  012767  007434  001624          MOV     #2$,FREEZ1              ;SET UP TO LOOP WITH DATA
2225  007416  012700  000020                  MOV     #20,R0                  ;SET UP TO TEST 20(OCTAL)
2226                                                                          ;LOCATIONS IN MEMORY EXTENSION MEMORY
2227  007422  005003                          CLR     R3                      ;FIRST LOCATION TO BE
                                                                              ;WRITEEN INTO IS 0
2230  007424  012701  000020          1$:     MOV     #20,R1                  ;SET UP TO CLEAR 20 (OCTAL)
2231                                                                          ;LOCATIONS IN MEMORY EXTENSION MEMORY
2232  007430  005077  001542                  CLR     @DHSCR                  ;START AT LOCATION 0
2233  007434  042777  000060  001534  2$:     BIC     #60,@DHSCR              ;CLEAR LOCATION IN
2234  007442  012777  000000  001534          MOV     #0,@DHBA                ;MEMORY EXTENSION MEMORY
2235  007450  005277  001522                  INC     @DHSCR                  ;ADVANCE TO NEXT LOCATION
2236  007454  005301                          DEC     R1                      ;CONTINUE CLEARING
2237  007456  001360                          BNE     2$                      ;IF NOT DONE
2240  007460  010377  001512                  MOV     R3,@DHSCR               ;SELECT ADDRESS TO BE TESTED
2241  007464  052777  000020  001504          BIS     #20,@DHSCR              ;WRITE LOW INTO LOCATION
2242  007472  012777  000000  001504          MOV     #0,@DHBA                ;LOAD ADDRESS
2243  007500  005077  001472                  CLR     @DHSCR                  ;ADDRESS LOCATION 0
2244  007504  012701  000020                  MOV     #20,R1                  ;SET UP TO CHECK ALL ADDRESSES
2245                                                                          ;IN MEMORY EXTENSION MEMORY
2246  007510  012705  000100          3$:     MOV     #100,R5         ;LOW=EXPECTED RESULT
2247                                                                          ;IF ADDRESS READ IS LOCATION
2250                                                                          ;WRITTEN INTO
2251  007514  017704  001474                  MOV     @DHSSR,R4               ;READ MEMORY LOCATION
2252  007520  027703  001452                  CMP     @DHSCR,R3               ;IF LINE NUMBER=ADDRESS
2253                                                                          ;OF LOCATION WRITTEN INTO
2254                                                                          ;EXPECTED CONTENTS=LOW
2255  007524  001401                          BEQ     4$
2256  007526  005005                          CLR     R5                      ;OTHERWISE, EXPECTED RESULTS=0
2257  007530  020504                  4$:     CMP     R5,R4                   ;DOES MEMORY LOCATION CONTAIN
2260  007532  001401                          BEQ     5$                      ;EXPECTED RESULT
2261  007534  104005                          HLT     5                       ;MEMORY EXTENSION DATA ERROR
2262  007536  104410                  5$:     SCOPE1                          ;CHECK FOR LOOP WITH CURRENT DATA
2263  007542  005277  001432                  INC     @DHSCR                  ;CHECK CONTENTS OF NEXT LOCATION
2264  007546  005301                          DEC     R1
2265  007550  001360                          BNE     3$
2266  007552  005203                          INC     R3                      ;NEXT ADDRESS TO BE WRITTEN
2267  007554  005300                          DEC     R0
2270  007556  001323                          BNE     1$
2271  007560  104400                  6$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
2272
2273                                          ;MEMORY EXTENSION MEMORY TEST
2274                                          ;VERIFY THAT HIGH ORDER MEMORY EXTENSION BIT CAN BE
2275                                          ;SET AND CLEARED IN SELECTED MEMORY EXTENSION MEMORY LOCATION
2276  007562  012767  000340  170210  T55:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
2277  007570  012767  000100  001450          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
2300  007576  012767  007750  001436          MOV     #ES,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
2301  007604  012767  007626  001432          MOV     #2$,FREEZ1              ;SET UP TO LOOP WITH DATA
2302  007612  012700  000020                  MOV     #20,R0                  ;SET UP TO TEST 20(OCTAL)
2303                                                                          ;LOCATIONS IN MEMORY EXTENSION MEMORY
2304  007616  005003                          CLR     R3                      ;FIRST LOCATION TO BE
2305                                                                          ;WRITEEN INTO IS 0
2306  007620  012701  000020          1$:     MOV     #20,R1                  ;SET UP TO CLEAR 20 (OCTAL)
2307                                                                          ;LOCATIONS IN MEMORY EXTENSION MEMORY
2310  007624  005077  001350                  CLR     @DHSCR                  ;START AT LOCATION 0
2311  007630  042777  000060  001342  2$:     BIC     #60,@DHSCR              ;CLEAR LOCATION IN
2312  007636  012777  000000  001342          MOV     #0,@DHBA                ;MEMORY EXTENSION MEMORY
```

```
 2280  007642  005277  001330            INC     @DHSCR                  ;ADVANCE TO NEXT LOCATION
 2281  007646  005301                    DEC     R1                      ;CONTINUE CLEARING
 2282  007650  001366                    BNE     2$                      ;IF NOT DONE
 2283  007652  010377  001320            MOV     R3,@DHSCR               ;SELECT ADDRESS TO BE TESTED
 2284  007656  052777  000040  001312    BIS     #40,@DHSCR              ;WRITE HIGH INTO LOCATION
 2285  007664  012777  000000  001312    MOV     #0,@DHBA                ;LOAD ADDRESS
 2286  007672  005077  001300            CLR     @DHSCR                  ;ADDRESS LOCATION 0
 2287  007676  012701  000020            MOV     #20,R1                  ;SET UP TO CHECK ALL ADDRESSES
 2288                                                                    ;IN MEMORY EXTENSION MEMORY
 2289  007702  012705  000200    3$:     MOV     #200,R5         ;HIGH=EXPECTED RESULT
 2290                                                                    ;IF ADDRESS READ IS LOCATION
 2291                                                                    ;WRITTEN INTO
 2292  007706  017704  001302            MOV     @DHSSR,R4               ;READ MEMORY LOCATION
 2293  007712  027703  001260            CMP     @DHSCR,R3               ;IF LINE NUMBER=ADDRESS
 2294                                                                    ;OF LOCATION WRITTEN INTO
 2295                                                                    ;EXPECTED CONTENTS=HIGH
 2296  007716  001401                    BEQ     4$
 2297  007720  005005                    CLR     R5                      ;OTHERWISE, EXPECTED RESULTS=0
 2298  007722  020504            4$:     CMP     R5,R4                   ;DOES MEMORY LOCATION CONTAIN
 2299  007724  001401                    BEQ     5$                      ;EXPECTED RESULT
 2300  007726  104005                    HLT     5                       ;MEMORY EXTENSION DATA ERROR
 2301  007730  104410            5$:     SCOPE1                          ;CHECK FOR LOOP WITH CURRENT DATA
 2302  007732  005277  001240            INC     @DHSCR                  ;CHECK CONTENTS OF NEXT LOCATION
 2303  007736  005301                    DEC     R1
 2304  007740  001360                    BNE     3$
 2305  007742  005203                    INC     R3                      ;NEXT ADDRESS TO BE WRITTEN
 2306  007744  005300                    DEC     R0
 2307  007746  001323                    BNE     1$
 2308  007750  104400            6$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
```

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

007752  104401                  EOP:    TYPE                            ;TYPE NAME OF TEST
007754  011643                          MEPASS
007756  005067  001312                  CLR     LAST                    ;CLEAR LAST ERROR PC
007762  005067  001242                  CLR     ERRFLG                  ;CLEAR ERROR FLAG
007766  005267  001240                  INC     PASCNT                  ;UPDATE PASS COUNT
007772  016767  001234  167570          MOV     PASCNT,LIGHTS           ;DISPLAY PASS COUNT
010000  013701  000042                  MOV     @#42,R1                 ;CHECK FOR ACT-11 OR ODP
010004  001405                          BEQ     RESTRT                  ;IF NOT, CONTINUE TESTING
010006  000005                          RESET
010010  004711                  LOGICAL:        JSR     PC,(R1)
010012  000240                          NOP
010014  000240                          NOP
010016  000240                          NOP
010020  000167  171156          RESTRT: JMP     BEGIN

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

010024  032767  002000  167536  SCOPER: BIT     #SW10,SWR
010032  001030                          BNE     4$
010034  032767  040000  167526  1$:     BIT     #SW14,SWR
010042  001021                          BNE     3$
010044  032767  004000  167516          BIT     #SW11,SWR
010052  001006                          BNE     2$
010054  005267  001166                  INC     LPCNT
010060  026767  001162  001156          CMP     LPCNT,ICOUNT
010066  001007                          BNE     3$
010070  005067  001152          2$:     CLR     LPCNT
010074  005067  001130                  CLR     ERRFLG
010100  011667  001132                  MOV     (SP),RETURN
010104  000002                          RTI
010106  016716  001124          3$:     MOV     RETURN,(SP)
010112  000002                          RTI
010114  005767  001110          4$:     TST     ERRFLG
010120  001745                          BEQ     1$
010122  000762                          BR      2$

                                        ;CHECK FOR FREEZE ON CURRENT DATA

010124  032767  001000  167436  SCOPIR: BIT     #SW09,SWR
010132  001402                          BEQ     1$
010134  016716  001102                  MOV     FREEZ1,(SP)
010140  000002                  1$:     RTI
```

J04


```


                                                     ;ERROR HANDLER

010142  032767  020000  167420  ERRORS: BIT     #SW13,SWR
010150  001051                          BNE     HALTS
010152  021667  001116                  CMP     (SP),LAST
010156  001404                          BEQ     1$
010160  011667  001110                  MOV     (SP),LAST
010164  005067  001040                  CLR     ERRFLG
010170  104406                  1$:     SAVOSP
010172  011605                          MOV     (SP),R5
010174  162705  000002                  SUB     #2,R5
010200  011504                          MOV     (R5),R4
010202  006304                          ASL     R4
010204  006304                          ASL     R4
010206  042704  177001                  BIC     #177001,R4
010212  062704  011752                  ADD     #ERRTAB,R4
010216  012467  000034                  MOV     (R4)+,ERRMSG
010222  011467  000042                  MOV     (R4),DATABP
010226  005767  000776                  TST     ERRFLG
010232  001403                          BEQ     TYPMSG
010234  005767  000030                  TST     DATABP
010240  001007                          BNE     TYPDAT
010242  104402                  TYPMSG: OCTASC
010244  010336                          ERTABO
010246  012767  000001  000754          MOV     #1,ERRFLG
010254  104401                          TYPE
010256  000000                  ERRMSG: 0
010260  005767  000004          TYPDAT: TST     DATABP
010264  001402                          BEQ     RESREG
010266  104402                          OCTASC
010270  000000                  DATABP: 0
010272  104407                  RESREG: RESOS
010274  005767  167270          HALTS:  TST     SWR
010300  100005                          BPL     EXITER
010302  010046                          PUSHRO
010304  016600  000002                  MOV     2(SP),RO
010310  000000                          HALT
010312  012600                          POPRO
010314  005267  000714          EXITER: INC     ERRCNT
010320  032767  002000  167242          BIT     #SW10,SWR
010326  001402                          BEQ     1$
010330  016716  000704                  MOV     ESCAPE,(SP)
010334  000002                  1$:     RTI
010336  000001                  ERTABO: 1
010340     006     002                  .BYTE   6,2
010342  011266                          SAVPC
```

```
2407                                                    ;TRAP DISPATCH SERVICE
2408                                                    ;ARGUMENT OF TRAP IS EXTRACTED
2409                                                    ;AND USED AS OFFSET TO OBTAIN POINTER
2410                                                    ;TO SELECTED SUBROUTINE
2411
2412  010344  011646                        TRPSRV:  MOV     (SP),-(SP)          ;GET PC OF RETURN
2413  010346  162716  000002                        SUB     #2,(SP)             ;=PC OF TRAP
2414  010352  017616  000000                        MOV     @(SP),(SP)          ;GET TRP
2415  010356  006316                        TRPOK:   ASL     (SP)                ;MULTIPLY TRAP ARG BY 2
2416  010360  042716  177001                        BIC     #177001,(SP)        ;CLEAR UNWANTED BITS
2417  010364  062716  011672                        ADD     #TRPTAB,(SP)        ;POINTER TO SUBROUTINE ADDRESS
2418  010370  017616  000000                        MOV     @(SP),(SP)          ;SUBROUTINE ADDRESS
2419  010374  000136                                JMP     @(SP)+              ;GO TO SUBROUTINE
2420
2421                                                    ;SAVE PC OF TEST THAT FAILED AND R0-R5
2422
2423  010376  016667  000004  000662  SVOSP:   MOV     4(SP),SAVPC
2424
2425                                                    ;SAVE R0-R5
2426
2427  010404  010567  000652          SV05:    MOV     R5,SAVR5
2428  010410  010467  000644                   MOV     R4,SAVR4
2429  010414  010367  000636                   MOV     R3,SAVR3
2430  010420  010267  000630                   MOV     R2,SAVR2
2431  010424  010167  000622                   MOV     R1,SAVR1
2432  010430  010067  000614                   MOV     R0,SAVR0
2433  010434  000002                           RTI
2434                                                    ;RESTORE R0-R5
2435
2436  010436  016700  000606          RS05:    MOV     SAVR0,R0
2437  010442  016701  000604                   MOV     SAVR1,R1
2438  010446  016702  000602                   MOV     SAVR2,R2
2439  010452  016703  000600                   MOV     SAVR3,R3
2440  010456  016704  000576                   MOV     SAVR4,R4
2441  010462  016705  000574                   MOV     SAVR5,R5
2442  010466  000002                           RTI
```

```
2443
2444                                                ;TELETYPE OUTPUT ROUTINE
2445
2446  010470  017605  000000          TYPER:  MOV     @(SP),R5
2447  010474  062716  000002                  ADD     #2,(SP)
2448  010500  105777  000466          1$:     TSTB    @TPCSR
2449  010504  100375                          BPL     1$
2450  010506  105715                          TSTB    (R5)
2451  010510  001001                          BNE     2$
2452  010512  000002                          RTI
2453  010514  112577  000454          2$:     MOVB    (R5)+,@TPDBR
2454  010520  000767                          BR      1$
2455
2456                                                ;ASCII STRING INPUT ROUTINE
2457
2458  010522  017667  000000  000006  INSTRG: MOV     @(SP),MSG
2459  010530  062716  000002                  ADD     #2,(SP)
2460  010534  104401                  INSTR1: TYPE
2461  010536  000000                  MSG:    0
2462  010540  012704  011714                  MOV     #INBUF,R4
2463  010544  012703  000007                  MOV     #7,R3
2464  010550  105777  000412          1$:     TSTB    @TKCSR
2465  010554  100375                          BPL     1$
2466  010556  117714  000406                  MOVB    @TKDBR,(R4)
2467  010562  142714  000200                  BICB    #200,(R4)
2468  010566  122427  000015                  CMPB    (R4)+,#15
2469  010572  001413                          BEQ     INSTR2
2470  010574  117777  000370  000372          MOVB    @TKDBR,@TPDBR
2471  010602  105777  000364          2$:     TSTB    @TPCSR
2472  010606  100375                          BPL     2$
2473  010610  005303                          DEC     R3
2474  010612  001356                          BNE     1$
2475  010614  104401                  INSTRE: TYPE
2476  010616  011547                          MQM
2477  010620  000745                          BR      INSTR1
2478  010622  000002                  INSTR2: RTI
```

```
2479
2480                                            ;CONVERT ASCII STRING TO OCTAL
2481
2482  010624  011605                PARAMS: MOV     (SP),R5
2483  010626  012567  000146                MOV     (R5)+,LOLIM
2484  010632  012567  000144                MOV     (R5)+,HILIM
2485  010636  012567  000142                MOV     (R5)+,DEVADR
2486  010642  112567  000140                MOVB    (R5)+,LOBITS
2487  010646  112567  000135                MOVB    (R5)+,ADRCNT
2488  010652  010516                        MOV     R5,(SP)
2489  010654  005005                PARAM1: CLR     R5
2490  010656  012704  011714                MOV     #INBUF,R4
2491  010662  122714  000015                CMPB    #15,(R4)
2492  010666  001420                        BEQ     PARERR
2493  010670  121427  000060        1$:     CMPB    (R4),#60
2494  010674  002415                        BLT     PARERR
2495  010676  121427  000067                CMPB    (R4),#67
2496  010702  003012                        BGT     PARERR
2497  010704  142714  000060                BICB    #60,(R4)
2498  010710  152405                        BISB    (R4)+,R5
2499  010712  122714  000015                CMPB    #15,(R4)
2500  010716  001406                        BEQ     LIMITS
2501  010720  006305                        ASL     R5
2502  010722  006305                        ASL     R5
2503  010724  006305                        ASL     R5
2504  010726  000760                        BR      1$
2505  010730  104404                PARERR: INSTER
2506  010732  000750                        BR      PARAM1
2507
2508                                            ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2509
2510  010734  020567  000042        LIMITS: CMP     R5,HILIM
2511  010740  101373                        BHI     PARERR
2512  010742  020567  000032                CMP     R5,LOLIM
2513  010746  103770                        BLO     PARERR
2514  010750  136705  000032                BITB    LOBITS,R5
2515  010754  001365                        BNE     PARERR
2516
2517                                            ;STORE NUMBER AT SPECIFIED ADDRESS
2518
2519  010756  016704  000022                MOV     DEVADR,R4
2520  010762  010524                1$:     MOV     R5,(R4)+
2521  010764  062705  000002                ADD     #2,R5
2522  010770  105367  000013                DECB    ADRCNT
2523  010774  001372                        BNE     1$
2524  010776  000002                        RTI
2525  011000  000000                LOLIM:  0
2526  011002  000000                HILIM:  0
2527  011004  000000                DEVADR: 0
2528  011006  000000                LOBITS: 0
2529          011007                ADRCNT=LOBITS+1
```

```
2530
2531                                        ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2532
2533  011010  104401              OCTASN: TYPE
2534  011012  011553                      MCRLF
2535  011014  017601  000000              MOV     @(SP),R1
2536  011020  062716  000002              ADD     #2,(SP)
2537  011024  012167  000130              MOV     (R1)+,WRDCNT
2538  011030  112167  000126      1$:     MOVB    (R1)+,CHRCNT
2539  011034  112167  000123              MOVB    (R1)+,SPACNT
2540  011040  013167  000120              MOV     @(R1)+,BINWRD
2541  011044  016704  000114      2$:     MOV     BINWRD,R4
2542  011050  116705  000106              MOVB    CHRCNT,R5
2543  011054  012700  011726              MOV     #TEMP,R0
2544  011060  010403              3$:     MOV     R4,R3
2545  011062  042703  177770              BIC     #177770,R3
2546  011066  062703  000260              ADD     #260,R3
2547  011072  110320                      MOVB    R3,(R0)+
2548  011074  006204                      ASR     R4
2549  011076  006204                      ASR     R4
2550  011100  006204                      ASR     R4
2551  011102  005305                      DEC     R5
2552  011104  001365                      BNE     3$
2553  011106  012703  011740              MOV     #MDATA,R3
2554  011112  114023              4$:     MOVB    -(R0),(R3)+
2555  011114  105367  000042              DECB    CHRCNT
2556  011120  001374                      BNE     4$
2557  011122  105767  000035              TSTB    SPACNT
2558  011126  001405                      BEQ     6$
2559  011130  112723  000240      5$:     MOVB    #240,(R3)+
2560  011134  105367  000023              DECB    SPACNT
2561  011140  001373                      BNE     5$
2562  011142  105013              6$:     CLRB    (R3)
2563  011144  104401                      TYPE
2564  011146  011740                      MDATA
2565  011150  005367  000004              DEC     WRDCNT
2566  011154  001325                      BNE     1$
2567  011156  000002                      RTI
2568  011160  000000              WRDCNT: 0
2569  011162  000000              CHRCNT: 0
2570          011163              SPACNT=CHRCNT+1
2571  011164  000000              BINWRD: 0
```

```
                                        ;INDIRECT POINTERS

011166  177560         TKCSR:  177560
011170  177562         TKDBR:  177562
011172  177564         TPCSR:  177564
011174  177566         TPDBR:  177566
011176  000000         DHSCR:  0
011200  000000         DHNRC:  0
011202  000000         DHLPR:  0
011204  000000         DHBR:   0
011206  000000         DHBC:   0
011210  000000         DHCAR:  0
011212  000000         DHBCR:  0
011214  000000         DHSSR:  0
011216  000000         DHSLR:  0
011220  000000         DHRVEC: 0
011222  000000         DHRLVL: 0
011224  000000         DHTVEC: 0
011226  000000         DHTLVL: 0
                                        ;PROGRAM VARIABLES

011230  000000         ERRFLG: 0               ;ERROR FLAG
011232  000000         PASCNT: 0               ;PASS COUNT
011234  000000         ERRCNT: 0               ;ERROR COUNT
011236  000000         RETURN: 0               ;SCOPE RETURN ADDRESS FOR TEST LOOPING
011240  000000         ESCAPE: 0               ;ADDRESS FOR ERROR ESCAPE
011242  000000         FREEZ1: 0               ;DATA LOOPING RETURN ADDRESS
011244  000000         ICOUNT: 0               ;ITERATION COUNT FOR TEST IN PROGRESS
011246  000000         LPCNT:  0               ;NUMBER OF ITERATIONS THIS TEST
011250  000000         SAVR0:  0               ;R0 SAVE AREA
011252  000000         SAVR1:  0               ;R1 SAVE AREA
011254  000000         SAVR2:  0               ;R2 SAVE AREA
011256  000000         SAVR3:  0               ;R3 SAVE ARE
011260  000000         SAVR4:  0               ;R4 SAVE AREA
011262  000000         SAVR5:  0               ;R5 SAVE AREA
011264  000000         SAVSP:  0               ;STACK POINTER SAVE AREA
011266  000000         SAVPC:  0               ;CALLING ROUTINE SAVE AREA
011270  000000         INIFLG: 0               ;PROGRAM INITIALIZATION FLAG
011272  000000         STFLG:  0               ;PROGRAM START FLAG
011274  000000         LAST:   0               ;LAST ERROR PC
```

```
                                                ;ENTER HERE ON POWER FAILURE
                                                
                                                
011276  010046                        PFAIL:  MOV     R0,-(SP)                ;SAVE R0-R5 ON PROCESSOR STACK
011300  010146                                MOV     R1,-(SP)
011302  010246                                MOV     R2,-(SP)
011304  010346                                MOV     R3,-(SP)
011306  010446                                MOV     R4,-(SP)
011310  010546                                MOV     R5,-(SP)
011312  016746  166506                        MOV     24,-(SP)
011316  010667  177742                        MOV     SP,SAVSP                ;SAVE STACK POINTER
011322  012767  011334  166474                MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
011330  000000                                HALT                            ;HALT ON POWER DOWN NORMAL
011332  000777                                BR      .
                                                
                                                ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
                                                
011334  016706  177724                RESTAR: MOV     SAVSP,SP                ;RESTORE STACK POINTER
011340  012605                                MOV     (SP)+,R5                ;RESTORE R0-R5
011342  012604                                MOV     (SP)+,R4
011344  012603                                MOV     (SP)+,R3
011346  012602                                MOV     (SP)+,R2
011350  012601                                MOV     (SP)+,R1
011352  012600                                MOV     (SP)+,R0
011354  012767  011276  166442                MOV     #PFAIL,24               ;SET UP FOR POWER FAILURE
011362  012767  000340  166406                MOV     #340,PS
011370  012706  012606                        MOV     #STACK,SP
011374  005067  000326                        CLR     TEMP
011400  005267  000322                        INC     TEMP
011404  001375                                BNE     .-4
011406  104402                                OCTASC
011410  011432                                PFTAB
011412  104401                                TYPE
011414  011556                                MPFAIL
011416  005067  177606                        CLR     ERRFLG
011422  005067  177646                        CLR     LAST
011426  000177  177604                        JMP     @RETURN
011432  000001                        PFTAB:  1
011434  000006  000002                        6,2
011440  000207                                RETURN
```

```
                011442  005015  042012  030510  MTITLE: .ASCIZ  <15><12><12>/DH11 MEMORY TEST /<15><12>
                011450  020061  042515  047515
                011456  054522  052040  051505
                011464  020124  005015      000
                011471      015  053012  041505  MVECTO: .ASCIZ  <15><12>/VECTOR ADDRESS-/
                011476  047524  020122  042101
                011504  051104  051505  026523
                011512      000
2660            011513      015  041412  047117  MREGAD: .ASCIZ  <15><12>/CONTROL REGISTER ADDRESS-/
2661            011520  051124  046117  051040
2662            011526  043505  051511  042524
2663            011534  020122  042101  051104
2664            011542  051505  026523      000
2665            011547      040  037440      000  MQM:    .ASCIZ  /  ?/
2666            011553      015  005012           MCRLF:  .ASCIZ  <15><12>
2667            011556  020040  047520  042527  MPFAIL: .ASCIZ  /  POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
2668            011564  020122  040506  046111
2669            011572  051125  026105  050040
2670            011600  047522  051137  046501
2671            011606  051040  051505  040524
2672            011614  052122  040440  020124
2673            011622  042524  052123  044440
2674            011630  020116  051120  043517
2675            011636  042522  051523      000
2676            011643      015  042012  042132  MEPASS: .ASCIZ  <15><12>/DZDH@/
2677            011650  041110      000
2678            011653      015  051012      000  MR:     .ASCIZ  <15><12>/R/
2679            011657      015  052012  051505  MTSTPC: .ASCIZ  <15><12>/TEST PC-/
2680            011664  020124  041520  000055
2681
2682                                                ;TABLE OF POINTERS FOR TRAP DECODING
2683
2684            011672  010024                      TRPTAB: SCOPER
2685            011674  010470                              TYPEP
2686            011676  011010                              OCTASN
2687            011700  010522                              INSTRG
2688            011702  010614                              INSTRE
2689            011704  010624                              PARAMS
2690            011706  010376                              SVOSP
2691            011710  010436                              RSOS
2692            011712  010124                              SCOPIR
2693
2694                                                ;BUFFERS FOR INPUT-OUTPUT
2695
2696            011714  000000                      INBUF:  0
2697                    011726                      .=.+10
2698            011726  000000                      TEMP:   0
2699                    011740                      .=.+10
2700            011740  000000                      MDATA:  0
2701                    011752                      .=.+10
2702
2703                                                ;TABLE OF POINTERS TO ERROR MESSAGES AND DATA
2704
2705            011752                              ERRTAB:
2706            011752  000000                              0
2707            011754  000000                              0
```

EOS


```
2708  011756  012002                                  EM1
2709  011760  012340                                  DT1
2710  011762  012064                                  EM2
2711  011764  012340                                  DT1
2712  011766  012154                                  EM3
2713  011770  012340                                  DT2
2714  011772  012213                                  EM4
2715  011774  012340                                  DT2
2716  011776  012360                                  EM5
2717  012000  012340                                  DT3
2718  012002  052502  020123  042101  EM1:    .ASCIZ  /BUS ADDRESS MEMORY ERROR/<15><12>/EXP     REC     ADDRESS/
2719  012024  051104  051505  020123
2720  012036  047515  047515  054522
2721  012050  052440  051122  051117
2722  012062  020115  054105  020120
2723  012074  020040  020040  042522
2724  012106  020103  020040  020040
2725  012120  042101  051104  051505
2726  012132  000123
2727  012134  054502  042524  041440  EM2:    .ASCIZ  /BYTE COUNT MEMORY ERROR/<15><12>/EXP     REC     ADDRESS/
2728  012172  052517  052116  046440
2729  012210  046505  051117  020131
2730  012226  051105  047522  006522
2731  012244  042412  050130  020040
2732  012262  020040  051040  041505
2733  012300  020040  020040  040440
2734  012316  042104  042522  051523
2735  012334      000
2736  012335      102  051525  040440  EM3:    .ASCIZ  /BUS ADDRESS MEMORY ERROR/<15><12>/EXP     REC/
2737  012353  042104  042522  051523
2738  012371  046440  046505  051117
2739  012407  020131  051105  047522
2740  012425  006522  042412  050130
2741  012443  020040  020040  051040
2742  012461  041505      000
2743  012464      102  052131  020105  EMH:    .ASCIZ  /BYTE COUNT MEMORY ERROR/<15><12>/EXP     REC/
2744  012520  047503  047125  020124
2745  012536  042515  047515  054522
2746  012554  042440  051122  051117
2747  012572  005015  054105  020120
2748  012610  020040  020040  042522
2749  012626  000103
2750  012630  042515  047515  054522  EM5:    .ASCIZ  /MEMORY EXTENSION ERROR/<15><12>/EXP     REC     ADDRESS/
2751  012666  042440  052130  047105
2752  012704  044523  047117  042440
2753  012722  051122  051117  005015
2754  012740  054105  020120  020040
2755  012756  020040  042522  020103
2756  013014  020040  020040  042101
2757  013032  051104  051505  000123
2758                          .EVEN
2759  013040  000003          DT1:    3
2760  013042      006  002     .BYTE   6,2
2761  013044  011254          SAVR2
2762  013046      006  002     .BYTE   6,2
2763  013050  011256          SAVR3
```

```
 2764  012352      002      000          .BYTE   2,0
 2765  012354  011260                            SAVR4
 2766  012356  000002          DT2:        2
 2767  012360      006      002          .BYTE   6,2
 2768  012362  011262                            SAVR5
 2769  012364      006      002          .BYTE   6,2
 2770  012366  011260                            SAVR4
 2771  012370  000003          DT3:        3
 2772  012372      006      002          .BYTE   6,2
 2773  012374  011262                            SAVR5
 2774  012376      006      002          .BYTE   6,2
 2775  012400  011260                            SAVR4
 2776  012402      002      000          .BYTE   2,0
 2777  012404  011256                            SAVR3
 2778  012406  000000          ENDCOD: 0
 2779          000001          .END
```

```
ADRCNT= 011007           2487*  2522*  2529#
ADRS  = 000000           1011#  1395#
BEGIN   001202            982    911    917#  2330
BINWRD  011164           2540*  2541   2571#
BIT00 = 000001            572#
BIT01 = 000002            571#
BIT02 = 000004            570#
BIT03 = 000010            569#
BIT04 = 000020            568#
BIT05 = 000040            567#
BIT06 = 000100            566#
BIT07 = 000200            565#
BIT08 = 000400            564#
BIT09 = 001000            563#
BIT10 = 002000            562#
BIT11 = 004000            561#
BIT12 = 010000            560#
BIT13 = 020000            559#
BIT14 = 040000            558#
BIT15 = 100000            557#
CADRS = 000000           1011#  1395#
CHRCNT  011162           2538*  2542   2555*  2569#  2570
DATABP  010270           2378*  2381   2388   2391#
DEVADR  011004           2485*  2519   2527#
DHBA    011204            954*   964   1022*  1023   1028*  1029   1046*  1047   1052*  1053   1070*  1071   1076*
                         1077   1094*  1095   1100*  1101   1118*  1119   1124*  1125   1142*  1143   1148*  1149
                         1166*  1167   1172*  1173   1190*  1191   1196*  1197   1214*  1215   1220*  1221   1238*
                         1239   1244*  1245   1262*  1263   1268*  1269   1286*  1287   1292*  1293   1310*  1311
                         1316*  1317   1334*  1335   1340*  1341   1358*  1359   1364*  1365   1382*  1383   1388*
                         1389   1798*  1804*  1811   1847*  1853*  1860   1896*  1902*  1909   2092*  2098*  2105
                         2187*  2193*  2233*  2239*  2279*  2285*  2581#
DHBAR   011210           2583#
DHBC    011206            991*  1001   1406*  1407   1412*  1413   1430*  1431   1436*  1437   1454*  1455   1460*
                         1461   1478*  1479   1484*  1485   1502*  1503   1508*  1509   1526*  1527   1532*  1533
                         1550*  1551   1556*  1557   1574*  1575   1580*  1581   1598*  1599   1604*  1605   1622*
                         1623   1628*  1629   1646*  1647   1652*  1653   1670*  1671   1676*  1677   1694*  1695
                         1700*  1701   1718*  1719   1724*  1725   1742*  1743   1748*  1749   1766*  1767   1772*
                         1773   1945*  1951*  1958   1994*  2000*  2007   2043*  2049*  2056   2141*  2147*  2154
                         2582#
DHBCR   011212           2584#
DHLPR   011202           2580#
DHNRC   011200           2579#
DHRLVL  011222           2588#
DHRVEC  011220            897   2587#
DHSCR   011176            905    952*   963*   989*  1000*  1020*  1044*  1068*  1092*  1116*  1140*  1164*  1188*
                         1212*  1236*  1260*  1284*  1308*  1332*  1356*  1380*  1404*  1428*  1452*  1476*  1500*
                         1524*  1548*  1572*  1596*  1620*  1644*  1668*  1692*  1716*  1740*  1764*  1797*  1800*
                         1803*  1805*  1812    1821*  1846*  1849*  1852*  1854*  1861   1870*  1895*  1898*  1901*
                         1903*  1910   1919*  1944*  1947*  1950*  1952*  1959   1968*  1993*  1996*  1999*  2001*
                         2008   2017*  2042*  2045*  2048*  2050*  2057   2066*  2091*  2094*  2097*  2099*  2106
                         2115*  2140*  2143*  2146*  2148*  2155   2164*  2185*  2186*  2188*  2191*  2192*  2194*
                         2201   2210*  2231*  2232*  2234*  2237*  2238*  2240*  2247   2256*  2277*  2278*  2280*
                         2283*  2284*  2286*  2293   2302*  2578#
DHSLR   011216            908*   909*  2586#
DHSSR   011214            908   2200   2246   2292   2585#
DHTLVL  011226           2590#
```

```
DHTVEC  011224          2589#
DT1     012340          2709    2711    2759#
DT2     012356          2713    2715    2766#
DT3     012370          2717    2771#
EMI     012002          2708    2718#
EM2     012064          2710    2727#
EM3     012145          2712    2736#
EM4     012213          2714    2743#
EM5     012260          2716    2750#
ENDCOD  012406           544    2778#
EOP     007752          2317#
ERRCNT  011234           873#   2399#   2595#
ERRFLG  011230           874#    875#   2320#   2345#   2350    2368#   2379    2385#   2593#   2646#
ERRMSG  010256          2377#   2387#
ERRVRS  010142           837    2363#
ERRTAB  011752          2376    2705#
ERTABO  010336          2384    2404#
ESCAPE  011240           947#    984#   1018#   1042#   1066#   1090#   1114#   1138#   1162#   1186#   1210#   1234#   1258#
                        1282#   1306#   1330#   1354#   1378#   1402#   1426#   1450#   1474#   1498#   1522#   1546#   1570#
                        1594#   1618#   1642#   1666#   1690#   1714#   1738#   1762#   1789#   1838#   1887#   1936#   1985#
                        2034#   2053#   2132#   2177#   2223#   2269#   2402    2597#
EXITER  010314          2394    2399#
FREEZI  011242          1790#   1839#   1888#   1937#   1986#   2035#   2084#   2133#   2178#   2224#   2270#   2358    2598#
HALTS   010274          2364    2393#
HILIM   011002          2484#   2510    2526#
ICOUNT  011244           946#    983#   1017#   1041#   1065#   1089#   1113#   1137#   1161#   1185#   1209#   1233#   1257#
                        1281#   1305#   1329#   1353#   1377#   1401#   1425#   1449#   1473#   1497#   1521#   1545#   1569#
                        1593#   1617#   1641#   1665#   1689#   1713#   1737#   1761#   1788#   1837#   1886#   1935#   1984#
                        2033#   2082#   2131#   2176#   2222#   2268#   2342    2599#
INCIF   011714          2482    2490    2696#
INTFLG  011270           877     910     912#   2609#
INSTEK= 104404           854#   2505
INSTR = 104403           853#    892     900     921
INSTRE  010614          2475#   2638
INSTRG  010622          2458#   2637
INSTRI  010634          2460#   2477
INSTR2  010622          2469    2478#
LRST    011274          2319#   2365    2367#   2611#   2647#
LIGHTS= 177570           542#   2322#
LIMITS  010734          2500    2510#
LOBITS  011006          2486#   2514    2528#   2529
LOGICA  010010          2326#
LOLIM   011000          2483#   2512    2525#
LPCNT   011246          2341#   2342    2344#   2600#
MCRLF   011553          2534    2666#
MDATA   011740          2553    2564    2700#
M.PASS  011643          2318    2676#
M.FAIL  011556          2645    2667#
MMM     011547          2476    2665#
MMR     011653           934    2678#
MMREGAD 011513           901    2660#
MSG     010536          2458#   2461#
MTITLE  011442           876    2652#
MTSTPC  011657           922    2679#
MVECTO  011471           893    2656#
N       000001             1#
```

DZDHB.PFC CROSS REFERENCE TABLE -- USER SYMBOLS

```
OCTASC= 104402            852#   2383   2390   2642
OCTASN  011010           2533#   2686
PARAM = 104405            855#    894    902    923
PARAMS  010624           2482#   2689
PARAM1  010654           2489#   2506
PARERR  010730           2492    2494   2496   2505#  2511   2513   2515
PDISCNT 011232            872#   2321#  2322   2594#
PC     =%000007           537#   2326#
PFAIL   011276            835     870   2615#  2636
PFTAB   011432           2643    2649#
POPRO = 012600            551#   2398
POP1SP= 005726            549#
POP2SP= 022626            553#
PS     = 177776           543#    868#   917#   945#   982#  1016#  1040#  1064#  1088#  1112#  1136#  1160#  1184#
                         1203#   1232#  1256#  1280#  1304#  1328#  1352#  1376#  1400#  1424#  1448#  1472#  1496#
                         1520#   1544#  1568#  1592#  1616#  1640#  1664#  1688#  1712#  1736#  1760#  1787#  1836#
                         1885#   1934#  1983#  2032#  2081#  2130#  2175#  2221#  2267#  2637#
PUSHRO= 010046            550#   2395
PUSH1S= 005746            548#
PUSH2S= 024646            552#
RESREG  010272           2389    2392#
RESTAR  011334           2623    2629#
RESTRT  010020           2324    2330#
RESOS = 104407            857#   2392
RETURN  011236            930#    935   2346#  2348   2596#  2648
RSOS    010436           2436#   2691
R0     =%000000           530#    948#   957#   959#   971#   985#   994#   996#  1008#  1791#  1825#  1840#  1874#
                         1859#   1923#  1939#  1972#  1987#  2021#  2036#  2070#  2085#  2119#  2134#  2168#  2179#
                         2214#   2225#  2260#  2271#  2306#  2396#  2432   2436#  2543#  2547#  2554   2615   2635#
R1     =%000001           531#    883#   886#   888#   890   1795#  1801#  1806#  1812#  1844#  1850#  1855#  1871#
                         1893#   1899#  1904#  1950#  1942#  1948#  1953#  1959#  1991#  1937#  2002#  2018#  2040#
                         2046#   2051#  2067#  2089#  2095#  2100#  2116#  2138#  2144#  2149#  2165#  2183#  2189#
                         2195#   2211#  2229#  2235#  2241#  2257#  2275#  2281#  2287#  2303#  2323#  2326   2431
                         2437#   2535#  2537   2538   2539   2540   2616   2634#
R2     =%000002           532#    884#   886    887#   889#   951#   954    955#   962#   965    970#   988#   991
                          992#    999#  1002   1007#  2430   2438#  2617   2633#
R3     =%000003           533#    885#   888    889    964#   965   1001#  1002   1793#  1803   1812   1824#  1842#
                         1852    1861   1873#  1891#  1901   1910   1932#  1940#  1950   1959   1971#  1989#  1999
                         2008    2020#  2038#  2048   2057   2079#  2087#  2097   2105   2118#  2136#  2146   2155
                         2167#   2181#  2191   2201   2213#  2237#  2337   2247   2283#  2273#  2303   2293   2305#
                         2429    2439#  2463#  2473#  2544#  2545#  2546#  2547   2553#  2554#  2559#  2562#  2618
                         2632#
R4     =%000004           534#    950#   952    956#   961#   973    979#   987#   989    993#   998#  1000   1006#
                         1023#   1024   1038#  1047#  1048   1053#  1071#  1072   1077#  1083#  1096   1101#  1119#
                         1120    1125#  1143#  1144   1149#  1167#  1168   1173#  1191#  1192   1197#  1215#  1216
                         1221#   1239#  1240   1245#  1263#  1264   1269#  1287#  1289   1293#  1311#  1312   1317#
                         1335#   1336   1341#  1359#  1360   1385   1393   1404   1389#  1407#  1408   1413#  1431#
                         1432    1437#  1455#  1456   1461#  1479#  1480   1485   1503#  1504   1509#  1527#  1528
                         1533#   1551#  1553   1557#  1575#  1576   1581#  1589#  1600   1605#  1623#  1624   1629#
                         1647#   1648   1653#  1671#  1672   1677#  1695#  1696   1701#  1719#  1720   1725#  1743#
                         1744    1749#  1767#  1768   1773#  1811#  1817   1860#  1866   1909#  1915   1958#  1964
                         2007#   2013   2056#  2062   2105#  2111   2154#  2160   2200#  2206   2246#  2252   2292#
                         2313    2372#  2373#  2374#  2375#  2376#  2377   2378   2428   2440#  2462#  2466#  2467#
                         2483    2490#  2491   2493   2495   2497#  2498   2499   2519#  2520#  2541#  2544   2548#
                         2543#   2550#  2619   2631#
R5     =%000005           535#   1019#  1024   1027#  1043#  1048   1051#  1067#  1072   1075#  1091#  1096   1099#
```

DZDH88.PFC CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    1115*  1120   1123*  1139*  1144   1147*  1163*  1168   1171*  1187*  1192   1195*  1211*
                    1216   1219*  1235*  1240   1243*  1259*  1264   1267*  1283*  1289   1291*  1307*  1312
                    1315*  1331*  1336   1339*  1355*  1360   1363*  1379*  1384   1387*  1403*  1408   1411*
                    1427*  1432   1435*  1451*  1456   1459*  1475*  1480   1483*  1499*  1504   1507*  1523*
                    1528   1531*  1547*  1552   1555*  1571*  1576   1579*  1595*  1600   1603*  1619*  1634
                    1627*  1643*  1648   1651*  1667*  1672   1675*  1691*  1696   1699*  1715*  1720   1723*
                    1739*  1744   1747*  1763*  1768   1771*  1808*  1816*  1817   1857*  1865*  1866   1906*
                    1914*  1915   1955*  1963*  1964   2004*  2012*  2013   2053*  2061*  2062   2102*  2110*
                    2111   2151*  2159*  2160   2197*  2205*  2206   2243*  2251*  2252   2289*  2297*  2298
                    2370*  2371*  2372   2427   2441*  2446*  2450   2453   2482*  2483   2484   2495   2496
                    2487   2488   2489*  2498*  2501*  2502*  2503*  2510   2512   2514   2520   2521*  2542*
                    2551*  2620   2630*
SAVPC   011266      2406   2423*  2608#
SAVR0   011250      2432*  2436   2601#
SAVR1   011252      2431*  2437   2602#
SAVR2   011254      2430*  2438   2603#  2761
SAVR3   011256      2429*  2439   2604#  2763   2777
SAVR4   011260      2428*  2440   2605#  2765   2770   2775
SAVR5   011262      2427*  2441   2606#  2768   2773
SAVSP   011264      2607#  2622*  2629
SAVOSP= 104406       856#  2369
SCOPE  = 104400      850#   973   1010   1034   1058   1082   1106   1130   1154   1178   1202   1226   1250
                    1274   1298   1322   1346   1370   1394   1418   1442   1466   1490   1514   1538   1562
                    1596   1610   1634   1658   1682   1706   1730   1754   1778   1827   1876   1925   1974
                    2023   2072   2121   2170   2216   2262   2308
SCOPER  010024      2335#  2684
SCOPE1= 104410       858#  1820   1869   1918   1967   2016   2065   2114   2163   2209   2255   2301
SCOPIR  010124      2396#  2692
SP     =%000006      536#   869*   918*  2346   2348*  2358*  2365   2367   2370   2396   2402*  2412*  2413*
                    2414*  2415*  2416*  2417*  2418*  2419   2423   2446   2447*  2458   2459*  2482   2488*
                    2535   2536*  2615*  2616*  2617*  2618*  2619*  2620*  2621*  2622   2629*  2630   2631
                    2632   2633   2634   2635   2638*
SPACNT= 011163      2539*  2557   2560*  2570#
STACK  = 012606      544#   669    918   2638
START   001000       842    868#
STFLG   011272       871*   931    933*  2610#
SVOS    010404      2427#
SVOSP   010376      2423#  2690
SWR    = 177570      541#   881    919   2335   2337   2339   2356   2363   2393   2400
SW00   = 000001      524#   881
SW01   = 000002      523#   919
SW02   = 000004      522#
SW03   = 000010      521#
SW04   = 000020      520#
SW05   = 000040      519#
SW06   = 000100      518#
SW08   = 000400      517#
SW09   = 001000      516#  2356
SW10   = 002000      515.  2335   2400
SW11   = 004000      514*  2339
SW12   = 010000      513#
SW13   = 020000      512#  2363
SW14   = 040000      511#  2337
SW15   = 100000      510#
TEMP    011726      2543   2639*  2640*  2698#
TKCSR   011166      2464   2574#
```

DZDMB8.PFC CROSS REFERENCE TABLE -- USER SYMBOLS

```
TKDBR   011170            2466    2470    2575#
TPCSR   011172            2448    2471    2576#
TPDBR   011174            2453#   2470#   2577#
TRPOK   010356            2415#
TRPSRV  010344             839    2412#
TRPTAB  011672            2417    2684#
TYPDAT  010260            2382    2388#
TYPE  = 104401             851#    876     934    2317    2386    2460    2475    2533    2563    2644
TYPER   010470            2446#   2685
TYPMSG  010242            2380    2383#
T1      001274             930     945#
T10     002216            1136#
T11     002314            1160#
T12     002412            1184#
T13     002510            1208#
T14     002606            1232#
T15     002704            1256#
T16     003002            1280#
T17     003100            1304#
T2      001412             982#
T20     003176            1328#
T21     003274            1352#
T22     003372            1376#
T23     003470            1400#
T24     003566            1424#
T25     003664            1448#
T26     003762            1472#
T27     004060            1496#
T3      001530            1016#
T30     004156            1520#
T31     004254            1544#
T32     004352            1568#
T33     004450            1592#
T34     004546            1616#
T35     004644            1640#
T36     004742            1664#
T37     005040            1688#
T4      001626            1040#
T40     005136            1712#
T41     005234            1736#
T42     005332            1760#
T43     005430            1784#
T44     005526            1808#
T45     005624            1832#
T46     005722            1856#
T47     006310            1983#
T5      001724            1064#
T50     006464            2032#
T51     006540            2081#
T52     007016            2130#
T53     007174            2175#
T54     007346            2221#
T55     007500            2267#
T6      002002            1088#
T7      002100            1112#
VEC1    001060             878     881#
```

DZDHB8.PFC CROSS REFERENCE TABLE -- USER SYMBOLS

```
VEC2    001070            880    883#
WRDCNT  011160           2537#  2565#  2568#
X       = 000000            1#
XADRS   = 000020         1011#  1035#  1059#  1083#  1107#  1131#  1155#  1179#  1203#  1227#  1251#  1275#  1299#
                         1323#  1347#  1371#  1395#  1419#  1443#  1467#  1491#  1515#  1539#  1563#  1587#  1611#
                         1635#  1659#  1683#  1707#  1731#  1755#  1779#
XCADRS= 000020           1011#  1035#  1059#  1083#  1107#  1131#  1155#  1179#  1203#  1227#  1251#  1275#  1299#
                         1323#  1347#  1371#  1395#  1419#  1443#  1467#  1491#  1515#  1539#  1563#  1587#  1611#
                         1635#  1659#  1683#  1707#  1731#  1755#  1779#
XN      = 000056            1#   945    948#   982    985#  1016  1019#  1040  1043#  1064  1067#  1088  1091#
                         1112  1115#  1136  1139#  1160  1163#  1184  1187#  1208  1211#  1232  1235#  1256
                         1259#  1280  1283#  1304  1307#  1328  1331#  1352  1355#  1376  1379#  1400  1403#
                         1424  1427#  1448  1451#  1472  1475#  1496  1499#  1520  1523#  1544  1547#  1568
                         1571#  1592  1595#  1616  1619#  1640  1643#  1664  1667#  1688  1691#  1712  1715#
                         1736  1739#  1760  1763#  1787  1791#  1836  1840#  1885  1889#  1934  1938#  1983
                         1987#  2032  2036#  2081  2085#  2130  2134#  2175  2179#  2221  2225#  2267  2271#
Y       = 000011            1#   850    851#   852#   853#   854#   855#   856#   857#   858#   859#
.       = 012410          574#   575    577    579    531    583    585    587    589    591    593    595    597
                          599    601    603    605    607    609    611    613    615    617    619    621    623
                          625    627    629    631    633    635    637    639    641    643    645    647    649
                          651    653    655    657    659    661    663    665    667    669    671    673    675
                          677    679    681    683    685    687    689    691    693    695    697    699    701
                          703    705    707    709    711    713    715    717    719    721    723    725    727
                          729    731    733    735    737    739    741    743    745    747    749    751    753
                          755    757    759    761    763    765    767    769    771    773    775    777    779
                          781    783    785    787    789    791    793    795    797    799    801    803    805
                          807    809    811    813    815    817    819    821    823    825    827    829    834#
                          841#   859#  2625   2641   2697#  2699#  2701#
```

DZDHBB.PFC CROSS REFERENCE TABLE -- MACRO NAMES

```
CODEM1    859#    908
HLT       554#    968   1005   1026   1032   1050   1056   1074   1080   1098   1104   1122   1128   1146   1152
         1170   1176   1194   1200   1218   1224   1242   1248   1266   1272   1290   1296   1314   1320   1338
         1344   1362   1368   1386   1392   1410   1416   1434   1440   1458   1464   1482   1488   1506   1512
         1530   1536   1554   1560   1578   1584   1602   1608   1626   1632   1650   1656   1674   1680   1698
         1704   1722   1728   1746   1752   1770   1776   1819   1868   1917   1966   2015   2064   2113   2162
         2208   2254   2300
MEMT1     936#    937    974
MEMT2     936#   1011   1035   1059   1083   1107   1131   1155   1179   1203   1227   1251   1275   1299   1323
         1347   1371   1395   1419   1443   1467   1491   1515   1539   1563   1587   1611   1635   1659   1683
         1707   1731   1755
MEMT3     936#   1779   1828   1877   1926   1975   2024
MEMT4     936#   2073   2122
MXTST2    936#   2171   2217   2263
TRPDEF      1#    850    851    852    853    854    855    856    857    858
TS          1#    945    982   1016   1040   1064   1098   1112   1136   1160   1184   1208   1232   1256   1280
         1304   1328   1352   1376   1400   1424   1448   1472   1496   1520   1544   1568   1592   1616   1640
         1664   1688   1712   1736   1760   1787   1836   1885   1934   1983   2032   2081   2130   2175   2221
         2267
.BUFFE      1#   2693
.CATCH      1#    573
.ENDCO      1#   2778
.EOP        1#   2310
.ERROR      1#   2360
.ERRTA      1#   2702
.HEADE      1#    488
.INSTR      1#   2455
.MSG        1#   2652
.OCTAS      1#   2530
.PARAM      1#   2479
.PFAIL      1#   2612
.POINT      1#   2572
.RESRE      1#   2434
.SAVRE      1#   2420
.SCOPE      1#   2331
.SCOPI      1#   2353
.SETVE      1#    831
.START      1#    859
.SYMBO      1#    507
.TRPDE      1#    845
.TRPSR      1#   2407
.TRPTA      1#   2681
.TYPER      1#   2443
.VARIA      1#   2591
```

DZDMB8.PFC CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
A00      888   889   955   970   992  1007  2376  2417  2447  2459  2521  2536  2546
ASL     2373  2374  2415  2501  2502  2503
ASR     2548  2549  2550
BEQ      882   920   967  1004  1025  1031  1049  1055  1073  1079  1097  1103  1121  1127  1145
        1151  1169  1175  1193  1199  1217  1223  1241  1247  1265  1271  1289  1295  1313  1319
        1337  1343  1361  1367  1385  1391  1409  1415  1433  1439  1457  1463  1481  1487  1505
        1511  1529  1535  1553  1559  1577  1583  1601  1607  1625  1631  1649  1655  1673  1679
        1697  1703  1721  1727  1745  1751  1769  1775  1815  1818  1864  1867  1913  1916  1962
        1965  2011  2014  2060  2063  2109  2112  2158  2161  2204  2207  2250  2253  2296  2299
        2324  2351  2357  2366  2380  2389  2401  2469  2492  2500  2558
BGT     2496
BHI     2511
BIC     1028  1052  1076  1100  1124  1148  1172  1196  1220  1244  1268  1292  1316  1340  1364
        1388  1412  1436  1460  1484  1508  1532  1556  1580  1604  1628  1652  1676  1700  1724
        1748  1772  2186  2232  2278  2375  2416  2545
BICB    2467  2497
BIS     2192  2238  2284
BISB    2498
BIT      881   919  2335  2337  2339  2356  2363  2400
BITB    2514
BLO     2513
BLT     2494
BNE      878   891   911   932   958   972   995  1009  1802  1823  1826  1851  1872  1875  1900
        1921  1924  1949  1970  1973  1998  2019  2022  2047  2068  2071  2096  2117  2120  2145
        2166  2169  2190  2212  2215  2236  2258  2261  2282  2304  2307  2336  2338  2340  2343
        2364  2382  2451  2474  2515  2523  2552  2556  2561  2566  2641
BPL     2394  2449  2465  2472
BR       880   929  2352  2454  2477  2504  2506  2625
CLR      871   872   873   874   875   887   950   951   961   962   987   988   998   999  1027
        1051  1075  1099  1123  1147  1171  1195  1219  1243  1267  1291  1315  1339  1363  1387
        1411  1435  1459  1483  1507  1531  1555  1579  1603  1627  1651  1675  1699  1723  1747
        1771  1793  1797  1798  1815  1816  1842  1846  1847  1854  1865  1891  1895  1896  1903
        1914  1945  1944  1945  1952  1963  1989  1993  1994  2001  2012  2038  2042  2043  2050
        2061  2067  2091  2099  2136  2140  2148  2181  2185  2194  2205  2227  2231  2240  2251
        2273  2277  2286  2297  2310  2320  2344  2345  2368  2489  2639  2646  2647
CLRB    2552
CMP      890   965  1002  1024  1048  1072  1096  1120  1144  1168  1192  1216  1240  1264  1288
        1312  1336  1360  1384  1408  1432  1456  1480  1504  1528  1552  1576  1600  1624  1648
        1672  1696  1720  1744  1768  1812  1817  1861  1886  1910  1915  1959  1964  2008  2013
        2057  2062  2106  2111  2155  2160  2201  2206  2247  2252  2293  2298  2342  2365  2510
        2512
CMPB    2468  2491  2493  2495  2499
COM      912   933  2110  2159
DEC      957   971   994  1008  1801  1822  1825  1850  1871  1874  1899  1920  1923  1948  1969
        1972  1997  2018  2021  2046  2067  2070  2095  2116  2119  2144  2165  2168  2189  2211
        2214  2235  2257  2260  2281  2303  2306  2473  2551  2565
DECB    2522  2555  2560
EMT      554
HALT     576   578   580   582   584   586   588   590   592   594   596   598   600   602   604
         606   608   610   612   614   616   618   620   622   624   626   628   630   632   634
         636   638   640   642   644   646   648   650   652   654   656   658   660   662   664
         666   668   670   672   674   676   678   680   682   684   686   688   690   692   694
         696   698   700   702   704   706   708   710   712   714   716   718   720   722   724
         726   728   730   732   734   736   738   740   742   744   746   748   750   752   754
         756   758   760   762   764   766   768   770   772   774   776   778   780   782   784
         786   788   790   792   794   796   798   800   802   804   806   808   810   812   814
```

	816	818	820	822	824	826	828	830	2397	2624					
INC	909	956	969	993	1006	1830	1821	1824	1849	1870	1873	1898	1919	1922	1947
	1968	1971	1996	2017	2020	2045	2066	2069	2094	2115	2118	2143	2164	2167	2188
	2210	2213	2234	2256	2259	2280	2302	2305	2321	2341	2399	2640			
JMP	842	935	2330	2419	2648										
JSR	2326														
MOV	868	869	870	883	884	895	896	909	917	918	930	945	946	947	948
	952	954	959	963	964	982	983	984	985	389	991	996	1000	1001	1016
	1017	1018	1019	1020	1022	1023	1029	1040	1041	1042	1043	1044	1046	1047	1053
	1064	1065	1066	1067	1068	1070	1071	1077	1088	1089	1090	1091	1092	1094	1095
	1101	1112	1113	1114	1115	1116	1118	1119	1125	1136	1137	1138	1139	1140	1142
	1143	1149	1160	1161	1162	1163	1164	1166	1167	1173	1184	1185	1186	1187	1188
	1190	1191	1197	1208	1209	1210	1211	1212	1214	1215	1221	1232	1233	1234	1235
	1236	1238	1239	1245	1256	1257	1258	1259	1260	1262	1263	1269	1280	1281	1282
	1283	1284	1286	1287	1293	1304	1305	1306	1307	1308	1310	1311	1317	1328	1329
	1330	1331	1332	1334	1335	1341	1352	1353	1354	1355	1356	1358	1359	1365	1376
	1377	1378	1379	1380	1382	1383	1389	1400	1401	1402	1403	1404	1406	1407	1413
	1424	1425	1426	1427	1428	1430	1431	1437	1448	1449	1450	1451	1452	1454	1455
	1461	1472	1473	1474	1475	1476	1478	1479	1485	1496	1497	1498	1499	1500	1502
	1503	1509	1520	1521	1522	1523	1524	1526	1527	1533	1544	1545	1546	1547	1548
	1550	1551	1557	1568	1569	1570	1571	1572	1574	1575	1581	1592	1593	1594	1595
	1596	1598	1599	1605	1616	1617	1618	1619	1620	1622	1623	1629	1640	1641	1642
	1643	1644	1646	1647	1653	1664	1665	1666	1667	1668	1670	1671	1677	1688	1689
	1690	1691	1692	1694	1695	1701	1712	1713	1714	1715	1716	1718	1719	1725	1736
	1737	1738	1739	1740	1742	1743	1749	1760	1761	1762	1763	1764	1766	1767	1773
	1787	1788	1789	1790	1791	1795	1803	1804	1806	1808	1811	1836	1837	1838	1839
	1840	1844	1852	1853	1855	1857	1860	1885	1886	1887	1898	1869	1893	1901	1902
	1904	1906	1909	1934	1935	1936	1937	1938	1942	1950	1951	1953	1955	1958	1983
	1984	1985	1986	1987	1991	1999	2000	2002	2004	2007	2032	2033	2034	2035	2036
	2040	2048	2049	2051	2053	2056	2081	2082	2083	2084	2085	2089	2092	2097	2098
	2100	2102	2105	2130	2131	2132	2133	2134	2138	2141	2146	2147	2149	2151	2154
	2175	2176	2177	2178	2179	2183	2187	2191	2193	2195	2197	2200	2221	2222	2223
	2224	2225	2229	2233	2237	2239	2241	2243	2246	2267	2269	2269	2270	2271	2275
	2279	2283	2285	2287	2289	2292	2322	2323	2346	2348	2358	2367	2370	2372	2377
	2378	2385	2396	2402	2412	2414	2418	2423	2427	2428	2429	2430	2431	2432	2436
	2437	2438	2439	2440	2441	2446	2458	2462	2463	2482	2483	2484	2485	2488	2490
	2519	2520	2535	2537	2540	2541	2543	2544	2553	2615	2616	2617	2618	2619	2620
	2621	2622	2623	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638		
MOVB	2452	2466	2470	2486	2487	2538	2539	2542	2547	2554	2559				
NOP	2327	2328	2329												
RESET	2325														
RETURN	926	2651													
RTI	2347	2349	2369	2403	2433	2442	2452	2478	2524	2567					
SUB	2271	2413													
TRAP	850	851	852	853	854	855	856	857	858						
TST	877	910	931	2350	2379	2381	2383	2393							
TSTB	2448	2450	2464	2471	2557										
.ASCIZ	2652	2656	2660	2665	2666	2667	2676	2678	2679	2718	2727	2736	2743	2750	
.BYTE	898	899	906	907	927	928	2405	2760	2762	2764	2767	2769	2772	2774	2776
.ENABL	507														
.END	2779														
.ENDC	880	881	908	910	948	995	1019	1043	1067	1091	1115	1139	1163	1187	1211
	1235	1259	1283	1307	1331	1355	1379	1403	1427	1451	1475	1499	1523	1547	1571
	1595	1619	1643	1667	1691	1715	1739	1763	1791	1840	1889	1938	1987	2036	2085
	2134	2179	2225	2271											
.EQUIV	554														

C06


```
.EVEN    2758
.IF       878   880   908   948   985  1019  1043  1067  1091  1115  1139  1163  1187  1211  1235
         1259  1283  1307  1331  1355  1379  1403  1427  1451  1475  1499  1523  1547  1571  1595
         1619  1643  1667  1691  1715  1739  1763  1790  1839  1888  1937  1986  2035  2084  2133
         2178  2224  2270
.IFF      680   681
.IIF      867   968   969  1005  1026  1027  1032  1033  1050  1051  1056  1057  1074  1075  1080
         1081  1098  1099  1104  1105  1122  1123  1128  1129  1146  1147  1152  1153  1170  1171
         1176  1177  1194  1195  1200  1201  1218  1219  1224  1225  1242  1243  1248  1249  1266
         1267  1272  1273  1290  1291  1296  1297  1314  1315  1320  1321  1338  1339  1344  1345
         1362  1363  1368  1369  1386  1387  1392  1393  1410  1416  1434  1440  1458  1464  1482
         1488  1506  1512  1530  1536  1554  1560  1578  1584  1602  1608  1626  1632  1650  1656
         1674  1680  1698  1704  1722  1728  1746  1752  1770  1776  1819  1820  1868  1869  1917
         1918  1962  2015  2064  2113  2114  2162
.IRP     2578  2612
.LIST       1   488   507   851   852   853   854   855   856   857   858   859   936   948   985
         1011  1019  1035  1043  1059  1067  1083  1091  1107  1115  1131  1139  1155  1163  1179
         1187  1203  1211  1227  1235  1251  1259  1275  1283  1299  1307  1323  1331  1347  1355
         1371  1379  1395  1403  1419  1427  1443  1451  1467  1475  1491  1499  1515  1523  1539
         1547  1563  1571  1587  1595  1611  1619  1635  1643  1659  1667  1683  1691  1707  1715
         1731  1739  1755  1763  1779  1791  1840  1889  1938  1987  2036  2085  2134  2179  2225
         2271
.MACRO      1   859   936
.NLIST      1   488   507   851   852   853   854   855   856   857   858   859   936   948   985
         1011  1019  1035  1043  1059  1067  1083  1091  1107  1115  1131  1139  1155  1163  1179
         1187  1203  1211  1227  1235  1251  1259  1275  1283  1299  1307  1323  1331  1347  1355
         1371  1379  1395  1403  1419  1427  1443  1451  1467  1475  1491  1499  1515  1523  1539
         1547  1563  1571  1587  1595  1611  1619  1635  1643  1659  1667  1683  1691  1707  1715
         1731  1739  1755  1763  1779  1791  1840  1889  1938  1987  2036  2085  2134  2179  2225
         2271
.PAGE     526   573   831   859  2360  2407  2443  2479  2530  2572  2612  2652
.REM        1
.REPT     575  1011  1395
.TITLE    307
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

```
@DZDHB8,DZDHB8.SEQ/SOL/CRF/PAGNUM=DSK2:UTIL2.P11,DSKM:DZDHB8.PFC
RUN-TIME: 13 22 3 SECONDS
RUN-TIME RATIO: 106/40=2.6
CORE USED:  12K  (23 PAGES)
```

DOS

Spooler runtime 10 Seconds, 96 KCS, 274 disk reads, 3 disk writes, 67 pages

[illegible]